# 环境准备

## Fabric和fabric-samples的版本为1.1.0

git checkout -b release-1.1 origin/release-1.1

## 设置fabric-samples的docker images的版本为1.1.0

进入fabric-samples/first-network目录，修改.env文件IMAGE\_TAG=1.1.0

## 安装编译依赖libtool和libtdhl-dev

sudo apt install libtool libltdl-dev

## 安装 fabric-ca-server 和 fabric-ca-client

go get -u github.com/hyperledger/fabric-ca/cmd/...

以上命令是执行的流程是git上拉取fabric-ca，之后进行abric-ca-server 和 fabric-ca-client的安装。安如果在执行命令之后提示，安装失败。则手动安装。

也可进行分步骤进行安装，

先拉取fabric-ca：

git clone <https://github.com/hyperledger/fabric-ca.git>

再将版本切换到1.1.0

git checkout -b release-1.1 origin/release-1.1

切换到fabric-ca源码目录下

进入目录执行以下命令手动安装fabric-ca-server 和 fabric-ca-client

先编译：

make fabric-ca-server

make fabric-ca-client

再安装：

  go get -v -ldflags "-X github.com/hyperledger/fabric-ca/lib/metadata.Version=1.1.1" github.com/hyperledger/fabric-ca/cmd/...

go get -u github.com/hyperledger/fabric-ca/cmd/...

输入命令检验安装是否成功：如果出现帮助提示信息表示安装成功。

fabric-ca-server

fabric-ca-client

# 生成证书文件

## 启动 fabric-ca-server

使用默认配置启动fabric-ca-server：

fabric-ca-server start -b admin:adminpw

因为下面操作需要删除联盟和删除用户的需求，需要用下面的方式启动：

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

## 生成ca-server管理员凭证

生成凭证的路径

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

登录管理员用户并生成凭证的操作，admin:adminp是用户名和用户密码

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

## 创建联盟

查看现有的联盟信息

fabric-ca-client affiliation list

将现有的两个联盟删除

fabric-ca-client affiliation remove --force org1

fabric-ca-client affiliation remove --force org2

创建所需联盟

因为联盟是有层级的，所以需要逐级创建

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

## 为每个组织生成证书文件

### 新建一个文件夹用于存储所有证书文件

mkdir -p /root/fabric-ca-files

### 生成example.com的证书

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

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

命令执行结束后，会在fabric-ca-files/example.com/msp得到文件：

example.com/msp/

-- cacerts

`-- localhost-7054.pem

-- intermediatecerts

`-- localhost-7054.pem

-- keystore

-- signcerts

注意通过getcacert得到msp目录中只有CA证书，而且这里没有使用中间CA，fabric-ca-files/example.com/msp/intermediatecerts/localhost-7054.pem是一个空文件。

### 生成org1.example.com和org2.example.com的根证书

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

这里是用getcacert为每个组织准备需要的ca文件，在生成创始块的时候会用到。

如果在你的环境中，各个组件域名的证书，是由第三方CA签署的，就将第三方CA的根证书添加到msp/tlscacerts目录中。

组织的msp目录中，包含都是CA根证书，分别是TLS加密的根证书，和用于身份验证的根证书。另外还需要admin用户的证书，后面的操作中会添加。

### 注册example.com的管理员Admin@example.com

可以直接用命令行也可以通过修改fabric-ca-client-config.yaml文件来修改参数，以下统一通过修改fabric-ca-client-config.yaml来的方式来修改参数然后进行注册。

将fabric-ca-files/admin/fabric-ca-client-config.yaml其中的id部分修改为：

id:

name: Admin@example.com

type: client

affiliation: com.example

maxenrollments: 0

attributes:

- name: hf.Registrar.Roles

value: client,orderer,peer,user

- name: hf.Registrar.DelegateRoles

value: client,orderer,peer,user

- name: hf.Registrar.Attributes

value: "\*"

