

区块链底层搭建 2.0（本地搭建）

注意本文档需要在Linux操作系统下进行

推荐使用ubuntu20.04版本

可选择VMware Workstation Pro 17虚拟机进行安装

安装教程——>[ubuntu20.04安装](#)

1. 依赖安装

```
1 sudo su //进入root用户
```

1.1 Java 安装


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


1.2 Java 环境配置


```
1 sudo gedit /etc/profile
2
3 //进入配置文件
4 export JAVA_HOME=地址 //地址一般为 /usr/lib/jvm/java-1.11.0-openjdk-amd64
   or java-1.17.0-openjdk-amd64
5
6 export PATH=$JAVA_HOME/bin:$PATH
7 export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
8
9
```


2. 框架搭建


1. 前置要求——离线安装必备基础文件


 build_chain.sh


 console.tar.gz

 fisco.zip

 fisco-bcos

 fisco-bcos.tar.gz

 gen_node_cert.sh

 webase-front.zip

2. 搭建节点（单群组 4节点）

2.1 安装过程（命令行）

2.1.1 核心文件

fisco-bcos.tar.gz：区块链框架文件，需要解压

build_chain.sh：安装脚本文件

2.1.2 解压 fisco-bcos.tar.gz

参数	含义
tar	Linux压缩/解压缩命令
-z	代表gzip，使用gzip工具进行压缩或解压
-x	代表extract，解压文件（压缩文件是-c）
-v	代表verbose，显示解压过程（文件列表）
-f	代表file，指定要解压的文件名（or 要压缩成的文件名）

```
1 tar -zxvf fisco-bcos.tar.gz
```

2.1.3 使用脚本安装

```
1 bash build_chain.sh -l 127.0.0.1:4 -p 30300,20200,8545 -e ./fisco-bcos
```

- i. -L: ip地址:节点数量
- ii. -p: port端口号 分别是 p2p_port , channel_port , jsonrpc_port
- iii. -e: 表示离线安装 后面跟离线安装地址

安装成功:

```
akiira@akiira-machine:~/Desktop/fisco$ bash build_chain.sh -l 127.0.0.1:4 -p 30300,20200,8545 -e ./fisco-bcos
Checking fisco-bcos binary...
Binary check passed.
=====
Generating CA key...
=====
Generating keys and certificates ...
Processing IP=127.0.0.1 Total=4 Agency=agency Groups=1
=====
Generating configuration files ...
Processing IP=127.0.0.1 Total=4 Agency=agency Groups=1
=====
[INFO] FISCO-BCOS Path : ./fisco-bcos
[INFO] Start Port      : 30300 20200 8545
[INFO] Server IP       : 127.0.0.1:4
[INFO] Output Dir      : /home/akiira/Desktop/fisco/nodes
[INFO] CA Path          : /home/akiira/Desktop/fisco/nodes/cert/
[INFO] RSA channel     : true
=====
[INFO] Execute the download_console.sh script in directory named by IP to get FISCO-BCOS console
```

2.2 启动节点

使用bash 命令执行脚本文件

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

启动成功:

```
akiira@akiira-machine:~/Desktop/fisco$ 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
node1 start successfully
node3 start successfully
node0 start successfully
node2 start successfully
```

2.3 验证节点进程运行状况

LINUX命令详解:

使用ps, grep命令

Ps: process status(进程状态)

grep: globally search a regular expression and print (全局搜索正则表达式并打印) [查询命令](#)

ps命令参数

我们这里需要用到ef

命令	含义
e	显示所有进程,环境变量
f	全格式
h	不显示标题
l	长格式
w	宽输出
a	显示终端上地所有进程,包括其他用户地进程
r	只显示正在运行地进程
x	显示没有控制终端地进程
u	以用户为主的格式来显示程序状况
au	显示较详细的资讯
aux	显示所有包含其他使用者的行程
-C<命令>	列出指定命令的状况
--lines<行数>	每页显示的行数
--width<字符数>	每页显示的字符数
--help	显示帮助信息
--version	显示版本显示

