# 区块链 lab3

学号：PB18111793

姓名：裴启智

# 实验目标

使用Fabric搭建一个peer节点，并用这个peer节点加入已经创建的通道之中

# 实验过程

在222.195.70.186这个服务器上已经搭建了**CA服务器**和**Orderer服务器**,CA服务端口号为**7054**，orderer服务端口号为**7050**。

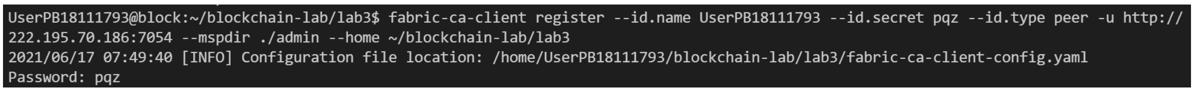
## CA服务

- 目标：使用CA服务器注册身份，并获得CA服务器颁发的身份证书
- 服务器中事先拥有CA的admin证书，组织的admin证书（两个管理员证书的作用域不同，前者管理CA服务，后者管理组织服务）。需要使用这个CA的admin证书去注册自己在fabric组织中的**peer证书**。主要使用的是CA客户端的服务。
- 主要指令如下

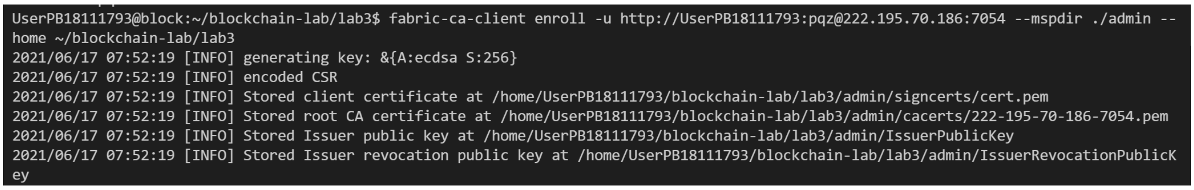
```
注册组织peer（一定要记得 --home指定目录或者export设置工作目录环境变量)
fabric-ca-client register --id.name UserPB18111793 --id.secret pqz --id.type peer -u http://222.195.70.186:7054 --mspdir ./admin --home ~/blockchain-lab/lab3

获得组织peer的msp
fabric-ca-client enroll -u http://UserPB18111793:pqz@222.195.70.186:7054 --mspdir ./admin --home ~/blockchain-lab/lab3
```

- 分别执行上述指令后结果如下



```
UserPB18111793@block:~/blockchain-lab/lab3$ fabric-ca-client register --id.name UserPB18111793 --id.secret pqz --id.type peer -u http://
222.195.70.186:7054 --mspdir ./admin --home ~/blockchain-lab/lab3
2021/06/17 07:49:40 [INFO] Configuration file location: /home/UserPB18111793/blockchain-lab/lab3/fabric-ca-client-config.yaml
Password: pqz
```