- name: hf.GenCRL

value: true

- name: hf.Revoker

value: true

- name: hf.AffiliationMgr

value: true

- name: hf.IntermediateCA

value: true

- name: role

value: admin

ecert: true

参数的的具体作用如下：

--id.name Admin@example.com //用户名

--id.type client //类型为client

--id.affiliation "com.example" //权利访问

hf.Registrar.Roles=client,orderer,peer,user //能够管理的用户类型

hf.Registrar.DelegateRoles=client,orderer,peer,user //可以授权给子用户管理的用户类型

hf.Registrar.Attributes=\* //可以为子用户设置所有属性

hf.GenCRL=true //可以生成撤销证书列表

hf.Revoker=true //可以撤销用户

hf.AffiliationMgr=true //能够管理联盟

hf.IntermediateCA=true //可以作为中间CA

role=admin:ecert //自定义属性

执行注册用户的命令，不指定配置文件的情况下，会默认使用/fabric-ca-client-config.yaml为配置文件。

fabric-ca-client register --id.secret=password

id.secret=password指定用户登录密码为password

后面要登录Admin@example.com然后生成凭证

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

-H 是指定生成凭证的文件路径

命令运行成功之后，此时在ca服务器上登录的是Admin@example.com用户，这时候可以用Admin@example.com的身份查看联盟：

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

将Admin@example.com的证书复制到example.com/msp/admincerts/中：

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/

只有这样，才能具备管理员权限。

注：fabric-ca-client register命令默认使用的配置文件是fabric-ca-client-config.yaml。使用-H可以指定注册时的目录，目录下必须有msp文件夹和fabric-ca-client-config.yaml文件。msp是fabric-ca-server管理员登录时生成的凭证。

fabric-ca-client-config.yaml文件位于fabric-ca-server管理员凭证目录下：$HOME/fabric-ca/clients/admin。

### 注册org1.example.com和org2.example.com管理员

类似注册example.com，需要先修改fabric-ca-client-config.yaml配置文件，然后进行注册，之后进行登录和生成凭证（证书）文件，最后将管理员证书复制到组织的admincerts文件夹。

具体操作如下：

为org1.example.com修改fabric-ca-client-config.yaml文件

id:

name: Admin@org1.example.com

type: client

affiliation: com.example.org1

maxenrollments: 0

attributes:

- name: hf.Registrar.Roles

value: client,orderer,peer,user

- name: hf.Registrar.DelegateRoles

value: client,orderer,peer,user

- name: hf.Registrar.Attributes

value: "\*"

- name: hf.GenCRL

value: true

- name: hf.Revoker

value: true

- name: hf.AffiliationMgr

value: true

- name: hf.IntermediateCA

value: true

- name: role

value: admin

ecert: true

登录生成凭证（证书）

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

fabric-ca-client register --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

[复制管理员证书，因为在操作fabric网络时，需要在Admin@org1.example.com目录使用admincerts目录中的管理员证书，所以也要复制一份。](mailto:复制管理员证书，因为在操作fabric网络时，需要在Admin@org1.example.com目录使用admincerts目录中的管理员证书，所以也要复制一份。)

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/

为org2.example.com修改fabric-ca-client-config.yaml文件

id:

name: Admin@org2.example.com

type: client

affiliation: com.example.org2

maxenrollments: 0

attributes:

- name: hf.Registrar.Roles

value: client,orderer,peer,user

- name: hf.Registrar.DelegateRoles

value: client,orderer,peer,user

- name: hf.Registrar.Attributes

value: "\*"

- name: hf.GenCRL

value: true

- name: hf.Revoker

value: true

- name: hf.AffiliationMgr

value: true

- name: hf.IntermediateCA

value: true

- name: role

value: admin

ecert: true

登录生成凭证（证书）

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

fabric-ca-client register --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

[复制管理员证书。](mailto:复制管理员证书，因为在操作fabric网络时，需要在Admin@org1.example.com目录使用admincerts目录中的管理员证书，所以也要复制一份。)

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/

### 生成orderer的证书

具体操作如下：

为orderer修改fabric-ca-client-config.yaml文件

id:

name: orderer.example.com

type: orderer

affiliation: com.example

maxenrollments: 0

attributes:

- name: role

value: orderer

ecert: true

注册然后登录生成凭证（证书）

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

fabric-ca-client register --id.secret=password

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

[复制管理员证书到orderer的MSP的admincerts文件夹。](mailto:复制管理员证书，因为在操作fabric网络时，需要在Admin@org1.example.com目录使用admincerts目录中的管理员证书，所以也要复制一份。)

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/

### 生成peer的证书

**peer0.org1.example.com**

修改fabric-ca-client-config.yaml文件

id:

name: peer0.org1.example.com

type: peer

affiliation: com.example.org1

maxenrollments: 0

attributes:

- name: role

value: peer

ecert: true

注册然后登录生成凭证（证书）

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

fabric-ca-client register --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

[复制管理员证书。](mailto:复制管理员证书，因为在操作fabric网络时，需要在Admin@org1.example.com目录使用admincerts目录中的管理员证书，所以也要复制一份。)

