区块链实验三实验报告

PB19071405 王昊元

2022 年 06 月 24 日

1 实验目的及要求

1.1 实验目的

- 了解 fabric 上的基本配置
- 在 fabric 网络中添加 peer，并加入网络
- 了解 fabric 上的基本证书和网络

1.2 实验要求

1. 使用 CA 服务器注册身份，并获得 CA 服务器颁发的身份证书
2. 在本地准备 Peer 节点启动所需要的文件，启动 Peer 节点
3. 将自己搭建的 Peer 节点加入通道 mychannel

2 实验平台

- OpenStack 平台
- Ubuntu 22.04 LTS (GNU/Linux 5.15.0-30-generic x86_64)

3 实验步骤

3.1 注册身份，并且获取证书

注册部分按照实验文档进行环境变量配置及命令的运行，脚本如下：

```
to_tls_certfiles="/home/ubuntu/fabric/tls-ca/crypto/tls-ca-cert.pem"
to_tls_ca_admin="/home/ubuntu/fabric/tls-ca/admin"
to_hold_tlsmsp="/home/ubuntu/fabric/org1/peer2/tls-msp"
PEERNAME="PB19071405"
PEERSECRET="12345678"
HOSTNAME=${PEERNAME}

export FABRIC_CA_CLIENT_TLS_CERTFILES=${to_tls_certfiles}

export FABRIC_CA_CLIENT_HOME=${to_tls_ca_admin}

# 注册身份
fabric-ca-client register --id.name ${PEERNAME} --id.secret ${PEERSECRET} --id.
    type peer -u https://172.16.4.35:7052

# 获得对应的tls_msp
# 存储tls_msp的位置
export FABRIC_CA_CLIENT_MSPDIR=${to_hold_tlsmsp}
fabric-ca-client enroll -u https://${PEERNAME}:${PEERSECRET}@172.16.4.35:7052
    --enrollment.profile tls --csr.hosts ${HOSTNAME}

# 注册成功后，可以把keystore下的文件存储为证书. 当然，你也可以手动将其重新命名.
# mv tls-msp/keystore/*_sk tls-msp/keystore/key.pem
# mv msp/keystore/*_sk msp/keystore/key.pem
mv ${to_hold_tlsmsp}/keystore/*_sk ${to_hold_tlsmsp}/keystore/key.pem

unset FABRIC_CA_CLIENT_TLS_CERTFILES
unset FABRIC_CA_CLIENT_HOME
unset FABRIC_CA_CLIENT_MSPDIR


to_tls_certfiles="/home/ubuntu/fabric/org1/ca/crypto/ca-cert.pem"
to_org_ca_admin="/home/ubuntu/fabric/org1/ca/admin"
to_hold_msp="/home/ubuntu/fabric/org1/peer2/msp"

export FABRIC_CA_CLIENT_TLS_CERTFILES=${to_tls_certfiles}

export FABRIC_CA_CLIENT_HOME=${to_org_ca_admin}

```

```
fabric-ca-client register --id.name ${PEERNAME} --id.secret ${PEERSECRET} --id.type peer -u https://172.16.4.35:7054 --csr.hosts ${HOSTNAME}

# 存储msp的位置
export FABRIC_CA_CLIENT_MSPDIR=${to_hold_msp}
# 获得对应的msp
fabric-ca-client enroll -u https://${PEERNAME}:${PEERSECRET}@172.16.4.35:7054

mv ${to_hold_msp}/keystore/*_sk ${to_hold_msp}/keystore/key.pem

unset FABRIC_CA_CLIENT_TLS_CERTFILES
unset FABRIC_CA_CLIENT_HOME
unset FABRIC_CA_CLIENT_MSPDIR

unset PEERNAME
unset PEERSECRET
unset HOSTNAME
```