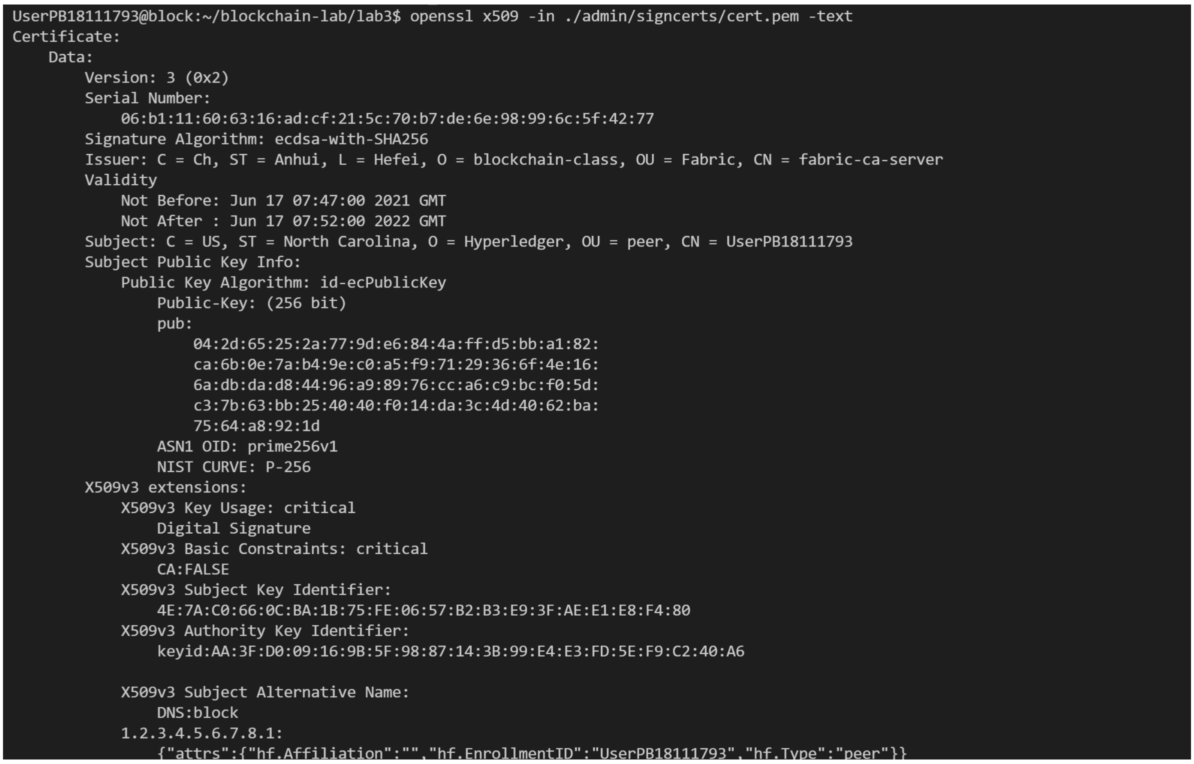


```
UserPB18111793@block:~/blockchain-lab/lab3$ fabric-ca-client enroll -u http://UserPB18111793:pqz@222.195.70.186:7054 --mspdir ./admin --
home ~/blockchain-lab/lab3
2021/06/17 07:52:19 [INFO] generating key: &{A:ecdsa S:256}
2021/06/17 07:52:19 [INFO] encoded CSR
2021/06/17 07:52:19 [INFO] Stored client certificate at /home/UserPB18111793/blockchain-lab/lab3/admin/signcerts/cert.pem
2021/06/17 07:52:19 [INFO] Stored root CA certificate at /home/UserPB18111793/blockchain-lab/lab3/admin/cacerts/222-195-70-186-7054.pem
2021/06/17 07:52:19 [INFO] Stored Issuer public key at /home/UserPB18111793/blockchain-lab/lab3/admin/IssuerPublicKey
2021/06/17 07:52:19 [INFO] Stored Issuer revocation public key at /home/UserPB18111793/blockchain-lab/lab3/admin/IssuerRevocationPublicK
ey
```

