腾讯云

第5章 FISCO BCOS区块链运维（下）

目录

CONTENTS

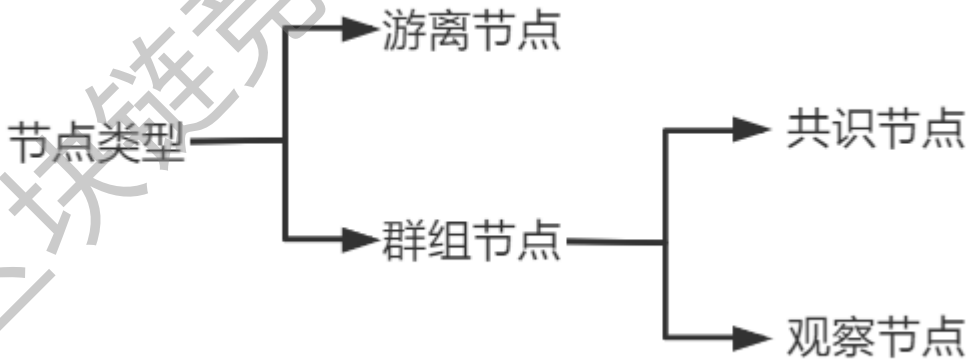
CONTENTS

章节

第一章 节点管理

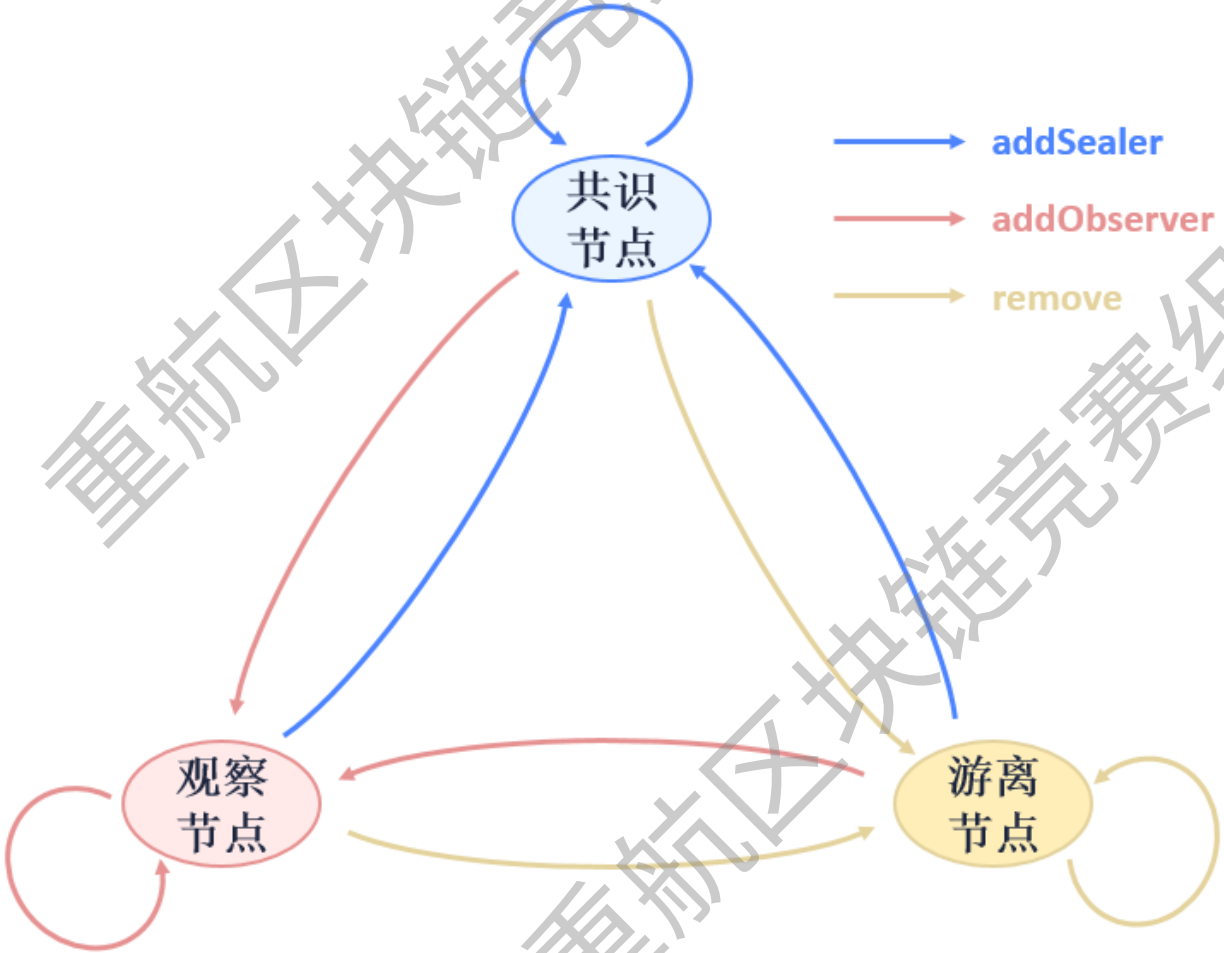
节点类型

- FISCO BCOS区块链系统中有三种不同的节点类型
  - 共识节点：完成网络准入并加入群组的节点，参与共识的节点，拥有群组的所有数据（搭链时默认都生成共识节点）。
  - 观察节点：完成网络准入并加入群组的节点，不参与共识，但能实时同步链上数据的节点。
  - 游离节点：完成网络准入但没有加入群组的节点。游离节点尚未通过群组准入，不参与共识和同步。游离节点无法获取链上数据，也无法连接到控制台。



节点类型

- 三种不同的节点类型可相互转换



节点操作命令

- 控制台提供了一系列命令以支持节点间的相互转换
  - getSealerList：查看群组中共识节点列表；
  - addSealer：根据节点NodeID设置对应节点为共识节点；
  - getObserverList：查看群组中观察节点列表；
  - addObserver：根据节点NodeID设置对应节点为观察节点；
  - getNodeIDList：查看连接p2p节点的nodeId列表；
  - removeNode：根据节点NodeID设置对应节点为游离节点。

基于控制台管理节点

- 准备工作
  - 使用开发部署工具搭建一个单群组4节点的区块链网络，启动所有节点，并保证区块链系统的正常运行（节点进程正常、共识正常等）。

```
$ bash nodes/127.0.0.1/start_all.sh
try to start node0
try to start node1
try to start node2
try to start node3
node0 start successfully
node1 start successfully
node2 start successfully
node3 start successfully
```

四个节点均启动成功

基于控制台管理节点

- 准备工作
  - 拷贝sdk证书到控制台，并启动控制台，使控制台成功连接到该区块链节点。

```
$ cp nodes/127.0.0.1/sdk/* console/conf          拷贝sdk证书文件
$ cp console/conf/config-example.toml console/conf/config.toml     重命名控制台配置文件
$ bash console/start.sh          启动控制台
=====================================================================================
Welcome to FISCO BCOS console(2.8.0)!
Type 'help' or 'h' for help. Type 'quit' or 'q' to quit console.
 ________  ______   ______   ______       _______   ______   ______   ______
|        \|      \ /      \ /      \     |       \ /      \ /      \ /      \
