

# 动态添加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 -ca -verbose
3 ./network.sh 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/u
   sers/Admin\@example.com/msp/
8 export
   CORE_PEER_TLS_ROOTCERT_FILE=$(pwd)/organizations/ordererOrganizations/example.c
   om/orderers/orderer4.example.com/tls/ca.crt
9 export
   ORDERER_CA=${PWD}/organizations/ordererOrganizations/example.com/tlsca/tlsca.ex
   ample.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 (JSON转Base64)

```
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         "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUN6ekNDQW5hZ0F3SUJBZ0lVRW9KeIhIWlNBQ3
20         pIRVFLZ3B2T0V2WlJLY3ZVd0NnWUllb1pJemowRUF3SXcKWwPFTe1Ba0dBmVVFQmhnNQ1ZWTXhFVEFQQ
21         md0VkJBZ1RDRTVsZlhlCWmIzSnJNUkV3RhdZRFZRUUhFd2hPWlhjZwpXVzlsYXpFVU1CSUdBmVVFQ2hN
22         TFpYaGhiWEJzWlM1amIyMHhGekFWQmd0VkJBTvREBU5oTG1WNFLXMXdiR1V1ClkyOXRNQjRyRFRJME1
23         ETXh0REF6TVRZd01Gb1hEVEkxTURNee5EQXpNakV3TUZvd1lERUxNQWtHQTFVRUJoTUMKVLZNeEZ6QV
24         ZCZ05WQkFnVERrNXZjb1JvSUV0aGNTOXNhVzVoTVJRd0VnWURWUWFLRXd0SWVYQmxjbXhsWkdkbApja
25         kVRTUE0R0EXVUVDeE1IYjNka1pYSmxjakVRTUE0R0EXVUVBeE1IYjNka1pYSmxjakJaTUJNR0J5cUdT
26         TTQ5CkFnRUdDQ3FHU000OUF3RUhBMElBQlBaTVV6SGhTWdJicU1zeXpNQXpSV004Nk82V3l3S3RIK1l
27         1enZ0cHk2NCsKUme1c0l5M0hpSzBvblVvOVltSEEydkgvUmVTRjNUM3hiWjLEZDdyZC83K2pnZ0VLTU
28         lJQkQjQU9CZ05WSFE4QgpBZjhFQkFNQ0E2Z3dIUUVLEVlIwbEJCWXdGQVlJS3dZQkJRvUhd0VHQ0NzR
29         0FRVUZCd01DTUF3R0EXVWRfd0VCCi93UUNNQUF3SFFZRFZSME9CQllFRkU1czNnZzgvOTg3bVR2UmxE
30         ZHFYwLJ5UXEzaE1COEdBMVVKsXdrWU1CYUEKRkFLSWpBNjdhsVp3V0Y1LzA3aXhKT3F5YlltaE1Db0d
31         BMVVKrVFRak1DR0NGRzls5WkdWeVpYSTBmbVY0WVcxdwpiR1V1WTI5dGdnbHNiMk5oYkdodmMzUXdXd1
32         lJS2dNRUJRwUhdQUVFVDNzaVlYUjBjbk1pT25zaWFHWXVRV1ptCmFXeHBZWfJwYjI0aU9pSWlMQ0pvW
33         mk1RmJuSnZiR3h0Wlc1MfNVUwLPaUp2Y21SbGNTVn1JaXdpYUdZdVZlbnBhcnVlbnNkltOXlaR1Z5WlhJ
34         aWZYMhDZl1lJS29aSXpqMEVBd0lEundBd1JBswdFazd6QitPTHhPOVhnaEJJNkhWawo0NnFUMlFaUVZ
35         vL1FnT2tPUVRlWk42SUNJQ1RobDQ2SW10djNhMElZWwY1TFJoSFNFsfJBVUF6UTFjZ09mNmRHCmV2ak
36         IKLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=",
37         "host": "orderer4.example.com",
38         "port": 7050,
39         "server_tls_cert": "同上"
40     }
41 }
42 "OrdererAddresses": {
43     "mod_policy": "/Channel/Orderer/Admins",
44     "value": {
45         "addresses": [
46             "orderer.example.com:7050",
47             "orderer2.example.com:7050",
48             "orderer3.example.com:7050",
49             "orderer4.example.com:7050"
50         ]
51     },
52     "version": "2"
53 }
54 }
55 ....

