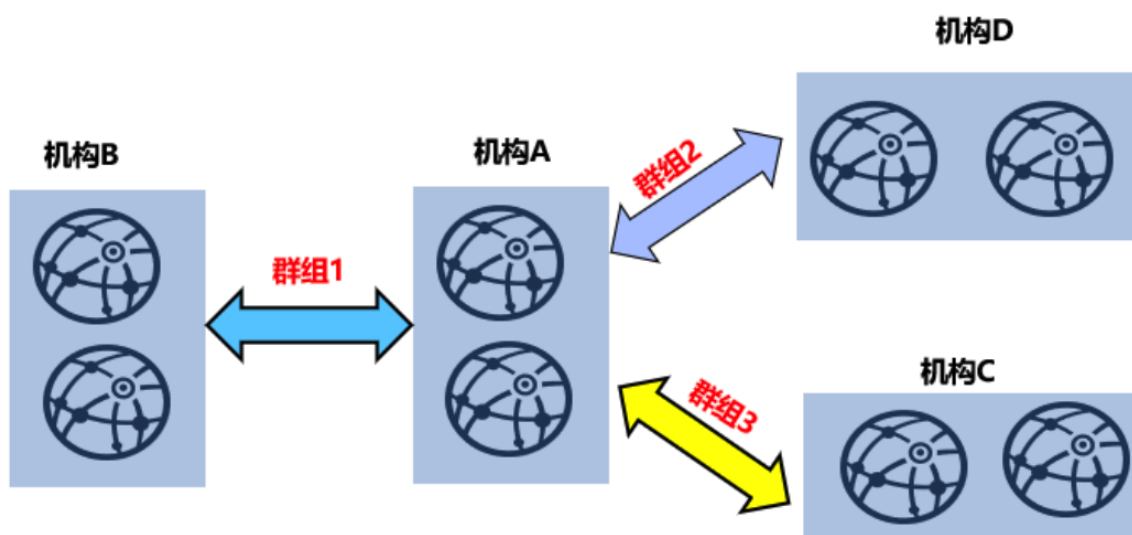


星形拓扑构建

星形拓扑：中心机构节点同时属于多个群组，运行多家机构应用，其他每家机构属于不同群组，运行各自应用；



本节构建上图所示的单机、四机构、三群组、八节点的星形组网拓扑。

agencyA: 在127.0.0.1上有2个节点，同时属于group1、group2、group3；
agencyB: 在127.0.0.1上有2个节点，属于group1；
agencyC: 在127.0.0.1上有2个节点，属于group2；
agencyD: 在127.0.0.1上有2个节点，属于group3。

第一步. 进入FISCO目录

以下命令，回到root目录下的fisco目录，后续的操作都将在此目录下进行

```
cd /root/fisco
```

第二步. 生成星形区块链系统配置文件

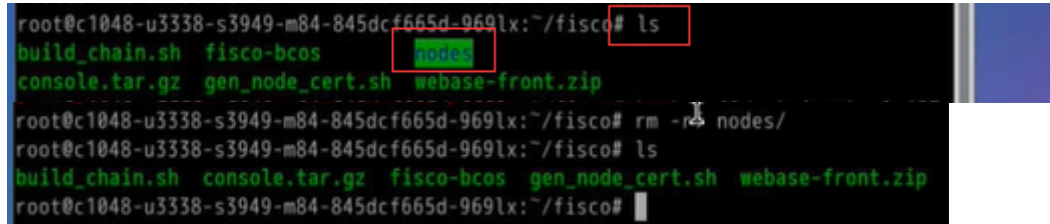
使用cat命令创建文件

```
cat > ipconf << EOF
127.0.0.1:2 agencyA 1,2,3
127.0.0.1:2 agencyB 1
127.0.0.1:2 agencyC 2
127.0.0.1:2 agencyD 3
EOF
```

第三步. 使用build_chain脚本构建星形区块链节点配置文件夹

在fisco目录下执行下面的指令，查看当前目录中是否存在nodes目录，如果存在，则删除此目录,如果不存在则跳过删除目录步骤。

```
ls
rm -rf nodes
```



A terminal window showing the execution of commands. The prompt is root@c1048-u3338-s3949-m84-845dcf665d-969lx:~/fisco#. The first command is ls, which lists the contents of the directory: build_chain.sh, fisco-bcos, console.tar.gz, gen_node_cert.sh, and webase-front.zip. The second command is rm -rf nodes/, which removes the nodes directory. The prompt then changes to root@c1048-u3338-s3949-m84-845dcf665d-969lx:~/fisco#.