| $$$$$$$$\$$$$$$| $$$$$$| $$$$$$| $$$$$$\    | $$$$$$$| $$$$$$| $$$$$$| $$$$$$\
| $$__     | $$ | $$___\$| $$   \$| $$  | $$    | $$__/ $| $$  \$| $$  | $| $$___\$$
| $$  \    | $$  \$$    \| $$     | $$  | $$    | $$    $| $$     | $$  | $$\$$    \
| $$$$$    | $$  _\$$$$$$| $$   __| $$  | $$    | $$$$$$$| $$   __| $$  | $$_\$$$$$$\
| $$      _| $$_|  \__| $| $$__/  | $$__/ $$    | $$__/ $| $$__/  | $$__/ $|  \__| $$
| $$     |   $$ \\$$    $$\$$    $$\$$    $$    | $$    $$\$$    $$\$$    $$\$$    $$
 \$$      \$$$$$$ \$$$$$$  \$$$$$$  \$$$$$$      \$$$$$$$  \$$$$$$  \$$$$$$  \$$$$$$
                                  成功进入控制台
=====================================================================================
[group:1]>
```

基于控制台管理节点

- 将共识节点转为观察者节点
  - 搭链时，所有节点都默认生成共识节点，输入如下命令，可查看区块链共识节点列表：getSealerList

```
[group:1]> getSealerList
[
    1479a33468e9fe7bbf61bd1a819426a0ccae8546a8f1f54ba5046caffb5bb6ee36c6264454aab63793e7ac3524c7fd1565d15e641234fe91105f843636731d62,
    6658920113d64a086fb4b191c58cc68e5a496525e765e2772929cb8f8d12d61f66f9effe1df058f85b4be4c8bc892c513e0fc8e486af240256fdd1ee16e68c67,
    8c9c5933e37016a07803402dac659408bab2ff53658ead77e96107fdf1ad4525a26f36995f9458ff5232a175ffcfa7245f70b2cc1619f978de31911340aa7612,
    f42e15c955894acca6e92ee1f97833fc44167756f7783303835b9a924609677663a744379573031822d7f9d3c568a3928c22fd6235ad16ebcf1691051676c79b
]
```