```

## 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

```

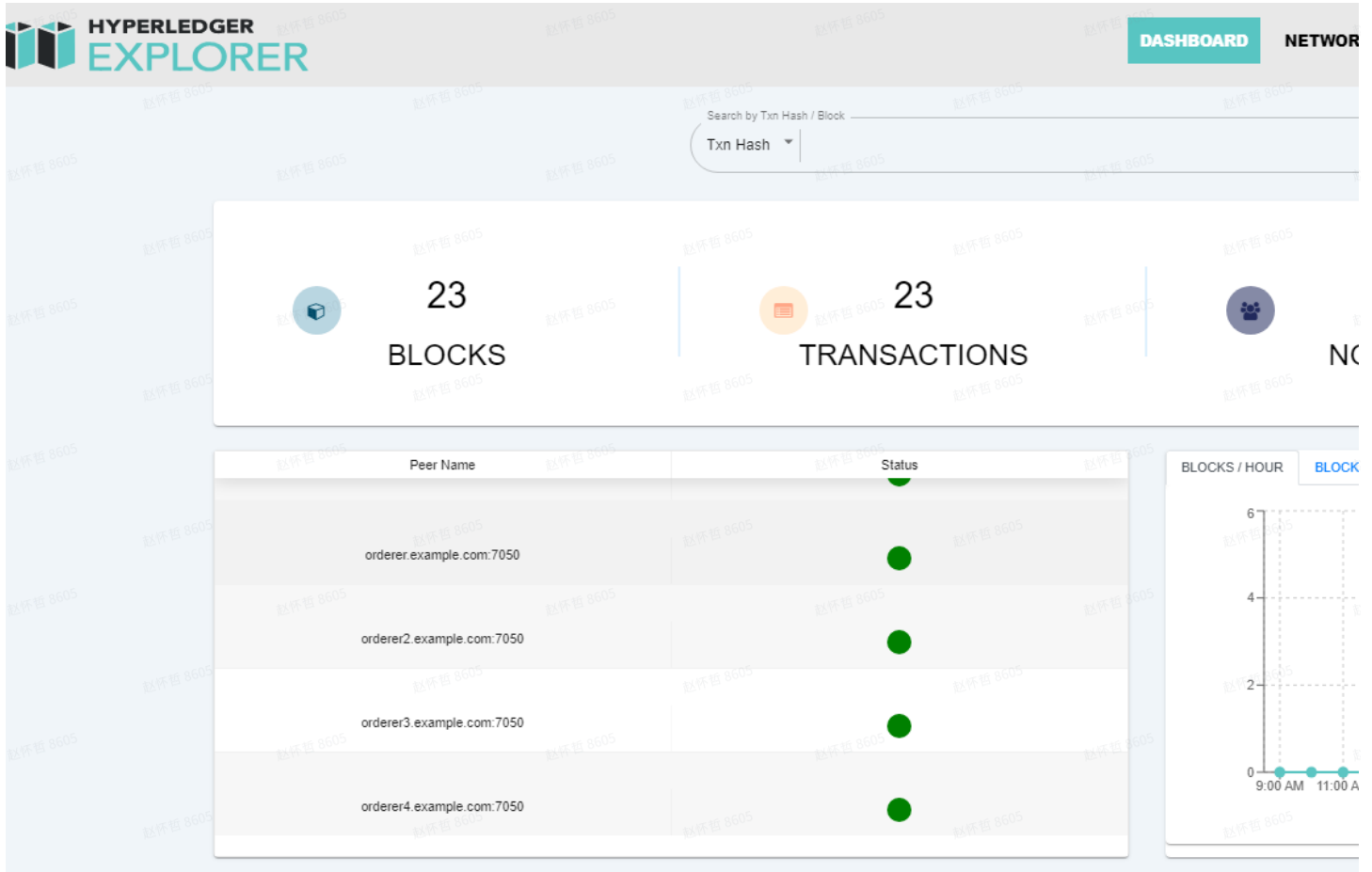
```
2 configtxlator proto_encode --input modified_config.json --type common.Config --
  output modified_config.pb
3 configtxlator compute_update --channel_id $channel_name --original config.pb --
  updated modified_config.pb --output config_update.pb
```

## 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 CST 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections
  initialized
4 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
3 jq '.data.data[0].payload.data.config' config_block.json > config.json
4 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 CST 0001 INFO [channelCmd] InitCmdFactory -> Endorser and orderer connections initialized
- 4 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



24  
BLOCKS



24  
TRANSACTIONS



3  
NODES

Peer Name	Status
peer0.org1.example.com:7051	<div></div>
orderer.example.com:7050	<div></div>
orderer2.example.com:7050	<div></div>
orderer3.example.com:7050	<div></div>