使用刚才输入的构建配置文件

```
bash build_chain.sh -f ipconf -p 30300,20200,8545 -e ./fisco-bcos
```

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

```
Generating CA key...
=====
Generating keys ...
Processing IP:127.0.0.1 Total:2 Agency:agencyA Groups:1,2,3
Processing IP:127.0.0.1 Total:2 Agency:agencyB Groups:1
Processing IP:127.0.0.1 Total:2 Agency:agencyC Groups:2
Processing IP:127.0.0.1 Total:2 Agency:agencyD Groups:3
=====
.....此处省略其他输出.....
=====
[INFO] FISCO-BCOS Path      : ./fisco-bcos
[INFO] IP List File        : ipconf
[INFO] Start Port          : 30300 20200 8545
[INFO] Server IP           : 127.0.0.1:2 127.0.0.1:2 127.0.0.1:2 127.0.0.1:2
[INFO] State Type          : storage
[INFO] RPC listen IP        : 127.0.0.1
[INFO] Output Dir           : /root/fisco/nodes
[INFO] CA Key Path          : /root/fisco/nodes/cert/ca.key
=====
[INFO] All completed. Files in /root/fisco/nodes
```

第四步. 启动FISCO BCOS链

启动所有节点，bash为linux系统自带的命令处理工具，下面命令由bash解析执行脚本，启动fisco区块链

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

启动成功会输出类似下面内容的响应。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
try to start node4
try to start node5
try to start node6
try to start node7
node0 start successfully
node2 start successfully
node1 start successfully
node5 start successfully
node6 start successfully
node4 start successfully
node3 start successfully
node7 start successfully
```

第五步. 检查进程

检查进程是否启动。

Linux ps (英文全拼: process status) 命令用于显示当前进程的状态, 类似于 windows 的任务管理器。Linux grep 命令用于查找文件里符合条件的字符串。下面的命令意思是使用ps查看linux中正在运行的进程(ps -ef), 然后找到不包含grep的进程列表(grep -v grep), 再找到包含fisco-bcos的进程列表(grep fisco-bcos), 中间的竖线代表管道, 将上一个命令的输出, 转换为下一个命令的输入。

```
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8, 则进程没有启动 (一般是端口被占用导致的)

```
root      1213      1  3 18:42 pts/0    00:00:02
/root/fisco/nodes/127.0.0.1/node0/../../fisco-bcos -c config.ini
root      1214      1  1 18:42 pts/0    00:00:00
/root/fisco/nodes/127.0.0.1/node5/../../fisco-bcos -c config.ini
root      1218      1  1 18:42 pts/0    00:00:00
/root/fisco/nodes/127.0.0.1/node2/../../fisco-bcos -c config.ini
root      1219      1  1 18:42 pts/0    00:00:00
/root/fisco/nodes/127.0.0.1/node4/../../fisco-bcos -c config.ini
root      1223      1  3 18:42 pts/0    00:00:02
/root/fisco/nodes/127.0.0.1/node1/../../fisco-bcos -c config.ini
root      1227      1  1 18:42 pts/0    00:00:00
/root/fisco/nodes/127.0.0.1/node7/../../fisco-bcos -c config.ini
root      1228      1  1 18:42 pts/0    00:00:00
/root/fisco/nodes/127.0.0.1/node6/../../fisco-bcos -c config.ini
root      1233      1  1 18:42 pts/0    00:00:00
/root/fisco/nodes/127.0.0.1/node3/../../fisco-bcos -c config.ini
```

第六步. 查看群组共识状态

不发交易时，共识正常的节点会输出+++日志，可通过`tail -f node/log/ | grep "++"`查看各节点是否正常。

节点正常共识打印 +++ 日志，+++ 日志字段含义：

g:: 群组ID
blkNum: Leader节点产生的新区块高度;
tx: 新区块中包含的交易数目;
nodeIdx: 本节点索引;
hash: 共识节点产生的最新区块哈希。

查看node0 group1是否正常共识

```
tail -f nodes/127.0.0.1/node0/log/log* | grep "g:1.*++"
```

查看node0 group2是否正常共识

```
tail -f nodes/127.0.0.1/node0/log/log* | grep "g:2.*++"
```

结果输出：

```
info|2019-02-11 15:33:31.021697| [g:2][p:520][CONSENSUS][SEALER]+++++++Generating  
seal on,blkNum=1,tx=0,nodeIdx=3,hash=ef59cf17...
```

其他节点可参考以上操作方法...

本节任务

- 1 通过配置文件创建了星形多群组联盟链
- 2 我们查看了日志与共识情况

完成本节联系后，请点击“**下一步**”进入下一步骤实验。

控制台安装

第一步. 进入fisco目录

```
cd /root/fisco
```

第二步. 解压缩控制台安装包

```
tar zxvf console.tar.gz
```

第三步. 拷贝节点证书到控制台配置目录

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

第四步. 创建控制台配置文件

```
cp console/conf/config-example.toml console/conf/config.toml
```

第五步. 启动控制台，向群组发送交易

启动控制台

```
bash console/start.sh
```

向group1发交易

```
$ [group:1]> deploy HelloWorld
contract address:0x8c17cf316c1063ab6c89df875e96c9f0f5b2f744
# 查看group1当前块高，块高增加为1表明出块正常，否则请检查group1是否共识正常
$ [group:1]> getBlockNumber
1
```