此时区块链系统中有4个共识节点

  - 查看node0的nodeid：cat nodes/127.0.0.1/node0/conf/node.nodeid

```
$ cat test_nodes/127.0.0.1/node0/conf/node.nodeid
f42e15c955894acca6e92ee1f97833fc44167756f7783303835b9a924609677663a744379573031822d7f9d3c568a3928c22fd6235ad16ebcf1691051676c79b
```

node0的nodeid

基于控制台管理节点

- 在控制台中输入如下命令，将node0转为观察者节点：addObserver f42e15c955894acca6e92ee1f97833fc44167756f7783303835b9a924609677663a744379573031822d7f9d3c568a3928c22fd6235ad16ebcf1691051676c79b
- 第二个参数为node0的nodeid，根据实际情况填写参数。

```
[group:1]> addObserver f42e15c955894acca6e92ee1f97833fc44167756f7783303835b9a924609677663a744379573031822d7f9d3c56
{
    "code":1,
    "msg":"Success"
}
```

转换节点成功

- 查看观察者节点列表：getObserverList

```
[group:1]> getObserverList
[
    f42e15c955894acca6e92ee1f97833fc44167756f7783303835b9a924609677663a744379573031822d7f9d3c568a3928c22fd6235ad16
]
```

node0的nodeid

基于控制台管理节点

- 将观察者节点转为共识节点
  - 在控制台中输入如下命令，将node0转为共识节点：addSealer f42e15c955894acca6e92ee1f97833fc44167756f7783303835b9a924609677663a744379573031822d7f9d3c568a3928c22fd6235ad16ebcf1691051676c79b
  - 第二个参数为node0的nodeid，根据实际情况填写参数。

```
[group:1]> addSealer f42e15c955894acca6e92ee1f97833fc44167756f7783303835b9a924609677663a744379573031822d7f9d3c568a3928c22fd6235ad16ebcf1691051676c7
{
    "code":1,
    "msg":"Success"
}
```

节点转换成功

  - 查看共识节点列表： getSealerList

```
[group:1]> getSealerList
[
    1479a33468e9fe7bbf61bd1a819426a0ccae8546a8f1f54ba5046caffb5bb6ee36c6264454aab63793e7ac3524c7fd1565d15e641234fe91105f843636731d62,
    6658920113d64a086fb4b191c58cc68e5a496525e765e2772929cb8f8d12d61f66f9effe1df058f85b4be4c8bc892c515e0fc8e486af240256fdd1ee16e68c67,
    8c9c5933e37016a07803402dac659408bab2ff53658ead77e96107fdf1ad4525a26f36995f9458ff5232a175ffcf7245f70b2cc1619f978de31911340aa7612,
    f42e15c955894acca6e92ee1f97833fc44167756f7783303835b9a924609677663a744379573031822d7f9d3c568a3928c22fd6235ad16ebcf1691051676c79b
]
```

node0的nodeid

基于控制台管理节点

- 将共识节点转为游离节点
  - 在控制台中输入如下命令，将node0转为游离节点： removeNode f42e15c955894acca6e92ee1f97833fc44167756f7783303835b9a924609677663a744379573031822d7f9d3c568a3928c22fd6235ad16ebcf1691051676c79b
  - 第二个参数为node0的nodeid，根据实际情况填写参数。

```
[group:1]> removeNode f42e15c955894acca6e92ee1f97833fc44167756f7783303835b9a924609677663a744379573031822d7f9d3c568
{
    "code":1,
    "msg":"Success"
}
```

节点转换成功

  - 查看共识节点列表：getSealerList

```
[group:1]> getSealerList
[
    1479a33468e9fe7bbf61bd1a819426a0ccae8546a8f1f54ba5046caffb5bb6ee36c6264454aab63793e7ac3524c7fd1565d15e641234fe91105f843636731d62,
    6658920113d64a086fb4b191c58cc68e5a496525e765e2772929cb8f8d12d61f66f9effe1df058f85b4be4c8bc892c513e0fc8e486af240256fdd1ee16e68c67,
    8c9c5933e37016a07803402dac659408bab2ff53658ead77e96107fdf1ad4525a26f36995f9458ff5232a175ffcfa7245f70b2cc1619f978de31911340aa7612,
]
```

此时node0不在共识列表中

基于控制台管理节点

- 查看观察者节点列表： getObserverList

```
[group:1]> getObserverList
[]
```

此时观察者列表应空

- 查看所有的p2p节点列表：getNodeIDList

```
[group:1]> getNodeIDList
[
    b900ad6f2c164adf8ef3e545a36358801eeb7c9b0976646a15f11c838b5150e66fbc06d5c7a170f3cc4c07e1488a2856965b406e3f971ab987a318c2b7bb0f5b,
    bdff4c27674a75a5f78743bdd45c0d59eda1fd8d67c393370d7f91c310589290a81426568a8891a351f92928b3822c1e8ed5f46d0e84ae8ebc7d2b42000b7801,
    65caf3cf7606e83d557b8117a20bfd00c1be430ea299c806e20425c9646f0f46c2ad923157b0e8f54e39a34d929d9cef2f0ca3f6d706f5f9847663afa88266fa,
    04f648f41d067eeeb17bf797c5b867ea2f743b18b258a531caa018a6c68e0d5ca55476f8dd914d4431659a40723a363f32de030af95ce92769914006cd4f80cc
]
```