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

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/

**peer1.org1.example.com**

修改fabric-ca-client-config.yaml文件

id:

name: peer1.org1.example.com

type: peer

affiliation: com.example.org1

maxenrollments: 0

attributes:

- name: role

value: peer

ecert: true

注册然后登录生成凭证（证书）

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

fabric-ca-client register --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

[复制管理员证书。](mailto:复制管理员证书，因为在操作fabric网络时，需要在Admin@org1.example.com目录使用admincerts目录中的管理员证书，所以也要复制一份。)

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/

**peer0.org2.example.com**

修改fabric-ca-client-config.yaml文件

id:

name: peer0.org2.example.com

type: peer

affiliation: com.example.org2

maxenrollments: 0

attributes:

- name: role

value: peer

ecert: true

注册然后登录生成凭证（证书）

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

fabric-ca-client register --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

[复制管理员证书。](mailto:复制管理员证书，因为在操作fabric网络时，需要在Admin@org1.example.com目录使用admincerts目录中的管理员证书，所以也要复制一份。)

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/

peer1.org2.example.com

修改fabric-ca-client-config.yaml文件

id:

name: peer1.org2.example.com

type: peer

affiliation: com.example.org2

maxenrollments: 0

attributes:

- name: role

value: peer

ecert: true

注册然后登录生成凭证（证书）

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

fabric-ca-client register --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

[复制管理员证书。](mailto:复制管理员证书，因为在操作fabric网络时，需要在Admin@org1.example.com目录使用admincerts目录中的管理员证书，所以也要复制一份。)

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

# 使用新证书重新部署Fabric网络

因为暂时还不会生成相关TLS证书文件，但是在网络中又需要使用TLS文件，所以这里先使用Cryptogen生成一整套证书文件（包含TLS证书和其他证书），然后再使用Fabric-ca生成的证书文件（非TLS证书）替换相应的证书文件。所以以下的操作是详细的将Fabric-ca生成的文件替换到Cryptogen生成的证书文件内。这里没有使用中间CA，但在Fabric-ca生成的证书文件时有intermediatecerts文件夹（中间CA文件夹），并且证书内容为空。因此生成genesisblock的时候会有警告提示，如果不想看到提示，将intermediatecerts文件夹删除即可。

## 替换ordererOrganizations\example.com中的文件

### 替换ca文件

将fabric-ca-files\example.com\msp\cacerts中的文件复制替换ca下的文件

### 替换msp文件

将fabric-ca-files\example.com\msp中的文件复制替换到msp文件夹下。

### 替换users\Admin@example.com\msp文件

将fabric-ca-files\example.com\admin\msp中的文件复制替换到users\Admin@example.com\msp文件夹下。

### 替换orderers\orderer.example.com\msp文件

将fabric-ca-files\example.com\orderer\msp中的文件复制替换到orderers\orderer.example.com\msp文件夹下。

## 替换peerOrganizations\org1.example.com中的文件

### 替换ca文件

将fabric-ca-files\org1.example.com\msp\cacerts中的文件复制替换ca下的文件。

### 替换msp文件

将fabric-ca-files\org1.example.com\msp中的文件复制替换到msp文件夹下。

### 替换users\Admin@org1.example.com文件

将fabric-ca-files\org1.example.com\admin\msp中的文件复制替换到users\Admin@org1.example.com\msp文件夹下。

### 替换peers\peer0.org1.example.com\msp文件

将fabric-ca-files\org1.example.com\peer0\msp中的文件复制替换到peers\peer0.org1.example.com\msp文件夹下。

### 替换peers\peer1.org1.example.com\msp文件

将fabric-ca-files\org1.example.com\peer1\msp中的文件复制替换到peers\peer1.org1.example.com\msp文件夹下。

## 替换peerOrganizations\org2.example.com中的文件

替换方式同peerOrganizations\org1.example.com，分为：替换ca文件、替换msp文件、替换users\Admin@org1.example.com文件、替换peers\peer0.org1.example.com\msp文件、替换peers\peer1.org1.example.com\msp文件。

## 用新证书生成交易配置文件

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

## 启动网络，执行网络节点操作

参照《搭建第一个fabric网络》，若执行报错TLS问题，可以将docker-compose-cli.yaml文件和base文件夹里两个yaml文件的TLS关闭。CORE\_PEER\_TLS\_ENABLED为false。

## 执行chaincode交易

参照《搭建第一个fabric网络》