Peer 节点配置与启动

Peer 节点的设置大致分为四步
1） 构建 peer 节点的 msp
2） 修改 peer 节点的 core.yaml 配置文件
3） 设置 FABRIC_CFG_PATH 环境变量指向 core.yaml 所在的文件夹（记住是文件夹不是 core.yaml 文件）
4） peer node start 启动 peer 节点

Peer 有五条子命令，其中跟节点启动相关的子命令是 peer node。其下又包含了三条子命令（operation）：
start： 启动节点
reset： 将所有的通道会重置为初始块。
rollback：将一个通道的区块链回滚到特定号码的块。
其中 reset 和 rollback 指令需要在离线状态下执行。

1）peer 节点的 MSP 内容如下：
admincerts\ admin 的证书
cacerts\ root ca 的自签名证书
intermediatecerts\ 中间 ca 的自签名证书
keystore\ 私钥
signcerts\ 证书
tlscacerts\ tls root ca 的证书
tlsintermediatecerts\ tls 中间 ca 的证书
config.yaml OU 配置文件
注：在官方文档中说明了 msp 在 1.4.3 版本以前包含了 admincerts 这个文件夹，里面存放的是 org 的 admin 的身份证书。然而在 1.4.3 以上版本，这个文件夹移除了，身份类别的判断是通过 ou 进行的，ou 的设置在 config.yaml 里面。内容如下：

```
NodeOUs:
  Enable: true
  ClientOUIdentifier:
    Certificate: intermediatecerts/localnode3-7054.pem
    OrganizationalUnitIdentifier: client
  PeerOUIdentifier:
    Certificate: intermediatecerts/localnode3-7054.pem
    OrganizationalUnitIdentifier: peer
  AdminOUIdentifier:
    Certificate: intermediatecerts/localnode3-7054.pem
    OrganizationalUnitIdentifier: admin
  OrdererOUIdentifier:
    Certificate: intermediatecerts/localnode3-7054.pem
    OrganizationalUnitIdentifier: orderer
```

利用 MSP 进行身份的判定：
1. 通过 config.yaml 中的 Certificate 中指定的 ca 证书与 signcerts 中的证书进行验证，判断 peer 节点证书的有效性
2. 然后提取证书中的 subject 的 OU 字段
3. 与 config.yaml 中的 Org*fier 匹配，来判断身份类别

2）修改 peer 节点的 core.yaml 配置文件

Peer 在逻辑上属于某一组织，在功能上需要与同组织的节点通过 gossip 协议进行交互，并且保存链数据等。

那么最基本的配置应该包括：

本节点的身份信息：

id：本节点的 id

mspConfigPath：　节点的 msp

节点所在组织标识

localMspId: 节点所在组织的 Id，应与通道中该组织的 ID 一致。

与组织中的节点或组织外的节点通信的部分：

listenAddress：

address：　这个地址是暴露给组织内部的节点，与 gossip 中的 endpoint 功能重复 ，设置成相同值

gossip

bootstrap：组织中其他节点的地址

endpoint：暴露给组织中其他节点的地址

externalEndpoint：暴露给组织外的节点地址

区块联的数据存放路径：

fileSystemPath

3）FABRIC_CFG_PATH 指向 core.yaml 所在的文件夹

4）node~:$ peer node start　　#启动 peer 节点