p2p节点列表包含了所有类型的节点

CONTENTS

章节

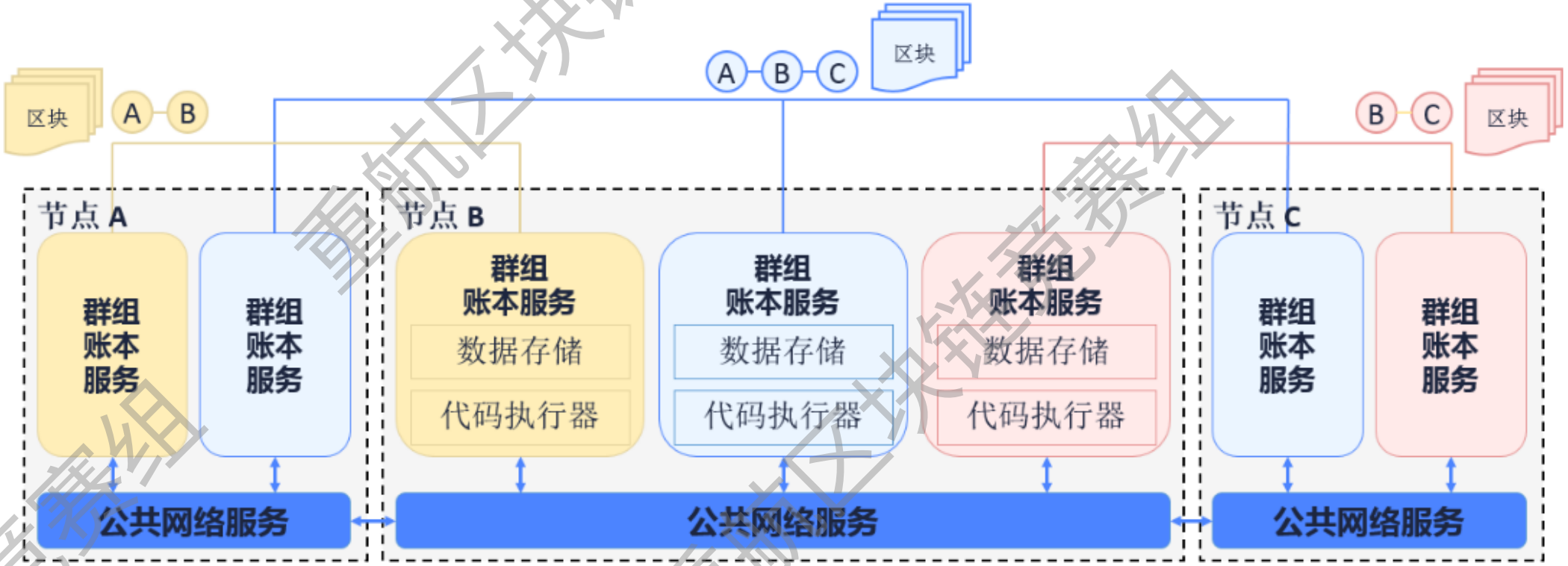
第二章 扩容和退出区块链

准入原则

- 区块链技术是一种去中心化、公开透明的分布式数据存储技术，能够降低信任成本，实现安全可靠的数据交互。然而区块链的交易数据面临着隐私泄露威胁。
  - 对于公有链，一节点可任意加入网络，从全局账本中获得所有数据；
  - 对于联盟链，虽有网络准入机制，但节点加入区块链后即可获取全局账本的数据。
- FISCO BCOS的多群组设计，使联盟链从原有一链一账本的存储/执行机制扩展为一链多账本的存储/执行机制，对链上隐私这一问题，提出了单链多账本的解决方案。