3.2 启动 Peer 节点

3.2.1 创建配置文件

先在相应的文件夹下创建 `config.yaml`，不同的文件所对应的证书不同，修改为对应的证书即可。例如在 `admin/msp/config.yaml` 中为

```
NodeOUs:
  Enable: true
  ClientOUIdentifier:
    Certificate: cacerts/org1-ca-cert.pem
    OrganizationalUnitIdentifier: client
  PeerOUIdentifier:
    Certificate: cacerts/org1-ca-cert.pem
    OrganizationalUnitIdentifier: peer
  AdminOUIdentifier:
    Certificate: cacerts/org1-ca-cert.pem
    OrganizationalUnitIdentifier: admin
  OrdererOUIdentifier:
    Certificate: cacerts/org1-ca-cert.pem
    OrganizationalUnitIdentifier: orderer
```

3.2.2 配置 docker

在/home/ubuntu/fabric/org1/peer2 文件夹下创建 docker-compose.yaml 文件，修改对应的信息为自己的信息，如 HOSTNAME 等，PEER 端口为 7051，GOSSIP 端口为 7052。因命名格式与助教一致（主要反映在 peer 的命名上），所以其他信息不需要改动，核实一下签名证书等命名即可。

在/home/ubuntu/fabric/org1/cli 文件夹下创建 docker-compose.yaml 文件，并修改信息。

3.2.3 启动 docker container

分别在/home/ubuntu/fabric/org1/peer2 和/home/ubuntu/fabric/org1/cli 下通过命令 docker-compose up -d 启动 peer 和 cli 对应的容器（好像实验文档上没有说单独启动 cli 的容器，我配置完顺手就执行了，但好像也不影响，从 docker 容器的角度看，都可以通过 docker exec -it cli-org1 /bin/bash 命令在 container 中的运行 bash）。

3.3 添加通道

由于 docker mount 的路径中不包括 mychannel.block 的路径，所以我们需要将 mychannel.block 复制到/home/ubuntu/fabric/org1/peer2（被挂载的任何路径应该都可以）下

执行 docker exec -it cli-org1 /bin/bash 命令进入 cli 容器中的 bash，分别执行 export CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/org1/admin/msp 和 peer channel join -b mychannel.block （我理解的 mychannel.block 应该对应其路径，所以我先修改工作目录到相应位置才执行的），最后执行 peer channel list 查看是否添加成功。

