

FISCO BCOS Tutorial

单群组FISCO BCOS联盟链的搭建

本节以搭建单群组FISCO BCOS链为例操作。使用build_chain.sh脚本在本地搭建一条4节点的FISCO BCOS链，以Ubuntu 16.04 64bit系统为例操作。

1、准备环境

安装依赖

```
sudo apt install -y openssl curl
```

创建操作目录

```
cd ~ && mkdir -p fisco && cd fisco
```

下载build_chain.sh脚本

```
curl -LO https://github.com/FISCO-BCOS/FISCO-BCOS/releases/download/`curl -s https://api.github.com/repos/FISCO-BCOS/FISCO-BCOS/releases | grep "\"v2\".[0-9]\".[0-9]\"\"" | sort -u | tail -n 1 | cut -d \" -f 4`/build_chain.sh && chmod u+x build_chain.sh
```

2、搭建单群组4节点联盟链

在fisco目录下执行下面的指令，生成一条单群组4节点的FISCO链。请确保机器的30300~30303，20200~20203，8545~8548端口没有被占用。

```
bash build_chain.sh -l "127.0.0.1:4" -p 30300,20200,8545
```

命令执行成功会输出All completed。如果执行出错，请检查nodes/build.log文件中的错误信息。

3、启动FISCO BCOS链

启动所有节点

```
bash nodes/127.0.0.1/start_all.sh
```

启动成功会输出类似下面内容的相应。否则请使用 `netstat -an | grep tcp` 检查机器的30300~30303，20200~20203，8545~8548端口是否被占用。

```
try to start node0
try to start node1
try to start node2
try to start node3
node1 start successfully
node2 start successfully
node0 start successfully
node3 start successfully
```

4、检查进程

检查进程是否启动

```
ps -ef | grep -v grep | grep fisco-bcos
```

正常情况会有类似下面的输出；如果进程数不为4，则进程没有启动（一般是端口被占用导致的）

```
fisco      5453      1  1 17:11 pts/0    00:00:02
/home/fisco/fisco/nodes/127.0.0.1/node0/./fisco-bcos -c config.ini
fisco      5459      1  1 17:11 pts/0    00:00:02
/home/fisco/fisco/nodes/127.0.0.1/node1/./fisco-bcos -c config.ini
fisco      5464      1  1 17:11 pts/0    00:00:02
/home/fisco/fisco/nodes/127.0.0.1/node2/./fisco-bcos -c config.ini
fisco      5476      1  1 17:11 pts/0    00:00:02
/home/fisco/fisco/nodes/127.0.0.1/node3/./fisco-bcos -c config.ini
```

5、检查日志输出

如下，查看节点node0链接的节点数

```
tail -f nodes/127.0.0.1/node0/log/log* | grep connected
```

正常情况会不停地输出链接信息，从输出可以看出node0与另外3个节点有链接。

```
ubuntu@VM-0-5-ubuntu:~/fisco$ tail -f nodes/127.0.0.1/node0/log/log* | grep
connected
info|2019-07-13 15:59:56.412634|[P2P][Service] heartBeat,connected count=3
info|2019-07-13 16:59:56.437505|[P2P][Service] heartBeat,connected count=3
info|2019-07-13 17:59:56.466274|[P2P][Service] heartBeat,connected count=3
info|2019-07-13 18:59:56.497670|[P2P][Service] heartBeat,connected count=3
info|2019-07-13 19:59:56.526350|[P2P][Service] heartBeat,connected count=3
info|2019-07-13 20:59:56.564834|[P2P][Service] heartBeat,connected count=3
```

执行下面指令，检查是否在共识

```
tail -f nodes/127.0.0.1/node0/log/log* | grep +++
```

正常情况会不停输出++++Generating seal，表示共识正常。

```
ubuntu@VM-0-5-ubuntu:~/fisco$ tail -f nodes/127.0.0.1/node0/log/log* | grep +++
info|2019-08-24 19:59:59.179225|[g:1][CONSENSUS][SEALER]++++++++++++++++
Generating seal on,blkNum=65,tx=0,nodeIdx=3,hash=bd180029...
info|2019-08-24 20:59:59.969556|[g:2][CONSENSUS][SEALER]++++++++++++++++
Generating seal on,blkNum=2,tx=0,nodeIdx=3,hash=01e48483...
info|2019-08-24 22:59:58.903671|[g:1][CONSENSUS][SEALER]++++++++++++++++
Generating seal on,blkNum=71,tx=0,nodeIdx=3,hash=606c1579...
info|2019-08-24 23:59:59.759694|[g:2][CONSENSUS][SEALER]++++++++++++++++
Generating seal on,blkNum=2,tx=0,nodeIdx=3,hash=aaf28074...
```

配置及使用控制台

1、准备依赖

安装openjdk

```
sudo apt install -y default-jdk
```

获取控制台并回到fisco目录

```
cd ~/fisco && bash <(curl -s https://raw.githubusercontent.com/FISCO-BCOS/console/master/tools/download_console.sh)
```