- 使用如下指令查看自己的证书信息

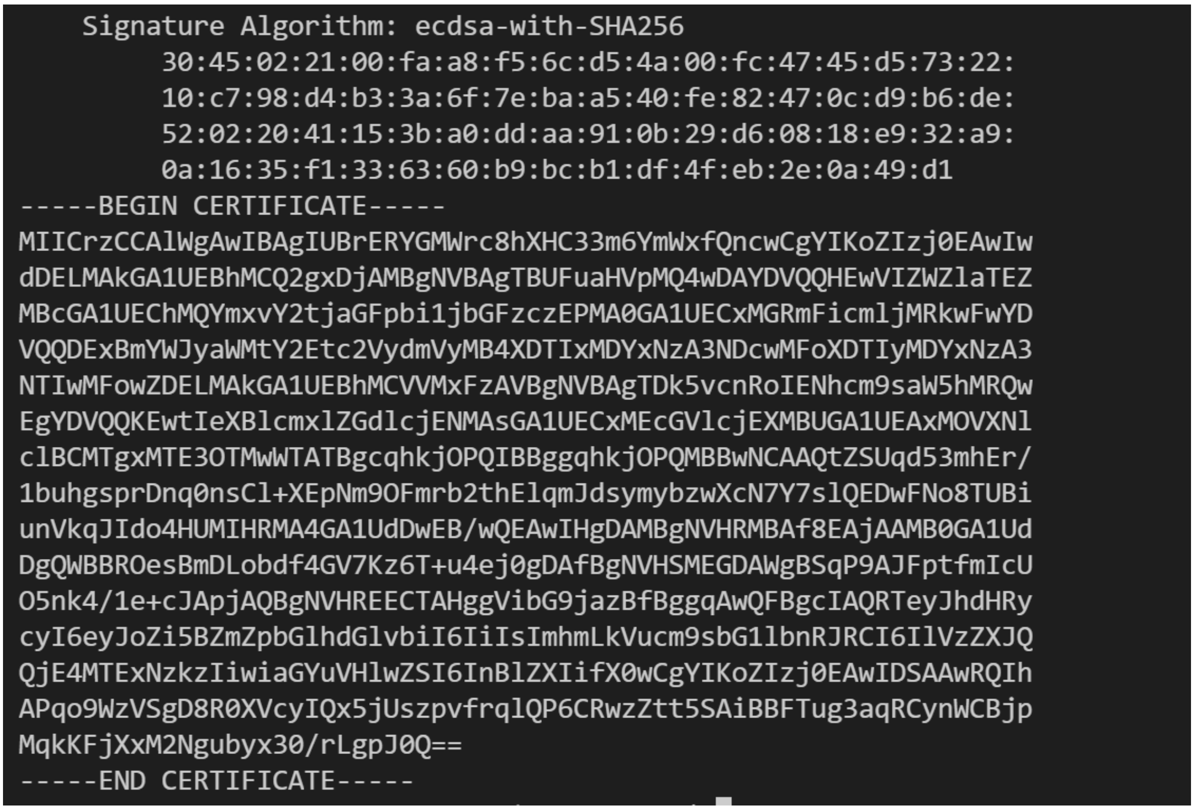
```
openssl x509 -in ./admin/signcerts/cert.pem -text
```

