

Order4

add

1. Orderer1

传输相关organizations

```
1 cd /data/fabric/fabric-orderer1/test-network
2 scp -r channel-artifacts/ organizations/
   orderer4.example.com:/data/fabric/fabric-orderer4/test-network
```

2. Orderer4

2.1 加入通道

```
1 [root@orderer4 ~]# cd /data/fabric/fabric-orderer4/test-network
2 ./network.sh up createChannel -c carbonchain
```

2.2 设置环境变量

```
1 export channel_name=carbonchain
2 export PATH=${PWD}/../bin:$PATH
3 export FABRIC_CFG_PATH=$PWD/../config/
4 export ORDERER_CONTAINER=orderer.example.com:7050
5 export CORE_PEER_TLS_ENABLED=true
6 export CORE_PEER_LOCALMSPID=OrdererMSP
7 export
   CORE_PEER_MSPCONFIGPATH=$(pwd)/organizations/ordererOrganizations/example.com/users/Admin\@example.com/msp/
8 export
   CORE_PEER_TLS_ROOTCERT_FILE=$(pwd)/organizations/ordererOrganizations/example.com/orderers/orderer4.example.com/tls/ca.crt
9 export
   ORDERER_CA=${PWD}/organizations/ordererOrganizations/example.com/tlsca/tlsca.example.com-cert.pem
```

```
10 export
   ordererCa=$(pwd)/organizations/ordererOrganizations/example.com/orderers/orderer4.example.com/msp/tlscacerts/tlsca.example.com-cert.pem
11 export
   ORDERER_ADMIN_TLS_SIGN_CERT=${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer4.example.com/tls/server.crt
12 export
   ORDERER_ADMIN_TLS_PRIVATE_KEY=${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer4.example.com/tls/server.key
```

2.3 获取通道配置块文件

```
1 peer channel fetch config channel-artifacts/config_block.pb -o
   $ORDERER_CONTAINER -c $channel_name --tls --cafile $ordererCa
```

2.4 将块解码成可读取的JSON对象

```
1 cd /data/fabric/fabric-orderer4/test-network/channel-artifacts/
2 configtxlator proto_decode --input config_block.pb --type common.Block --
   output config_block.json | jq '.data.data[0].payload.data.config'
   config_block.json > config.json
3 cp -r config.json modified_config.json
```

2.5 修改配置

```
../organizations/ordererOrganizations/example.com/orderers/orderer4.example.com/tls/server.crt
```

```
1 [root@orderer4 channel-artifacts]# vim modified_config.json
2 ....
3     "values": {
4         "Endpoints": {
5             "mod_policy": "Admins",
6             "value": {
7                 "addresses": [
8                     "orderer.example.com:7050",
9                     "orderer2.example.com:7050",
10                    "orderer3.example.com:7050",
11                    "orderer4.example.com:7050"
12                ]
13            },
```

```

14         "version": "2"
15     },
16     ...
17     {
18         "client_tls_cert":
19         "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUNaekNDQWc2Z0F3SUJBZ0tLSQU9uVXlwdU5QQX
20         FyRC9zWEd0Qnc0UUVV3Q2dZSUtvWk16ajBFQXdJd2JERUwKTUFR0ExVUVCaE1DVlZNeEV6QVJCZ05WQ
21         kFnVENrTmhiR2xtYjNKdWFXRXhGakFVQmd0VkJBY1REVk5oYm1CRwpjbUZ1WTJse1kyOHhGREFTQmd0
22         VkJBb1RDMLY0WVcxd2JHVXVZMj10TVJvd0dBWURWUWFERXhGMGJITmpZUZVsCmVHRnRjR3hsTG10dmJ
23         UQWVGdzB5TkRBeE1EZ3dNa1U1TURCYUZ3MHP0REF4TURVd01qVTVNREJhTUZreEN6QUoKQmd0VkJBWV
24         RBbFZUTVJNd0VRWURWUWVFJRXdWRF1XeHBabTl5Ym1saE1SWXdGQVlEVlFRSEV3MVRZVzRnUm5KaApid
25         U5wYzJ0dk1SMHdHd11EVlFRREV4UnZjbVJsY21WeU5DNWx1R0Z0Y0d4bExtTnZiVEJaTUJNR0J5cUdT
26         TTQ5CkFnRUdDQ3FHU000OUF3RUhBME1BQk9BMEhlQVAvVXJtWGwzL2paeDIwTk1NdDk1OHhId0pBcXN
27         zS29VZjl1cFMKbk5hRmRHdWJLTEorbzM1L0FXZkg0R05DSWhRRjBxYkdBMmxwYXVvVThMcWpnYU13Z2
28         FBd0RnWURWUjBQVVFILWpCQVFEQWdXZ01CMEdBMVVKSlFRV01CUUdDQ3NHQVFRKJ3TUJJCZ2dyQmdFR
29         kJRY0RBakFNQmd0VkhSTUJBZjhFckFqQUFNQ3NHQTFVZEL3UWtNQ0tBSUdIc1Nj5bG9waHpDczZD
30         d1cwSFFCaFJwWnRxdEVUTlR2SXJBVVCWU4KTURRR0ExVWRFUWV0TUN1Q0ZH0XlaR1Z5WlhJMExtVjR
31         ZVzF3YkdVdVky0XRnZ2h2Y21SbGNtVn10SU1KYkc5agpZV3hvYjNOME1Bb0dDQ3FHU000OUJBTUNBMG
32         NBTUVRQ0lCTjBXYURmVG5UaHpqUE1Ndmk1SXpFbUcydHdxWnR5CnRER1hISlhqcEhxQ0FpQXdIbE80U
33         kNqbXRUX3pr0GNndUhsMXRwdTB6SU5nMUNFZVUyRitrQWRGZke9PQotLS0tLUVORCBDRVJUSUZJQ0FU
34         RS0tLS0tCg==",
35         "host": "orderer4.example.com",
36         "port": 7050,
37         "server_tls_cert": "同上"
38     }
39 }
40
41 "OrdererAddresses": {
42     "mod_policy": "/Channel/Orderer/Admins",
43     "value": {
44         "addresses": [
45             "orderer.example.com:7050",
46             "orderer2.example.com:7050",
47             "orderer3.example.com:7050",
48             "orderer4.example.com:7050"
49         ]
50     },
51     "version": "2"
52 }
53 ....