拷贝控制台配置文件

```
cp -n console/conf/applicationContext-sample.xml console/conf/applicationContext.xml
```

配置控制台证书

```
cp nodes/127.0.0.1/sdk/* console/conf/
```

2、启动和退出控制台

启动

```
cd ~/fisco/console && bash start.sh
```

输出下述信息表明启动成功。

```
=====
=====
welcome to FISCO BCOS console(1.0.4)!
Type 'help' or 'h' for help. Type 'quit' or 'q' to quit console.
_____
|      |      \ /      \ /      \ /      \      \ /      \ /      \ /
|      |      \
| $$$$$$\\$$$$$| $$$$$$| $$$$$$| $$$$$$\\      | $$$$$$| $$$$$$| $$$$$$|
$$$$$\\
| $$__      | $$ | $$__\\$| $$  \\$| $$  | $$  | $$__/$$| $$  \\$| $$  | $$
$$__\\$$
| $$  \\      | $$  \\$$  \\| $$      | $$  | $$  | $$      $| $$      | $$  | $$\\$$
\\
| $$$$$      | $$  _\\$$$$$| $$  __| $$  | $$  | $$$$$$| $$  __| $$  |
$$_\\$$$$$\\
| $$      _| $$_|  \\_| $| $$__/$$  | $$__/$$  | $$__/$$  | $$__/$$  |
\\_| $$
| $$      |  $$  \\$$  $$\\$$  $$\\$$  $$  | $$  $$\\$$  $$\\$$  $$\\$$  $$\\$$
  $$
\\$$  \\$$$$$ \\$$$$$ \\$$$$$ \\$$$$$  \\$$$$$ \\$$$$$ \\$$$$$
\\$$$$$
=====
=====
```

退出

```
[group:1]> quit
```

3、使用控制台获取信息

获取客户端版本

```
[group:1]> getNodeVersion
{
  "Build Time":"20190705 13:17:29",
  "Build Type":"Linux/clang/Release",
  "Chain Id":"1",
  "FISCO-BCOS Version":"2.0.0",
  "Git Branch":"HEAD",
  "Git Commit Hash":"d8605a73e30148cfb9b63807fb85fa211d365014",
  "Supported Version":"2.0.0"
}
```

获取节点链接信息

```
[group:1]> getPeers
[
  {
    "Agency":"agency",
    "IPAndPort":"127.0.0.1:55982",
    "Node":"node1",

    "NodeID":"2c17843f4d4fcbfd3e705fde6bd33c0fda5f3a1513976eb5798de1b5cb843a681d2ff313174a24a4784853d04908f9ff268681cdc3df637d0124139613482c9",
    "Topic":[

    ]
  },
  {
    "Agency":"agency",
    "IPAndPort":"127.0.0.1:30302",
    "Node":"node2",

    "NodeID":"29109de8c8b1dc9ae9738c942ce2da35ff8cb1546523e0f9f33850ff5e0f0a77bdb15871c16890b1ad3ba18f1d62f2ecda24e8c3b569467d336bc386906e4dad",
    "Topic":[

    ]
  },
  {
    "Agency":"agency",
    "IPAndPort":"127.0.0.1:55974",
    "Node":"node3",

    "NodeID":"4224784d298bc84ac787bda47d15ccd541635ed952d1333c7aa0aa809d40a606dea5d4d60cb4bf89b1252f4de37203370f5ce5a2148d2803da36c8f72a7ee427",
    "Topic":[

    ]
  }
]
```

查看当前块高

```
[group:1]> getBlockNumber  
102
```

部署及调用SimpleStorage合约

1、SimpleStorage合约

```
pragma solidity >=0.4.0 <0.7.0;  
  
contract SimpleStorage {  
    uint storedData;  
  
    function set(uint x) public {  
        storedData = x;  
    }  
  
    function get() public view returns (uint) {  
        return storedData;  
    }  
}
```

2、部署SimpleStorage合约

把SimpleStorage合约保存到/fisco/console/contracts/solidity/SimpleStorage.sol，并使用deploy命令部署。

```
# 在控制台输入以下指令 部署成功则返回合约地址  
[group:1]> deploy SimpleStorage  
contract address: 0x6de4fd2d0193cc139bfe0ffc2d335dd1bb9dbb02
```

3、调用SimpleStorage合约

```
# call [contract_name] [contract_address] [function_name] [parameter1 parameter2  
...]  
# 参数之间用空格隔开  
[group:1]> call SimpleStorage 0x6de4fd2d0193cc139bfe0ffc2d335dd1bb9dbb02 set 666  
transaction hash:  
0x257e36adf91027c6b636506d3393a8b4dd681a8afdd79069a349e37bd0dffc40  
  
[group:1]> call SimpleStorage 0x6de4fd2d0193cc139bfe0ffc2d335dd1bb9dbb02 get  
666
```