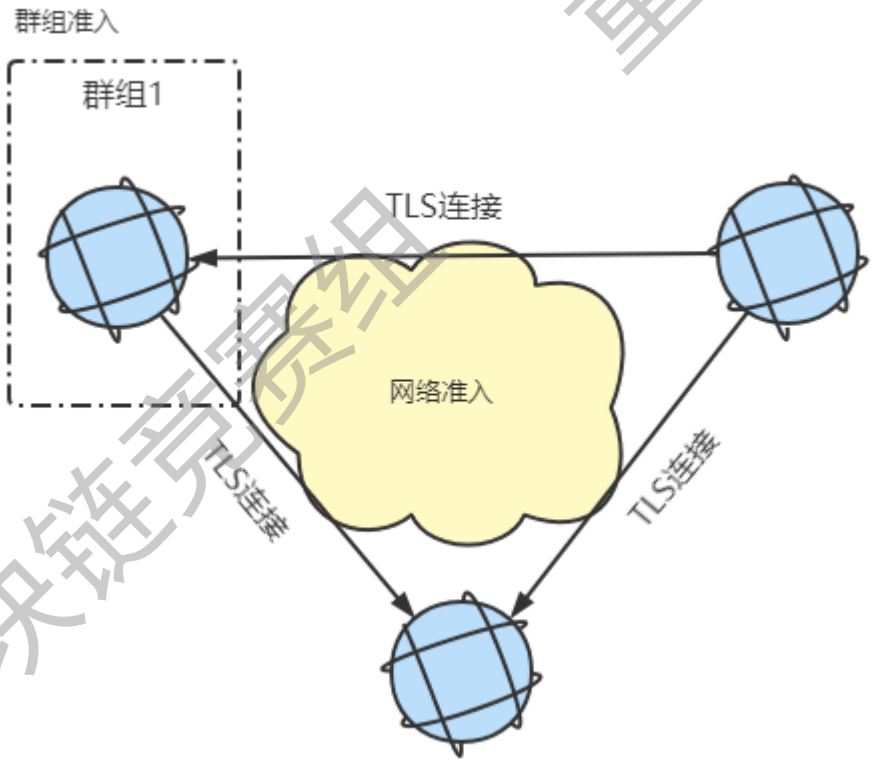
准入原则

- 节点ABC加入蓝色群组，并共同维护蓝色账本; 节点B和C加入粉色群组并维护粉红色账本; 节点A和B加入黄色群组并维护黄色账本。



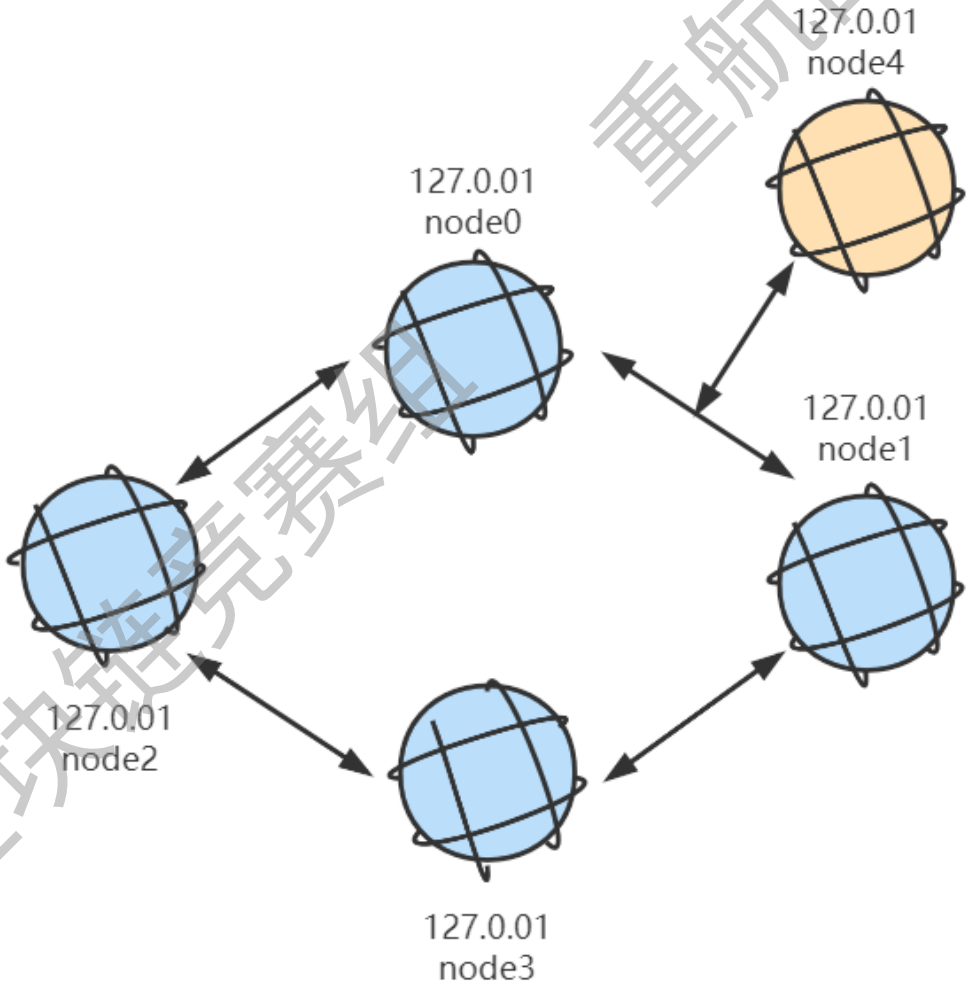
准入原则

- 基于群组概念的引入，FISCO BCOS节点准入管理可分为：
  - 网络准入机制：控制节点入网/退网；
  - 群组准入机制：控制节点加入/退出群组。



节点扩容

- 场景描述
  - 在一个单群组4节点的区块链网络中，添加一个新节点node4，组网结构如图。
- 扩容流程
  - 为节点生成证书（用于建立连接）
  - 准备节点配置文件
  - 加入网络
  - 加入群组