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

成功查询到进程:

```
akiira@akiira-machine:~/Desktop/fisco$ ps -ef |grep -v grep|grep fisco-bcos
akiira 6829 1426 0 12:39 pts/0 00:00:07 /home/akiira/Desktop/fisco/nodes/127.0.0.1/node2/./fisco-bcos -c config.ini
akiira 6831 1426 0 12:39 pts/0 00:00:07 /home/akiira/Desktop/fisco/nodes/127.0.0.1/node1/./fisco-bcos -c config.ini
akiira 6833 1426 0 12:39 pts/0 00:00:07 /home/akiira/Desktop/fisco/nodes/127.0.0.1/node3/./fisco-bcos -c config.ini
akiira 6835 1426 0 12:39 pts/0 00:00:07 /home/akiira/Desktop/fisco/nodes/127.0.0.1/node0/./fisco-bcos -c config.ini
```

2.4 验证区块链连接状态和共识状态日志输出

使用tail，grep命令

LINUX命令详解：

`tail` 的意思就是「尾部、尾巴」默认会显示指定文件的末尾 10 行的内容。

`tail` 命令通常用于查看日志文件或实时监控日志文件的更新。

```
1 tail [选项参数] 文件名
```

- 常用选项参数
- `-n`：指定要显示的尾部行数。
- `-f`：（follow）持续跟踪显示文件尾部最新的内容，比如你查看日志文件的时候，当日志发生变化，就能实时看到日志的输出显示

1. 连接状态输出

实际上是去日志文件里面使用grep命令查询“`connected`”关键词

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

2. 共识日志输出

实际上是去日志文件里面使用grep命令查询“`+++`”关键词

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

3. 搭建节点（星形拓扑）

https://fisco-bcos-documentation.readthedocs.io/zh-cn/latest/docs/manual/group_use_cases.html

3.1 单机、四机构、三群组、八节点

1. ipconf配置文件编写

```
1 #生成区块链配置文件ipconf
2
3 单机：即一个主机（虚拟机）
4 四机构：agency
5
6 ip地址:节点数 agency 群组
7 $ cat > ipconf << EOF #EOF: End of File
8 127.0.0.1:2 agencyA 1,2,3
9 127.0.0.1:2 agencyB 1
10 127.0.0.1:2 agencyC 2
11 127.0.0.1:2 agencyD 3
12 EOF
```

2. 使用脚本搭建

```
1 bash build_chain.sh -f ipconf -p 30300,20200,8545
```

4. 搭建节点（并行多组）

1. 要求

本次实验将构建**四节点、两群组**的并行多组网络区块链，他们的组网关系如下：

群组1：包括四个节点，节点IP均为127.0.0.1

群组2：包括四个节点，节点IP均为127.0.0.1

2. 搭建

1. 使用启链脚本

```
1 bash build_chain.sh -l 127.0.0.1:4 -o multi-nodes -p 30300,20200,8545
-e ./fisco-bcos
```

2. 拷贝配置文件 并修改 group.2.genesis

group.1.genesis group.1.ini 拷贝为group.2.genesis group.2.ini

```
1 # 拷贝group1的配置
2 $ cp node0/conf/group.1.genesis node0/conf/group.2.genesis
3 $ cp node0/conf/group.1.ini node0/conf/group.2.ini
4
```

```
5 $ gedit group.2.genesis
6 # 将id = 1 修改为 id = 2
7
8 # 拷贝node0的配置 到其他节点
9 $cp node0/conf/group.2.genesis node0/conf/group.2.ini node1/conf/
10 $cp node0/conf/group.2.genesis node0/conf/group.2.ini node2/conf/
11 $cp node0/conf/group.2.genesis node0/conf/group.2.ini node3/conf/
```

3. 启动链，并且查看各个节点共识

```
1 # 启动链
2 $ bash start_all.sh
3
4 #查看各个节点共识
5 $ tail -f node0/log/log* |grep +++
6 $ tail -f node1/log/log* |grep +++
7 $ tail -f node2/log/log* |grep +++
8 $ tail -f node3/log/log* |grep +++
```