```

2.6 原始和修改后的信道配置都转换回protobuf格式

- 1 configtxlator proto_encode --input config.json --type common.Config --output config.pb
- 2 configtxlator proto_encode --input modified_config.json --type common.Config --output modified_config.pb

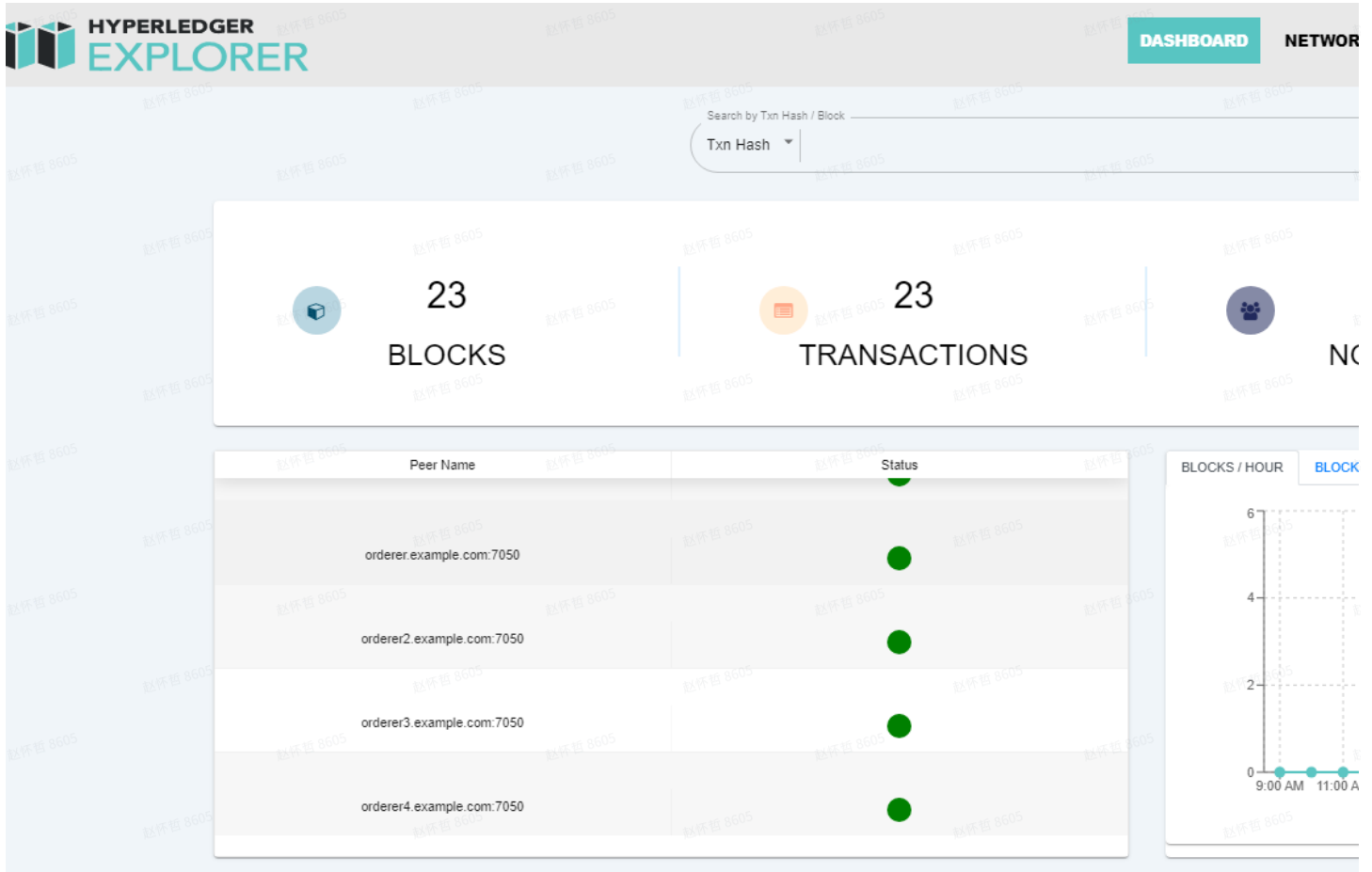
```
3 configtxlator compute_update --channel_id $channel_name --original config.pb --
  updated modified_config.pb --output config_update.pb
```