结果如下



```
UserPB18111793@block:~/blockchain-lab/lab3$ openssl x509 -in ./admin/signcerts/cert.pem -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            06:b1:11:60:63:16:ad:cf:21:5c:70:b7:de:6e:98:99:6c:5f:42:77
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: C = Ch, ST = Anhui, L = Hefei, O = blockchain-class, OU = Fabric, CN = fabric-ca-server
        Validity
            Not Before: Jun 17 07:47:00 2021 GMT
            Not After : Jun 17 07:52:00 2022 GMT
        Subject: C = US, ST = North Carolina, O = Hyperledger, OU = peer, CN = UserPB18111793
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:2d:65:25:2a:77:9d:e6:84:4a:ff:d5:bb:a1:82:
                    ca:6b:0e:7a:b4:9e:c0:a5:f9:71:29:36:6f:4e:16:
                    6a:db:da:d8:44:96:a9:89:76:cc:a6:c9:bc:f0:5d:
                    c3:7b:63:bb:25:40:40:f0:14:da:3c:4d:40:62:ba:
                    75:64:a8:92:1d
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Subject Key Identifier:
                4E:7A:C0:66:0C:BA:1B:75:FE:06:57:B2:B3:E9:3F:AE:E1:E8:F4:80
            X509v3 Authority Key Identifier:
                keyid:AA:3F:D0:09:16:9B:5F:98:87:14:3B:99:E4:E3:FD:5E:F9:C2:40:A6

            X509v3 Subject Alternative Name:
                DNS:block
            1.2.3.4.5.6.7.8.1:
                {"attrs":{"hf.Affiliation":"","hf.EnrollmentID":"UserPB18111793","hf.Type":"peer"}}
```



```
    Signature Algorithm: ecdsa-with-SHA256
         30:45:02:21:00:fa:a8:f5:6c:d5:4a:00:fc:47:45:d5:73:22:
         10:c7:98:d4:b3:3a:6f:7e:ba:a5:40:fe:82:47:0c:d9:b6:de:
         52:02:20:41:15:3b:a0:dd:aa:91:0b:29:d6:08:18:e9:32:a9:
         0a:16:35:f1:33:63:60:b9:bc:b1:df:4f:eb:2e:0a:49:d1
-----BEGIN CERTIFICATE-----
MIICrzCCAlWgAwIBAgIUBrERYGMWrc8hXHC33m6YmWxfQncwCgYIKoZIzj0EAwIw
dDELMAkGA1UEBhMCQ2gxDjAMBgNVBAgTBUFuaHVpMQ4wDAYDVQQHEwVIZWZlaTEZ
MBcGA1UEChMQYmxvY2tjaGFpbi1jbGFzczEPMA0GA1UECxMGRmFicmljMRkwFwYD
VQQDExBmYWJyaWMtY2Etc2VydmVyMB4XDTIxMDYxNzA3NDcwMFoXDTIyMDYxNzA3
NTIwMFowZDELMAkGA1UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRQw
EgYDVQQKEwtIeXBlcmxlZGdlcjENMAsGA1UECxMEcGVlcjEXMBUGA1UEAxMOVXNl
clBCMTgxMTE3OTMwWTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAAQtZSUqd53mhEr/
1buhgsprDnq0nsCl+XEpNm9OFmrb2thElqmJdsymybzwXcN7Y7slQEDwFNo8TUBi
unVkqJIdo4HUMIHRMA4GA1UdDwEB/wQEAwIHgDAMBgNVHRMBAf8EAjAAMB0GA1Ud
DgQWBBROesBmDLobdf4GV7Kz6T+u4ej0gDAfBgNVHSMEGDAWgBSqP9AJFptfmIcU
O5nk4/1e+cJApjAQBgNVHREECTAHggVibG9jazBfBggqAwQFBgcIAQRTeyJhdHRy
cyI6eyJoZi5BZmZpbGlhdGlvbiI6IiIsImhmLkVucm9sbG1lbnRJRCI6IlVzZXJQ
QjE4MTExNzkzIiwiaGYuVHlwZSI6InBlZXIifX0wCgYIKoZIzj0EAwIDSAAwRQIh
APqo9WzVSgD8R0XVcyIQx5jUszpvfrqlQP6CRwzZtt5SAiBBFTug3aqRCynWCBjp
MqkKFjXxM2Ngubyx30/rLgpJ0Q==
-----END CERTIFICATE-----
```

## 启动 Peer 节点

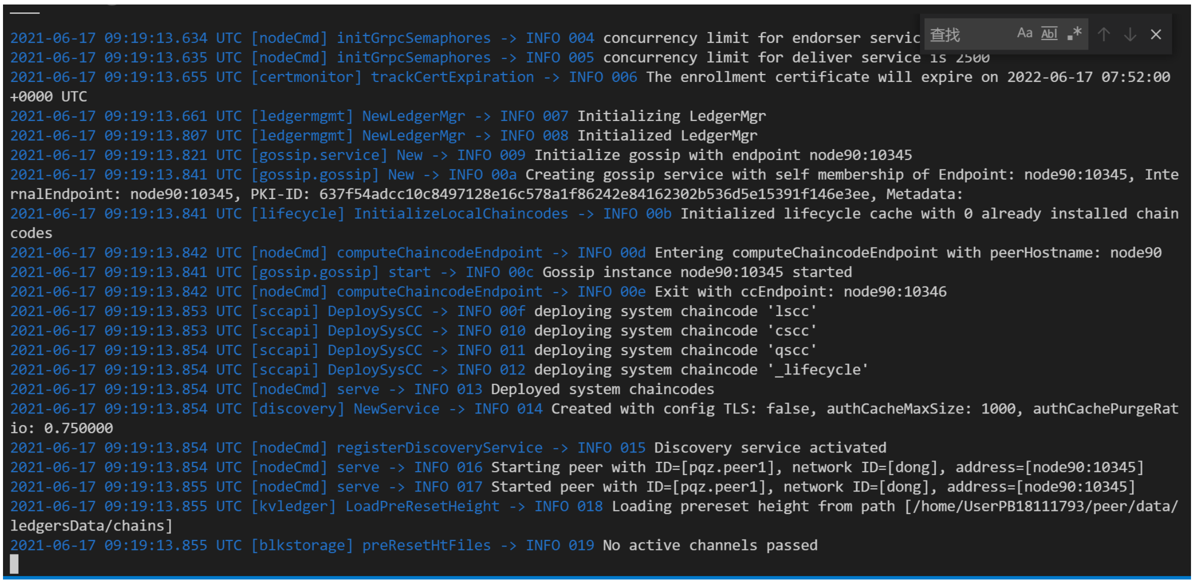
- 目标：在本地准备Peer节点启动所需要的文件，启动Peer节点
- 主要工作为
  - 构造msp文件夹
  - 配置好core.yaml文件
  - 设置环境变量（如果有需要）
  - 启动peer
- 主要修改了`core.yaml`文件，修改的主要部分如下（经过测试10345、10346端口是空闲端口）

```
peer:
    id: pqz.peer1
    listenAddress: node90:10345
    address: node90:10345
    gossip:
        bootstrap: node90:10345
        endpoint: node90:10345
        externalEndpoint: node90:10345
    fileSystemPath: /home/UserPB18111793/peer/data
    mspConfigPath: admin-msp (通过 scp 从222.195.70.186 服务器上拷贝)
    localMspId: Peer
    chaincodeListenAddress: node90:10346
```