4 实验结果

第一次启动 peer 之后的 log 截图如下：

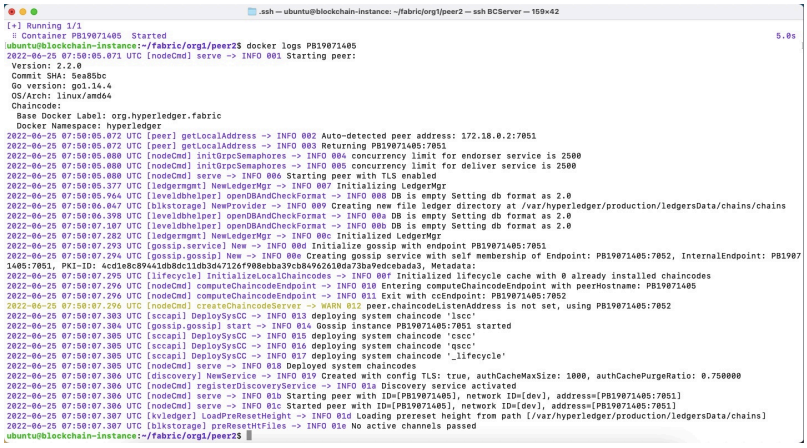

图 1: `docker logs PB19071405` 的结果（未添加 `channel`）

添加 channel 后在 `cli` 中执行 `peer channel list` 的结果如下：

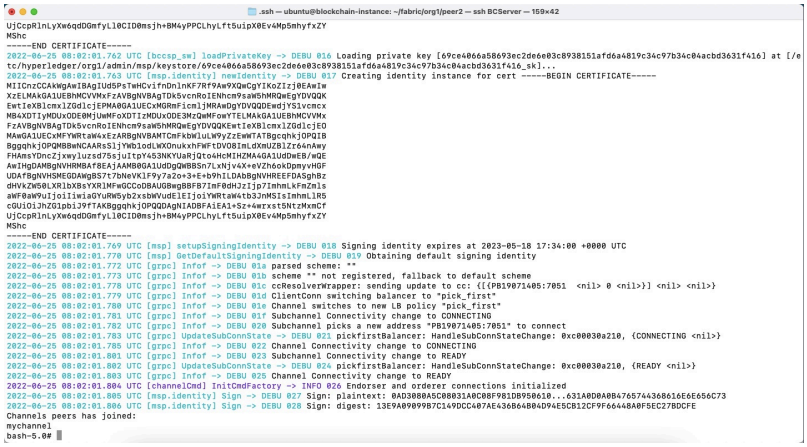

图 2: 添加通道后 `peer channel list` 的结果

再次查看 peer 容器的 log，如下：

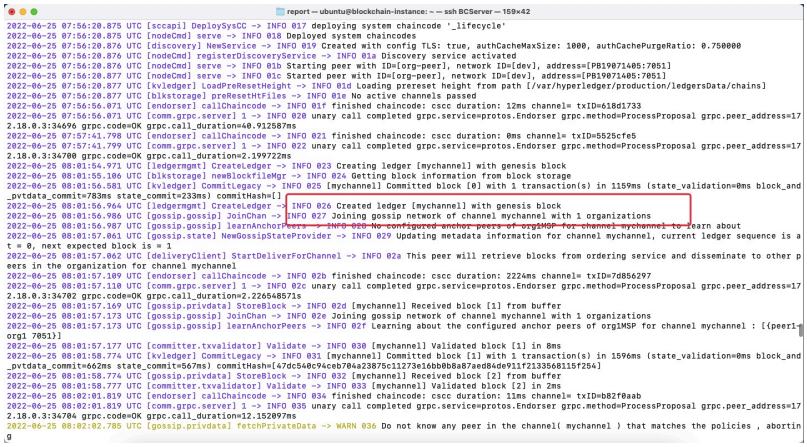

图 3: 添加通道后 peer 容器的日志

图3为之后截图，可以看到下面已经产生了链上广播的一些信息（如果我没理解错的话），从另一个方面验证了结果的正确性。

5 实验总结及反馈

5.1 实验总结

- 自己实际操作了 Fabric 的注册 Peer、通过 Docker 启动再添加通道的过程，对 Fabric 的理解有了更深的理解。
- 由于实验是在考试之后完成的，对于“考完就忘”的现状来说，也起到了很大的巩固知识的作用。

5.2 实验反馈

可能由于新课程新实验的缘故，实验文档有点杂乱的感觉。但正因为是新课程新实验，我才决定反馈一些信息给助教和老师。

- 提及比较细小的流程。例如对于 `peer2` 目录的创建，是需要我们手动创建的，助教没有在文档中提及这一点，一直让我认为我注册的操作是有误的。
- 上一点提到的例子或许是因为跟助教的实际操作有些许不同，导致这个目录必须我手动创建，但这也意味着可能文档没有考虑全面（毕竟很多人做实验不可能一模一样，除非文档将每一步在哪里输什么看到什么像博客一样写出来，不过那样实验就失去了一定的意

义），这也很正常，只是希望助教可以从我们不同人的反馈中不断完善它。例如，在注册所提供的命令重命名部分，助教默认了我们所指定的存储位置是在当前文件夹下的 `msp` 和 `tls-msp`（另外可能写两个是包括了两个部分的重命名，而非考虑两种命名情况，算是一个小失误吧），我在自己的脚本中引用了 `to_hold_tlsmsp` 变量来规避这一问题。

- 除此之外可能就是我认为的文档的一些小失误/小问题：
  - 两次注册命令里变量有修改，但是说明部分没有修改
  - yaml 文件注释使用双斜杠
  - 对 docker 文件挂载路径对应部分的说明其实可以用文字简单说明一下，我理解那个等号理解了很久（也有可能是因为我很久没用 docker 而又不愿意看文档导致对 docker 不熟悉所以没反应过来），对于不熟悉的同学可能理解上有困难。例如 “Docker 中 xx 路径” 会被挂载/对应 “磁盘中 xxx 路径”（主要是理解了很久那个等号）

此外，如果实验报告中体现出来我哪里理解有误，还希望助教指出。
最后还很感谢助教提供了详细的操作命令及配置文件模板，帮助了我很多。