1. 搭建节点(docker方法)-4/13日更新

[使用docker部署区块链 - FISCO BCOS 2.0 v2.9.0 文档](#)

1.1 docker安装 (略)

```
1 apt install update
2 apt install docker.io
```

1.2 节点搭建

[build_chain.sh详解](#)

```
 bash build_chain.sh -l ip地址:节点数量 -d
```

1. 指定镜像版本

```
 bash build_chain.sh -l ip地址:节点数量 -d -v 版本号
```

```
1 e.g.  
2 bash build_chain.sh -v 2.8.0 -l 127.0.0.1 -d
```

1.3 运行节点

 `bash start_all.sh`

1. 执行后会自动创建docker容器

如何查看? —————>[docker使用教程](#)

 `docker ps`

2. 查看日志 (检查节点是否正常运行)

进入任意节点根目录 (e.g: node0)

1. 查询链接状态 connected

需要用到通配符 *

 `tail -f ./log/log* | grep connected`

2. 查询共识状态


 `tail -f ./log/log* | grep +++`

1.4 节点扩容

1. 需要证书生成脚本:

`gen_node_cert.sh` [gen_node_cert.sh](#) 详解

2. 使用脚本, 生成节点私钥证书

 `bash gen_node_cert.sh -c 证书路径 -o 节点私钥证书输出目录`

证书路径: `nodes(节点根目录) /cert/agency`

3. 配置文件拷贝

- 拷贝 `node0/config.ini`、`node0/start.sh` 和 `node0/stop.sh` 到新节点(`node4`)目录

```
cp node0/config.ini node0/start.sh node0/stop.sh ./node4
```

4. 修改节点配置文件 (config.ini)

- 进入新节点目录 (`node4`为例)
- 使用任意编辑器(`gedit/vi/vim` 皆可), 打开`config.ini`

```
gedit/vi/vim config.ini
```

- 修改配置文件

下面是以3节点添加节点为例, 其他数量请自行计算端口号:

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;

```
// config.ini 配置文件
[rpc]
channel_listen_port = 根据节点填写
.....
jsonrpc_listen_port = 根据节点填写
.....
[p2p]
listen_port = 根据节点填写

node.0 = 127.0.0.1: 30300
.....
```


`node.节点号 = 127.0.0.1: 根据节点填写` 此处是在后面添加, 非修改

5. 启动节点

- 进入新节点目录 (node4为例)

```
 bash start.sh
```

6. 检查节点是否扩容成功

```
 tali -f log/log* | grep connected
```

此处检查共识节点会查询不到，因为该节点目前未加入共识节点

共识节点添加需完成控制台搭建，下方为控制台搭建

5. 控制台搭建

5.1 安装过程 (命令行)

1. 核心文件

console.tar.gz：区块链框架文件，需要解压

2. 解压 console.tar.gz

方式和之前节点一样不多赘述

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

3. 控制台目录结构

```
console
├── apps #控制台jar包目录
│   └── console.jar
├── conf #配置文件存放目录
│   ├── config-example.toml #配置文件模板
│   ├── group-generate-config.toml
│   └── log4j2.xml #日志配置文件
├── console.sh
├── contracts
│   ├── console #控制台部署合约时，编译的合约abi,bin,java文件目录
│   ├── sdk #sol2java脚本，编译的合约abi,bin,java文件目录
│   └── solidity #solidity合约存放目录
│       ├── Crypto.sol
│       ├── HelloWorld.sol
│       ├── KVTableTest.sol
│       ├── ShaTest.sol
│       ├── Table.sol
│       └── TableTest.sol
├── get_account.sh #账户生成脚本
├── get_gm_account.sh #账户生成脚本，国密版
├── lib #相关依赖的jar包目录
│   ├── bcprov-jdk15on-1.60.jar
│   ├── commons-cli-1.3.1.jar
│   └── .....
```