节点扩容

- 搭建单群组区块链网络
  - 使用开发部署工具搭建一个单群组4节点的区块链网络，启动所有节点，并保证区块链系统的正常运行（节点进程正常、共识正常等）。

```
$ bash nodes/127.0.0.1/start_all.sh
try to start node0
try to start node1
try to start node2
try to start node3
node0 start successfully
node1 start successfully
node2 start successfully
node3 start successfully
```

四个节点均启动成功

节点扩容

- 为新节点生成证书
  - 获取生成节点证书的脚本：wget http://res.zhonghui.vip/blockchain/fisco-bcos/01/resource/gen_node_cert.sh

```
$ wget http://res.zhonghui.vip/blockchain/fisco-bcos/01/resource/gen_node_cert.sh
--2022-04-23 14:59:20--  http://res.zhonghui.vip/blockchain/fisco-bcos/01/resource/gen_node_cert.sh
正在解析主机 res.zhonghui.vip (res.zhonghui.vip)... 121.89.193.240
正在连接 res.zhonghui.vip (res.zhonghui.vip)|121.89.193.240|:80... 已连接。
已发出 HTTP 请求，正在等待回应... 301 Moved Permanently
位置: http://res.handge.cn:8999/blockchain/fisco-bcos/01/resource/gen_node_cert.sh [跟随至新的 URL]
--2022-04-23 14:59:21--  http://res.handge.cn:8999/blockchain/fisco-bcos/01/resource/gen_node_cert.sh
正在解析主机 res.handge.cn (res.handge.cn)... 118.122.97.204
正在连接 res.handge.cn (res.handge.cn)|118.122.97.204|:8999... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度:  11377 (11K) [application/octet-stream]
正在保存至:  “gen_node_cert.sh”

gen_node_cert.sh            100%[=========================================================>]  11.11K

2022-04-23 14:59:21 (712 KB/s) - 已保存  “gen_node_cert.sh”  [11377/11377])
```

下载成功

节点扩容

- 修改脚本权限：chmod u+x gen_node_cert.sh
- 执行脚本，为新节点生成证书：bash gen_node_cert.sh -c nodes/cert/agency -o nodes/127.0.0.1/node4
  - ✓ -c：指定机构证书及私钥所在路径
  - ✓ -o：输出到指定文件夹，其中node4/conf中会保存机构agency新签发的证书和私钥

```
$ chmod u+x gen_node_cert.sh
$ bash gen_node_cert.sh -c ../cert/agency -o node4
============================================================
[INFO] Cert Path    : ../cert/agency
[INFO] Output Dir   : node4
============================================================
[INFO] All completed. Files in node4
```

修改脚本权限

为新节点生成证书

生成证书成功

节点扩容

- 准备新节点配置文件
  - 拷贝node0目录下的config.ini、start.sh和 stop.sh到node4目录：cp node0/config.ini node0/start.sh node0/stop.sh node4/
  - 查看文件，验证配置文件是否拷贝成功：ls node4

```
$ cp node0/config.ini node0/start.sh node0/stop.sh node4/
$ ls node4
conf  config.ini  start.sh  stop.sh
```

- 使用vim命令修改config.ini配置文件：vim node4/config.ini
  - 对于[rpc]模块，修改channel_listen_port=20204和jsonrpc_listen_port=8549；
  - 对于[p2p]模块，修改listen_port=30304并在node.中增加自身节点信息。

节点扩容

- node4/config.ini配置文件修改如下

```
$ vim node4/config.ini
[rpc]
    channel_listen_ip=0.0.0.0
    channel_listen_port=20204    channel监听端口
    jsonrpc_listen_ip=127.0.0.1
    jsonrpc_listen_port=8549    rpc监听端口
[p2p]
    listen_ip=0.0.0.0
    listen_port=30304    p2p监听端口
    ; nodes to connect
    node.0=127.0.0.1:30300
    node.1=127.0.0.1:30301
    node.2=127.0.0.1:30302
    node.3=127.0.0.1:30303
    node.4=127.0.0.1:30304    节点连接信息
```

节点扩容

- 拷贝node0/conf目录下的group.1.genesis和group.1.ini配置文件到node4/conf目录：cp node0/conf/group.1.genesis node0/conf/group.1.ini node4/conf/
- 查看文件，验证配置文件是否拷贝成功：ls node4/conf

```
$ cp node0/conf/group.1.genesis node0/conf/group.1.ini node4/conf/
$ ls node4/conf
agency.crt  ca.crt  group.1.genesis  group.1.ini  node.crt
node.key  node.nodeid  node.param  node.private  node.pubkey
```

