1. 根据配置文件生成mychannel.tx(channel的配置文件)

2. 连order节点生成创世区块，需要传orderer的证书，需要进docker容器执行命令，docker的环境变量和用户要确定，创建联盟必须用orderer的admin用户（无论那个组织都行），会生成mychannel.block的pb文件，然后把peer节点加入到mychannel。所以创建channel一定要把用户的环境变量配置好，然后上块（需要拿到channel）。channel必须跟orderer关联;

3. 实例化合约(上块)：client去相应节点背书，拿到提案，如果提案正常把提案发给orderer，利用orderer的Transaction来上块。实例化合约需要peer，因为合约连接到peer节点的，整个网络的peer节点都会实例化（同一个合约可以发送到不同的channel,所以发送合约需要channel来实现）

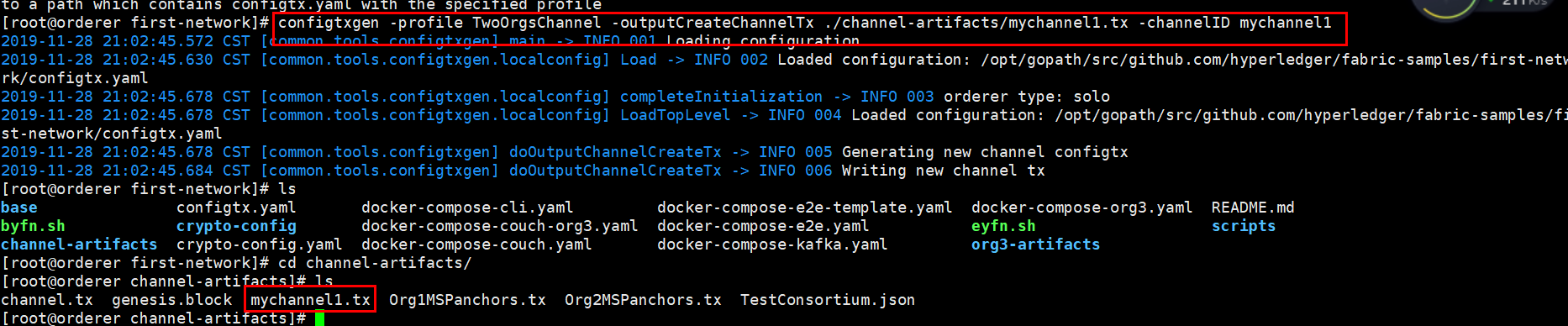
4. 实例化合约需要指定背书策略，不指定会有默认背书策略，默认一个同意就行

查看已安装的合约：peer chaincode list --installed

查看已实例化的合约：peer chaincode list --instantiated -C channel名称

常用命令：

1. 根据configtx.yaml的配置生成channel的配置文件：configtxgen -profile TwoOrgsChannel -outputCreateChannelTx ./channel-artifacts/mychannel1.tx -channelID mychannel1；



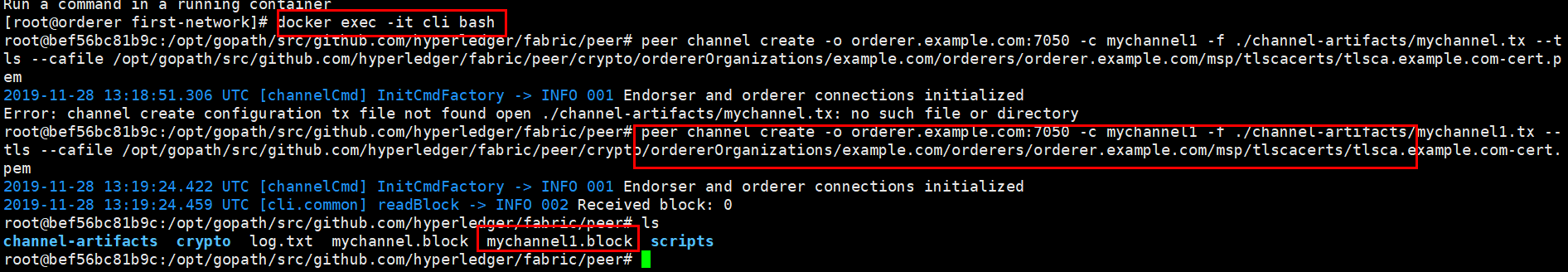
1. 生成mychannel1的创世区块，需要orderer的证书，且需要orderer组织的admin用户创建，可以去docker-compose-cli.yaml查看环境变量的配置，会生成block文件（创世区块的pb文件）；

peer channel create -o orderer.example.com:7050 -c mychannel1 -f ./channel-artifacts/mychannel1.tx --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

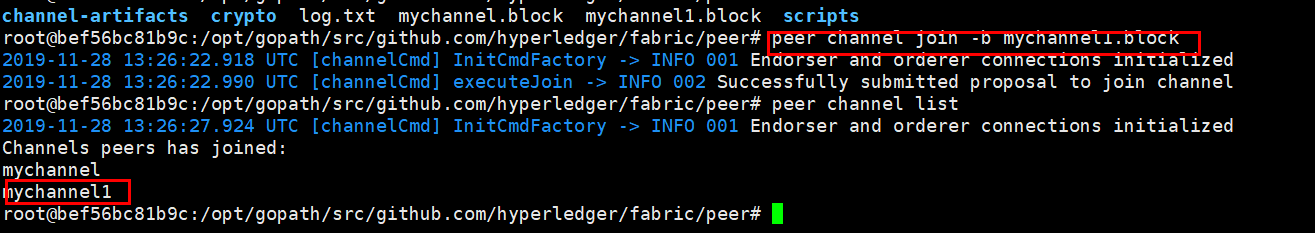
上一步也可以把orderer的证书配置到环境变量，简化为

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 create -o orderer.example.com:7050 -c testchannel -f ./channel-artifacts/testchannel.tx --tls --cafile $ORDERER\_CA



1. 把peer节点加入channel；peer channel join -b testchannel.block



1. 数字签名需要私钥和证书
2. 启动ca服务 ：docker-compose -f docker-compose-ca.yaml up -d