并关闭profile、metrics等服务

```
metrics 下的 provider 设置为disabled  //关闭后就不需要管operations部分的配置了
profile 下的 enabled 设置为false
```

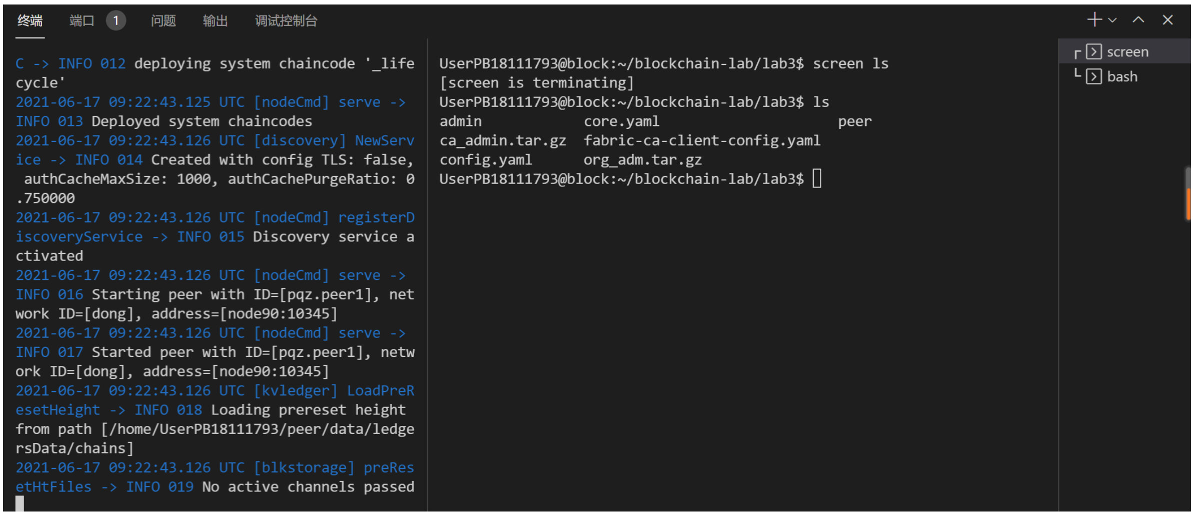
- 要注意的是：启动peer节点时需要使用peer类型证书，加入通道时需要使用admin类型证书。
- 启动 peer 结果如下



```
2021-06-17 09:19:13.634 UTC [nodeCmd] initGrpcSemaphores -> INFO 004 concurrency limit for endorser servic
2021-06-17 09:19:13.635 UTC [nodeCmd] initGrpcSemaphores -> INFO 005 concurrency limit for deliver service is 2500
2021-06-17 09:19:13.655 UTC [certmonitor] trackCertExpiration -> INFO 006 The enrollment certificate will expire on 2022-06-17 07:52:00
+0000 UTC
2021-06-17 09:19:13.661 UTC [ledgermgmt] NewLedgerMgr -> INFO 007 Initializing LedgerMgr
2021-06-17 09:19:13.807 UTC [ledgermgmt] NewLedgerMgr -> INFO 008 Initialized LedgerMgr
2021-06-17 09:19:13.821 UTC [gossip.service] New -> INFO 009 Initialize gossip with endpoint node90:10345
2021-06-17 09:19:13.841 UTC [gossip.gossip] New -> INFO 00a Creating gossip service with self membership of Endpoint: node90:10345, Inte
rnalEndpoint: node90:10345, PKI-ID: 637f54adcc10c8497128e16c578a1f86242e84162302b536d5e15391f146e3ee, Metadata:
2021-06-17 09:19:13.841 UTC [lifecycle] InitializeLocalChaincodes -> INFO 00b Initialized lifecycle cache with 0 already installed chain
codes
2021-06-17 09:19:13.842 UTC [nodeCmd] computeChaincodeEndpoint -> INFO 00d Entering computeChaincodeEndpoint with peerHostname: node90
2021-06-17 09:19:13.841 UTC [gossip.gossip] start -> INFO 00c Gossip instance node90:10345 started
2021-06-17 09:19:13.842 UTC [nodeCmd] computeChaincodeEndpoint -> INFO 00e Exit with ccEndpoint: node90:10346
2021-06-17 09:19:13.853 UTC [sccapi] DeploySysCC -> INFO 00f deploying system chaincode 'lscc'
2021-06-17 09:19:13.853 UTC [sccapi] DeploySysCC -> INFO 010 deploying system chaincode 'cscc'
2021-06-17 09:19:13.854 UTC [sccapi] DeploySysCC -> INFO 011 deploying system chaincode 'qscc'
2021-06-17 09:19:13.854 UTC [sccapi] DeploySysCC -> INFO 012 deploying system chaincode '_lifecycle'
2021-06-17 09:19:13.854 UTC [nodeCmd] serve -> INFO 013 Deployed system chaincodes
2021-06-17 09:19:13.854 UTC [discovery] NewService -> INFO 014 Created with config TLS: false, authCacheMaxSize: 1000, authCachePurgeRat
io: 0.750000
2021-06-17 09:19:13.854 UTC [nodeCmd] registerDiscoveryService -> INFO 015 Discovery service activated
2021-06-17 09:19:13.854 UTC [nodeCmd] serve -> INFO 016 Starting peer with ID=[pqz.peer1], network ID=[dong], address=[node90:10345]
2021-06-17 09:19:13.855 UTC [nodeCmd] serve -> INFO 017 Started peer with ID=[pqz.peer1], network ID=[dong], address=[node90:10345]
2021-06-17 09:19:13.855 UTC [kvledger] LoadPreResetHeight -> INFO 018 Loading prereset height from path [/home/UserPB18111793/peer/data/
ledgersData/chains]
2021-06-17 09:19:13.855 UTC [blkstorage] preResetHtFiles -> INFO 019 No active channels passed
```