2.7 配置更新封装在事务信封中以创建通道配置更新事务

```
1 configtxlator proto_decode --input config_update.pb --type common.ConfigUpdate
  --output config_update.json
2 echo '{"payload":{"header":{"channel_header":{"channel_id":"'${channel_name}',
  "type":2}},"data":{"config_update":"'$(cat config_update.json)'"}}}' | jq . >
  config_update_in_envelope.json
3 configtxlator proto_encode --input config_update_in_envelope.json --type
  common.Envelope --output config_update_in_envelope.pb
4 cd ..
```

2.8 更新通道

```
1 [root@orderer4 test-network]# peer channel update -f channel-
  artifacts/config_update_in_envelope.pb -c $channel_name -o $ORDERER_CONTAINER -
  -tls --cafile $ordererCa
2
3 2024-03-06 09:46:03.661 CST 0001 INFO [channelCmd] InitCmdFactory -> Endorser
  and orderer connections initialized
4 2024-03-06 09:46:03.686 CST 0002 INFO [channelCmd] update -> Successfully
  submitted channel update
```



del

3. orderer4

3.1 获取通道配置块文件

```
1 rm -rf channel-artifacts/*.pb channel-artifacts/*.json
2 peer channel fetch config channel-artifacts/config_block.pb -o
  $ORDERER_CONTAINER -c $channel_name --tls --cafile $ordererCa
```

3.2 将块解码成可读取的JSON对象

```
1 cd /data/fabric/fabric-orderer4/test-network/channel-artifacts/
2 configtxlator proto_decode --input config_block.pb --type common.Block --
  output config_block.json | jq '.data.data[0].payload.data.config'
  config_block.json > config.json
3 cp -r config.json modified_config.json
```

3.3 修改配置

- 1 [root@orderer4 channel-artifacts]# vim modified_config.json
- 2 删除关于orderer4.example.com的所有内容

3.4 原始和修改后的信道配置都转换回protobuf格式

- 1 configtxlator proto_encode --input config.json --type common.Config --output config.pb
- 2 configtxlator proto_encode --input modified_config.json --type common.Config --output modified_config.pb
- 3 configtxlator compute_update --channel_id \$channel_name --original config.pb --updated modified_config.pb --output config_update.pb

3.5 配置更新封装在事务信封中以创建通道配置更新事务

- 1 configtxlator proto_decode --input config_update.pb --type common.ConfigUpdate --output config_update.json
- 2 echo '{"payload":{"header":{"channel_header":{"channel_id":"'\${channel_name}'","type":2}},"data":{"config_update":"'\$(cat config_update.json)'"}}}' | jq . > config_update_in_envelope.json
- 3 configtxlator proto_encode --input config_update_in_envelope.json --type common.Envelope --output config_update_in_envelope.pb
- 4 cd ..

3.6 更新通道

- 1 [root@orderer4 test-network]# peer channel update -f channel-artifacts/config_update_in_envelope.pb -c \$channel_name -o \$ORDERER_CONTAINER -tls --cafile \$ordererCa
- 2
- 3 2024-03-06 11:13:32.739 CST 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
- 4 2024-03-06 11:13:32.765 CST 0002 INFO [channelCmd] update -> Successfully submitted channel update

3.7 explorer

- 1 cd /data/fabric/fabric-orderer1/test-network/explorer

```
2 [root@orderer1 explorer]# docker-compose down -v && docker-compose up -d
```

DGER
ORDERER

DASHBOARD

NETWORK

BLOCKS

TRA



24
BLOCKS



24
TRANSACTIONS



3
NODES

Peer Name	Status
peer0.org1.example.com:7051	
orderer.example.com:7050	
orderer2.example.com:7050	
orderer3.example.com:7050	

BLOCKS / HOUR

BLOCKS / MIN

TX / HOUR