4. 配置文件准备

使用cp命令

- -a：此选项通常在复制目录时使用，它保留链接、文件属性，并复制目录下的所有内容。其作用等于 dpR 参数组合。
- -d：复制时保留链接。这里所说的链接相当于 Windows 系统中的快捷方式。
- -r 或 --recursive：用于复制目录及其所有的子目录和文件，如果要复制目录，需要使用该选项。
- -i 或 --interactive：在复制前提示确认，如果目标文件已存在，则会询问是否覆盖，回答 y 时目标文件将被覆盖。。
- -u 或 --update：仅复制源文件中更新时间较新的文件。
- -v 或 --verbose：显示详细的复制过程。
- -p 或 --preserve：保留源文件的权限、所有者和时间戳信息。
- -f 或 --force：强制复制，即使目标文件已存在也会覆盖，而且不给出提示。

1. 拷贝控制台配置文件 都在conf下

使用cp -n命令可以完成文件复制和重命名的操作，这对于我们多次复制文件并需要重命名时很方便。即使目标文件已经存在，使用cp -n命令也可以创建一个新的文件副本。

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

2. 拷贝控制台证书

- `-r` 或 `--recursive` : 用于复制目录及其所有的子目录和文件, 如果要复制目录, 需要使用该选项。

```
1 cp -r nodes/127.0.0.1/sdk/* console/conf/
```

5. 启动并使用控制台

```
1 bash console/start.sh
2
3
4 #可以指定群组 and 账户
5 格式:
6 ./start.sh groupID -pem xxx.pem
```

6. 控制台命令

6.1 部署合约

使用deploy命令

```
1 deploy HelloWorld
```

6.2 调用合约

```
1 call 【合约名】 【合约地址】 【调用方法】
```

6.3 查看区块高度

```
1 getBlockNumber
```

3. 节点扩容（离线）

1. 前置要求

接下来的操作都在 `nodes/127.0.0.1` 目录下进行

证书生成脚本：

```
gen_node_cert.sh
```

2. 复制证书文件

1.复制cert.conf到nodes/cert/agency/channel/

```
1 cp -r nodes/cert.conf nodes/cert/agency/channel/
```

3. 生成新节点私钥证书

bash命令 -c -o

- 1 -c指定机构证书及私钥所在路径#
- 2 -o输出到指定文件夹，其中node4/conf中会存在机构agency新签发的证书和私钥
- 3
- 4 bash gen_node_cert.sh -c ../cert/agency -o node4

```
root@akiira-machine:/home/akiira/Desktop/fisco/nodes/127.0.0.1# bash gen_node_cert.sh -c ../cert/agency -o node4
=====
[INFO] Cert Path   : ../cert/agency
[INFO] Output Dir  : node4
[INFO] RSA cert    : true
[INFO] RSA SDK cert : true
=====
```

3. 准备节点配置文件

3.1 拷贝

`node0/config.ini`、`node0/start.sh` 和 `node0/stop.sh` 到node4目录;

```
1 cp node0/config.ini node0/start.sh node0/stop.sh node4/
```

拷贝 `node0/conf/group.1.genesis` (内含群组节点初始列表) 和 `node0/conf/group.1.ini` 到 `node4/conf` 目录下, 不需改动

```
1 cp node0/conf/group.1.genesis node0/conf/group.1.ini node4/conf/
```

3.2 修改配置文件

修改 `node4/config.ini`

【rpc】

1. 修改 `channel_listen_port=`

2. 修改 `jsonrpc_listen_port=`

【p2p】

3. 修改 `listen_port=`

4. 添加 `node.4=127.0.0.1:30304`

3.3 重启所有节点

```
1 bash stop_all.sh && bash start_all.sh
```

3.4 检查节点

```
1 tail -f node4/log/log* | grep connected
```

3.5 节点加入群组

1. 获取node4的nodeid

LINUX命令详解:

`cat`: concatenate files and print (连接文件并输出)

`cat` 命令用于连接一个或多个文件, 并将结果输出到终端或其他文件, 这样我们就能在终端直接看到输出的内容。 `cat` 命令适合查看内容较少、纯文本的文件。

```
1 cat [选项参数] 文件名
```