- 使用如下命令启动额外的shell后，再执行peer node start

```
screen -S pqz
```

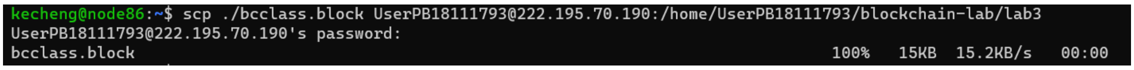


```
C -> INFO 012 deploying system chaincode '_life
cycle'
2021-06-17 09:22:43.125 UTC [nodeCmd] serve ->
INFO 013 Deployed system chaincodes
2021-06-17 09:22:43.126 UTC [discovery] NewServ
ice -> INFO 014 Created with config TLS: false,
 authCacheMaxSize: 1000, authCachePurgeRatio: 0
.750000
2021-06-17 09:22:43.126 UTC [nodeCmd] registerD
iscoveryService -> INFO 015 Discovery service a
ctivated
2021-06-17 09:22:43.126 UTC [nodeCmd] serve ->
INFO 016 Starting peer with ID=[pqz.peer1], net
work ID=[dong], address=[node90:10345]
2021-06-17 09:22:43.126 UTC [nodeCmd] serve ->
INFO 017 Started peer with ID=[pqz.peer1], netw
ork ID=[dong], address=[node90:10345]
2021-06-17 09:22:43.126 UTC [kvledger] LoadPreR
esetHeight -> INFO 018 Loading prereset height
from path [/home/UserPB18111793/peer/data/ledge
rsData/chains]
2021-06-17 09:22:43.126 UTC [blkstorage] preRes
etHtFiles -> INFO 019 No active channels passed
```

```
UserPB18111793@block:~/blockchain-lab/lab3$ screen ls
[screen is terminating]
UserPB18111793@block:~/blockchain-lab/lab3$ ls
admin             core.yaml                        peer
ca_admin.tar.gz   fabric-ca-client-config.yaml
config.yaml       org_adm.tar.gz
UserPB18111793@block:~/blockchain-lab/lab3$
```

## 加入通道

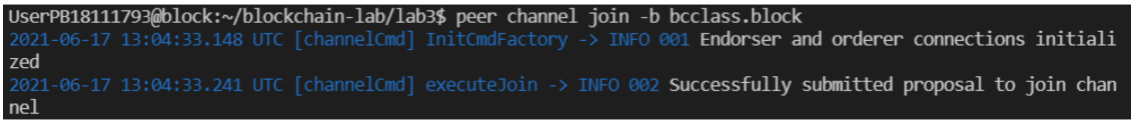
- 目标：将自己搭建的Peer节点加入通道bcclass
- 先使用 scp 命令从 222.195.70.186 服务器上拷贝`bcclass.block`



```
kecheng@node86:~$ scp ./bcclass.block UserPB18111793@222.195.70.190:/home/UserPB18111793/blockchain-lab/lab3
UserPB18111793@222.195.70.190's password:
bcclass.block                                                                   100%   15KB  15.2KB/s   00:00
```

- 使用如下命令将 peer 节点加入通道

```
peer channel join -b bcclass.block
```

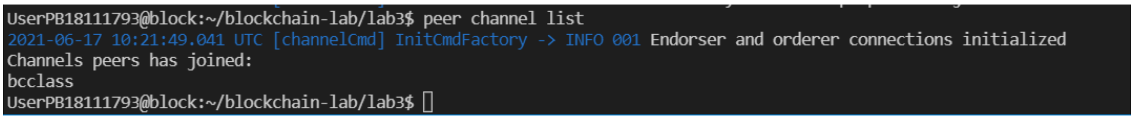


```
UserPB18111793@block:~/blockchain-lab/lab3$ peer channel join -b bcclass.block
2021-06-17 13:04:33.148 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initiali
zed
2021-06-17 13:04:33.241 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join chan
nel
```

- 使用如下命令查看peer加入的通道名称

```
peer channel list
```

可以看到结果正确



```
UserPB18111793@block:~/blockchain-lab/lab3$ peer channel list
2021-06-17 10:21:49.041 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
Channels peers has joined:
bcclass
UserPB18111793@block:~/blockchain-lab/lab3$ 
```

# 实验总结

通过本次实验，我对 Fabric 的部署有了更加深刻的了解。通过使用 Fabric 搭建一个 peer 节点，并用这个 peer 节点加入已经创建的通道之中，让我对 Fabric 中通道的具体建立过程有了体会。通道允许一组参与者创建各自的交易账本。 对于某些网络而言， 这是一个特别重要的选择。 这些网络中， 一些参与者可能是竞争对手， 并且不希望他们做出的每笔交易都被每个参与者知晓， 例如，他们只向某些参与者提供的特殊价格， 而其他人不是。 如果两个参与者组成一个通道， 那么只有这两个参与者拥有该通道的账本副本， 而其他参与者没有。 同时这次实验也让我对客户端-服务端架构的运行流程有了更加深刻的体会，例如使用CA服务器注册身份、使用scp从服务器拷贝数据等等。