拷贝的文件，无需改动

腾讯云

节点扩容

- 新节点加入网络
  - 启动新节点node4：bash node4/start.sh

```
$ bash node4/start.sh
node4 start successfully
```

node4启动成功

  - 查看node4的连接信息：tail -f node4/log/log* |grep “connected count “

```
$ tail -f node4/log/log* |grep "connected count"
info|2022-03-10 16:36:01.850422|[P2P][Service] heartBeat,connected count=4
info|2022-03-10 16:36:11.851120|[P2P][Service] heartBeat,connected count=4
info|2022-03-10 16:36:21.851336|[P2P][Service] heartBeat,connected count=4
```

node4已成功接入网络，并与另外4个节点建立了连接

想一想此时的node4属于什么类型的节点?

  - 查看node4的nodeid：cat node4/conf/node.nodeid

```
$ cat node4/conf/node.nodeid
7aa1919b3e7b75e55eca84dcfb0448fb16c1aa21d2e8ac193235947bc50f8b96321318d424f156b3b87b06
```

节点扩容

- 启动控制台
  - 拷贝sdk证书到控制台，并启动控制台，使控制台成功连接到该区块链节点。

```
$ cp nodes/127.0.0.1/sdk/* console/conf          拷贝sdk证书文件
$ cp console/conf/config-example.toml console/conf/config.toml     重命名控制台配置文件
$ bash console/start.sh          启动控制台
=====================================================================================================
Welcome to FISCO BCOS console(2.8.0)!
Type 'help' or 'h' for help. Type 'quit' or 'q' to quit console.
 ________ ______  ______   ______   ______       _______   ______   ______
|        |      \|      \ /      \ /      \     |       \ /      \ /      \|      \
| $$$$$$$$\$$$$$$| $$$$$$|  $$$$$$|  $$$$$$\    | $$$$$$$| $$$$$$|  $$$$$$|  $$$$$$\
| $$__     | $$  | $$___\$| $$   \$| $$  | $$    | $$__/ $| $$   \$| $$  | $| $$___\$$
| $$  \    | $$   \$$    \| $$     | $$  | $$    | $$    $| $$     | $$  | $$\$$    \
| $$$$$    | $$   _\$$$$$$| $$   __| $$  | $$    | $$$$$$$| $$   __| $$  | $$_\$$$$$$\
| $$      _| $$_ |  \__| $| $$__/  | $$__/ $$    | $$__/ $| $$__/  | $$__/ $|  \__| $$
| $$     |   $$ \\$$    $$\$$    $$\$$    $$    | $$    $$\$$    $$\$$    $$\$$    $$
 \$$      \$$$$$$ \$$$$$$  \$$$$$$  \$$$$$$      \$$$$$$$  \$$$$$$  \$$$$$$  \$$$$$$
                                  成功进入控制台
=====================================================================================================
[group:1]>
```

节点扩容

- 新节点加入群组
  - 查看群组1中的共识节点：getSealerList

```
[group:1]> getSealerList
[
    6cb4e3234bc75ea48a432dc517c2f912a4c8aa4c2593bd33827ee0d0a5908f200abb56e8e379ea957b24307b68d007f92144a5fa896acfb5cddc40f9ff2a2725,
    6d03f28808657321b4c8fda53288d63d6a114dc446fa9c4f8e0859bf68393bc0e3d29d0b72d885e96d9b27b4c445cf9cf11fc1084639bc8a516ebf53174b6a3e,
    9e481f6ab50668cff890aae7c1b2299e515e1aec22b2c25156b4bc5e385b3bd62d1b64f8b440c390dee4bdb849f900ae2b08012811144419db831e21d53723cda,
    ecf150911a70a2787700be4a256d0ad4607af0a4fb5da70a28514384b371f2d84b902dec166d19488fea1038c8d330295850c1d8c37b619d06eb4cde67341604
]
```

此时有4个共识节点

  - 将新节点 node4 作为共识节点加入群组1：addSealer

7aa1919b3e7b75e55eca84dcfb0448fb16c1aa21d2e8ac193235947bc50f8b96321318d424f156b3b87b0660c7eb3561854c33b623211aa54d6c0822e6b882b2

```
[group:1]> addSealer 7aa1919b3e7b75e55eca84dcfb0448fb16c1aa21d2e8ac193235947bc50f8b96321318d424f156b3b87b0660c7eb3561854c33b623211aa54d6c0822e6b882b2
{
    "code":1,
    "msg":"Success"
}
```