第六步. 查看日志

节点每出一个新块，会打印一条Report日志，Report日志中各字段含义如下：

g: 群组ID
num: 出块高度;
sealerIdx: 共识节点索引;
hash: 区块哈希;
next: 下一个区块高度;
tx: 区块包含的交易数;
nodeIdx: 当前节点索引。

进入节点目录

```
cd ~/fisco/nodes/127.0.0.1
```

查看group3出块情况：有新区块产生

```
cat node0/log/* |grep "g:3.*Report"
```

日志输出参考如下：

```
info|2019-02-11 16:14:33.930978| [g:3][p:776][CONSENSUS]
[PBFT]^^^^^^^Report,num=1,sealerIdx=1,hash=3a42fcd1...,next=2,tx=1,nodeIdx=2
```

本节任务

- 1 安装控制台
- 2 使用控制台发起交易
- 3 查看日志输出情况

完成本节联系后，请点击“**下一步**”进入下一步骤实验。

节点加入群组

通过控制台，将node2加入group2

第一步. 进入节点目录

```
cd ~/fisco/nodes/127.0.0.1
```

第二步. 从node0拷贝group2的配置到node2

```
cp node0/conf/group.2.* node2/conf
```

第三步. 重启node2节点

```
cd node2 && bash stop.sh && bash start.sh
```

第四步. 获取node2的节点ID

```
cat conf/node.nodeid
```

终端输出参考如下：

```
6dc585319e4cf7d73ede73819c6966ea4bed74aadbbcbba1bbb777132f63d355965c3502bed7a04425d99
cdcfb7694a1c133079e6d9b0ab080e3b874882b95ff4
```

第五步. 回到控制台目录，并启动控制台（直接启动到group2）

```
cd ~/fisco/console && bash start.sh 2
```

第六步. 查看当前共识节点列表

```
$ [group:2]> getSealerList
[
  9217e87c6b76184cf70a5a77930ad5886ea68aefbcce1909bdb799e45b520baa53d5bb9a5edddeab9475
  1df179d54d41e6e5b83c338af0a19c0611200b830442,
  227c600c2e52d8ec37aa9f8de8db016ddc1c8a30bb77ec7608b99ee2233480d4c06337d2461e24c26617
  b6fd53acfa6124ca23a8aa98cb090a675f9b40a9b106,
  7a50b646fcd9ac7dd0b87299f79ccaa2a4b3af875bd0947221ba6dec1c1ba4add7f7f690c95cf3e79629
  6cf4adc989f4c7ae7c8a37f4505229922fb6df13bb9e,
  8b2c4204982d2a2937261e648c20fe80d256dfb47bda27b420e76697897b0b0ebb42c140b4e8bf0f27df
  ee64c946039739467b073cf60d923a12c4f96d1c7da6
]
```

第七步. 将node2加入到共识节点

addSealer后面的参数是上步获取的node ID

```
$ [group:2]> addSealer
6dc585319e4cf7d73ede73819c6966ea4bed74aadbbcbab1bbb777132f63d355965c3502bed7a04425d99
cdcfb7694a1c133079e6d9b0ab080e3b874882b95ff4
{
  "code":0,
  "msg":"success"
}
```

第八步. 再次查看共识列表

观察新加入节点是否在共识列表中

```
$ [group:2]> getSealerList
[
  9217e87c6b76184cf70a5a77930ad5886ea68aefbcce1909bdb799e45b520baa53d5bb9a5eddddeab9475
  1df179d54d41e6e5b83c338af0a19c0611200b830442,
  227c600c2e52d8ec37aa9f8de8db016ddc1c8a30bb77ec7608b99ee2233480d4c06337d2461e24c26617
  b6fd53acfa6124ca23a8aa98cb090a675f9b40a9b106,
  7a50b646fcd9ac7dd0b87299f79ccaa2a4b3af875bd0947221ba6dec1c1ba4add7f7f690c95cf3e79629
  6cf4adc989f4c7ae7c8a37f4505229922fb6df13bb9e,
  8b2c4204982d2a2937261e648c20fe80d256dfb47bda27b420e76697897b0b0ebb42c140b4e8bf0f27df
  ee64c946039739467b073cf60d923a12c4f96d1c7da6,
  6dc585319e4cf7d73ede73819c6966ea4bed74aadbbcbba1bbb777132f63d355965c3502bed7a04425d99
  cdcfb7694a1c133079e6d9b0ab080e3b874882b95ff4
]
```

第九步. 获取group2当前块高

```
$ [group:2]> getBlockNumber
2
```

本节任务

- 1 将node2加入group2群组
- 2 将node2加入到共识节点，并查看了共识情况

完成本节联系后，请点击"**下一步**"进入下一步骤实验。