- 常用选项参数
- `-n` (`--number`) : 显示行号。
- `-b` (`--number-nonblank`) : 显示行号, 空行没有行号。
- `-s` (`--squeeze-blank`) : 多个空行压缩合并成一行。
- `-T` (`--show-tabs`) : 在文件中显示制表符 (将 TAB 字符显示为 `^I` 符号)

```
1 cat node4/conf/node.nodeid
```

2. 使用控制台将node4加入群组1

```
1 //启动控制台
2 bash start.sh
```

3. 使用addObserver将node4作为观察节点加入群组1

addObserver +nodeid

```
1 //控制台
2 getObserverList //查看观察节点
3
4 addObserver nodeid //添加观察节点
```

4. 使用addSealer将node4作为共识节点加入群组1

```
1 getSealerList //查看观察共识节点
2
3 addSealerList nodeid //添加共识节点
```

4. 节点退网

说明: 群组扩容需要先加入区块链网络, 再加入群组; 相反, 节点退网需要先退出群组, 再退出区块链网络, 否则可能会导致节点异常

4.1 退出群组（设置为游历节点）

```
1 #以下操作都在控制台进行
2
3 removeNode nodeid
4
5 //查看是否在共识列表内部,不在则成功
6 getSealerList
```

4.2 退出区块链网络

修改config.ini文件

```
1 #下面操作在终端进行
2
3 $gedit node4/config.ini
4
5 #下面是gedit页面
6 ; node to connect
7 //用 ; 注释掉下面 node.x
```

```
1 #重启节点
2 $bash stop_all.sh && start_all.sh
3
4 #查看连接状态 为0则成功
5 $tail -f node4/log/log*|grep connected
6
```

5. 区块链运维

1. 区块链节点运维

节点配置文件：config.ini

- i. 修改配置文件节点输出等级为警告级，日志存储阈值为100mb

```
1 //config.ini
2
```



```
3 [log]
4     level = xxx //文件节点输出等级
5     max_log_file_size= xxx //日志存储阈值
```

i. 验证

```
1 //shell
2
3 //重启所有节点
4 bash stop_all.sh
5 bash start_all.sh
6
7
8 //du指令主要用于统计文件和目录所占用的磁盘空间。
9 /*显示以人类可读的格式:
10 *     du -h [目录路径]
11 */  该指令将以人类可读的格式显示磁盘空间使用情况，将使用的容量转换为易于理解的单位
      (如GB、MB) 。
12 du -h node0/config.ini
```

2. 区块链网络运维

前置要求:

- 还是基于 nodes/127.0.0.1/node x/config.ini 文件内
- 需要获取目标节点的nodeid

```
1 cat node[填写节点号]/conf/node.nodeid
```

2.1 certificate_blacklist (黑名单设置)

https://fisco-bcos-documentation.readthedocs.io/zh-cn/latest/docs/design/security_control/certificate_list.html

//crl.X为本节点拒绝连接的对方节点NodeID。

每个节点的config.ini 都需要设置

```
1 //config.ini
2
3 [certificate_blacklist]
```

```
4 curl.1=af57c506be9ae60df8a4a16823fa948a68550a9b6a5624df44afcd3f75ce3afc6bb  
1416bcb7018e1a22c5ecbd016a80ffa57b4a73adc1aeaff4508666c9b633a
```

6. 账户权限控制

[https://fisco-bcos-documentation.readthedocs.io/zh-cn/latest/docs/manual/permission_control.html?](https://fisco-bcos-documentation.readthedocs.io/zh-cn/latest/docs/manual/permission_control.html?highlight=%E8%B4%A6%E6%88%B7%E6%9D%83%E9%99%90%E6%8E%A7%E5%88%B6)

[highlight=%E8%B4%A6%E6%88%B7%E6%9D%83%E9%99%90%E6%8E%A7%E5%88%B6](https://fisco-bcos-documentation.readthedocs.io/zh-cn/latest/docs/manual/permission_control.html?highlight=%E8%B4%A6%E6%88%B7%E6%9D%83%E9%99%90%E6%8E%A7%E5%88%B6)