node4的nodeid

节点扩容

- 再次查看群组1中的共识节点：getSealerList

```
[group:1]> getSealerList
[
    6cb4e3234bc75ea48a432dc517c2f912a4c8aa4c2503bd33827ee0d0a5908f200abb56e8e379ea957b24307b68d007f92144a5fa896acfb5cddc40f9ff2a2725,
    6d03f28808657321b4c8fda53288d63d6a114dc446fa9c4f8e0859bf68393bc0e3d29d0b72d885e96d9b27b4c445cf9cf11fc1084639bc8a516ebf53174b6a3e,
    9e481f6ab50668cff890aae7c1b2299e515e1aec22b2c25156b4bc5e385b3bd62d1b64f8b440c390dee4bdb849f900ae2b08012811144l9db831e21d53723cda,
    ecf150911a70a2787700be4a256d0ad4607af0a4fb5da70a28514384b371f2d84b902dec166d19488fea1038c8d330295850c1d8c37b619d06eb4cde67341604,
    7aa1919b3e7b75e55eca84dcfb0448fb16c1aa21d2e8ac193235947bc50f8b96321318d424f156b3b87b0660c7eb3561854c33b623211aa54d6c0822e6b882b2
]
```

此时node4已作为共识节点成功加入群组1

node4的nodeid

节点退网

- 退出群组
  - 节点退出网络的顺序应与加入时相反，即先退出群组，再退出网络，否则将触发节点异常。退出的顺序由用户保证，系统不再作校验。
  - 将node4设置为游离节点，即退出群组：removeNode 7aa1919b3e7b75e55eca84dcfb0448fb16c1aa21d2e8ac193235947bc50f8b96321318d424f156b3b87b0660c7eb3561854c33b623211aa54d6c0822e6b882b2

```
[group:1]> removeNode 7aa1919b3e7b75e55eca84dcfb0448fb16c1aa21d2e8ac193235947bc50f8b96321318d424f156b3b87b0660c7eb3561854c33b623211aa54d6c0822e6b882b2
{
    "code":1,
    "msg":"Success"
}
```

node4的nodeid

节点退网

- node4退出群组后，查询group1的共识节点是否包含node4的nodeid，如已消失，退出群组操作完成：getSealerList

```
[group:1]> getSealerList
[
    6cb4e3234bc75ea48a432dc517c2f912a4c8aa4c2593bd33827ee0d0a5908f200abb56e8e379ea957b24307b68d007f92144a5fa896acf
    6d03f28808657321b4c8fda53288d63d6a114dc446fa9c4f8e0859bf68393bc0e3d29d0b72d885e96d9b27b4c445cf9cf11fc1084639bc
    9e481f6ab50668cff890aae7c1b2299e515e1aec22b2c25156b4bc5e385b3bd62d1b64f8b440c390dee4bdb849f900ae2b080128111441
    ecf150911a70a2787700be4a256d0ad4607af0a4fb5da70a28514384b371f2d84b902dec166d19488fea1038c8d330295850c1d8c37b61
]
```

node4退出群组后，群组1中只有原先的4个共识节点

节点退网

- 退出网络
  - 编辑node4的config.ini配置文件，清除节点连接列表
  - 若其他节点的连接列表中有node4，需要同时将node4从其他节点的连接列表中清除。

```
[p2p]
    listen_ip=0.0.0.0
    listen_port=30304
    ; nodes to connect
    ; node.0=127.0.0.1:30300
    ; node.1=127.0.0.1:30301
    ; node.2=127.0.0.1:30302
    ; node.3=127.0.0.1:30303
    ; node.4=127.0.0.1:30304
```

在每一行前加分号“;”注释即可

节点退网

- 重启node4：bash node4/stop.sh && bash node4/start.sh

```
bash node4/stop.sh && bash node4/start.sh
stop node4 success.
node4 start successfully
```

节点重启成功

- 输入如下命令，查看连接数，若node4连接数为0，则说明退网成功：tail -f node4/log/log* |grep "connected count"

```
tail -f node4/log/log* |grep "connected count"
info|2022-04-22 15:59:59.073209|[P2P][Service] heartBeat,connected count=0
info|2022-04-22 16:59:59.357349|[P2P][Service] heartBeat,connected count=0
info|2022-04-22 17:59:59.673067|[P2P][Service] heartBeat,connected count=0
```

node4连接数为0

作业

1. FISCO BCOS RPC采用的是哪种协议？
2. RPC 请求包括哪些内容？
3. 调用成功的 RPC 响应包括哪些内容？
4. 控制台有哪些功能？
5. 写三个常用的控制台命令并写出其含义？
6. FISCO BCOS有哪些节点类型？分别有什么特征？
7. 将节点加入共识列表是什么命令？
8. 节点退出区块链网络的流程是什么？
9. 扩容一个节点的流程是什么？
10. 将节点加入网络后但未加入群组的节点是什么类型？

课程总结

- 本课程详细介绍了
  - 使用控制台对节点进行转换管理
  - 使用控制台扩容节点和退出网络

腾讯云

谢谢