6.1 生成账户

1. 通过get_account.sh脚本生成账户

1.1 生成 PEM 格式存储的账户

```
1 #无需任何参数, 因为默认生成pem格式  
2 $bash get_account.sh  
3  
4 #计算账户地址, 使用-f参数  
5 $bash get_account.sh -k account/xxx.pem
```

1.2 生成 PKCS12 格式存储的账户

```
1 #需要参数-p, 并且需要输入密码  
2 $bash get_account.sh -p  
3  
4 #计算账户地址, 使用-P参数  
5 bash get_account.sh -P account/xxx.p12
```

2. 通过控制台生成账户

2.1 生成 PEM 格式存储的账户

```
1 #同样无需任何参数, 因为默认生成pem格式  
2 newAccount
```

2.2 生成 PKCS12 格式存储的账户

```
1 #格式: newAccount p12 password
2 $newAccount p12 123123
```

3. 使用生成的账户进入控制台

```
1 #在控制台目录下
2 $bash start.sh 群组号 -加密格式pem/p12 xxx.加密格式
3
4 e.g
5 bash start.sh 1 -pem xxx.pem
6
7 #如果为p12需要输入密码
```

7. Webase-front搭建

7.1 解压webase-front.zip文件

```
1 unzip webase-front.zip
```

7.2 复制证书到配置目录下

```
1 cp -r nodes/127.0.0.1/sdk/* webase-front/conf/
```

8. webase搭建-非可视化

1. 依赖安装

1. PyMySQL部署

```
1 sudo apt-get install -y python3-pip //py3安装
2 pip3 install PyMySQL //PyMySQL安装
```

1.1 安装mysql数据库

```
1 sudo apt-get install aptitude
2 sudo aptitude install mysql-server-5.6
```

1.2 Java 环境

```
1 sudo gedit /etc/profile
```

2. 配置mysql

进入MySQL，

如果密码不正确

```
1 ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password';
```

建立webase账户，密码123456

```
1 CREATE USER 'webase'@'localhost' IDENTIFIED BY '123456';
```

建立两个数据库

```
1 create DATABASE webasenodemanager;
2 create DATABASE webasesign;
```

将两个数据库所有权限赋予本地的webase账号

```
1 GRANT ALL ON webasenodemanager.* TO 'webase'@'localhost';
2 GRANT ALL ON webasesign.* TO 'webase'@'localhost';
```

3. 修改配置文件

```
1 $cd webase-deploy
2 #进入common.properties
3 # 将mysql.user和mysql.password改为webase和123456
4 # Mysql database configuration of WeBASE-Node-Manager 修改相应的MySQL
5 用户名密码等信息
6 mysql.ip=localhost
7 mysql.port=3306
8 mysql.user=webase #修改此处
9 mysql.password=123456 #修改此处
10 mysql.database=webasenodemanager
11 sign.mysql.ip=localhost
12 sign.mysql.port=3306
13 sign.mysql.user=webase #修改此处
14 sign.mysql.password=123456 #修改此处
15 sign.mysql.database=webasesign
```

4. 一键部署

```
1 $python3 deploy.py startAll
2
```

注意一定要先去webase-deploy/common.properties下看端口号，小心有坑
看了端口号再去部署节点

节点管理服务忘记密码

登录到WeBASE-Node-Manager中配置的Mysql数据库（默认为webasenodemanager）后，在 `tb_account_info` 中插入一条新的管理员账号 `test`，密码 `Abcd1234`

<https://webasedoc.readthedocs.io/zh-cn/latest/docs/WeBASE-Node-Manager/appendix.html>

a. 检查进程端口

通过 `netstat` 命令，检查节点与节点前置的端口监听情况

检查方法如下，若无输出，则代表进程端口监听异常，需要到 `webase-front/log` 中查看日志的错误信息，并根据错误提示或根据[WeBASE-Front常见问题](#)进行错误排查

检查节点channel端口(默认为20200)是否已监听

b. 5000 端口密码不正确

需要去webase-deploy下搜索account-pwd

重航区

重航区块链竞赛组

重航区块链竞赛组

区块链竞赛组