

SDChain-Core 节点接入流程

1 环境要求

操作系统: Linux ubuntu 4.2.0-27-generic #32~14.04.1-Ubuntu SMP Fri Jan 22 15:32:26 UTC 2016

x86_64 x86_64 x86_64 GNU/Linux

编译环境: gcc version 5.4.1 20160904 (Ubuntu 5.4.1-2ubuntu1~14.04)

资源限制:

core file size	(blocks, -c) 0
data seg size	(kbytes, -d) unlimited
scheduling priority	(-e) 0
file size	(blocks, -f) unlimited
pending signals	(-i) 31449
max locked memory	(kbytes, -l) 64
max memory size	(kbytes, -m) unlimited
open files	(-n) 1024
pipe size	(512 bytes, -p) 8
POSIX message queues	(bytes, -q) 819200
real-time priority	(-r) 0
stack size	(kbytes, -s) 8192
cpu time	(seconds, -t) unlimited
max user processes	(-u) 31449
virtual memory	(kbytes, -v) unlimited
file locks	(-x) unlimited

2 软件下载

<https://github.com/SDChain/SDChain-Core/bin>

下载 sdchaind.tar.gz 文件

解压后包含如下文件:

sdchaind

SDChain-Core.cfg

validators.txt

libprotobuf.so.8

libstdc++.so.6

3 安装部署

3.1 部署可执行程序目录

```
mkdir /usr/local/ sdchaind
```

```
mv sdchaind /usr/local/ sdchaind
```

3.2 部署配置文件目录

```
mkdir /etc/opt/ sdchaind
```

```
mv SDChain-Core.cfg validators.txt /etc/opt/ sdchaind
```

3.3 部署数据库文件目录

```
mkdir /var/lib/sdchaind/db
```

3.4 部署日志文件目录

```
mkdir /var/log/ sdchaind
```

3.5 部署依赖库文件目录

```
mv libprotobuf.so.8 /usr/lib/x86_64-linux-gnu/libprotobuf.so.8
```

```
mv libstdc++.so.6 /usr/lib/x86_64-linux-gnu/libstdc++.so.6
```

4 配置参数

编辑 SDChain-Core.cfg 文件

```
[server]
port_rpc_admin_local
port_peer
port_ws_admin_local
port_ws_public
#ssl_key = /etc/ssl/private/server.key
#ssl_cert = /etc/ssl/certs/server.crt

[port_rpc_admin_local]
port = 5005
ip = 127.0.0.1
admin = 127.0.0.1
protocol = http

[port_peer]
port = 51235
ip = 0.0.0.0
protocol = peer
```

[port_ws_admin_local]

port = 6006

ip = 0.0.0.0

admin = 0.0.0.0

protocol = ws

[port_ws_public]

port = 6007

ip = 0.0.0.0

admin = 0.0.0.0

protocol = wss

[ledger_history]

full

[node_size]

medium

[node_db]

type=RocksDB

path=/var/lib/sdchaind/db/rocksdb

open_files=2000

filter_bits=12

cache_mb=256

file_size_mb=8

file_size_mult=2

#online_delete=2000

advisory_delete=0

[database_path]

/var/lib/ sdchaind /db

[debug_logfile]

/var/log/ sdchaind /debug.log

[ntp_servers]

time.windows.com

time.apple.com

time.nist.gov

pool.ntp.org

[ips]

genesis.sdchain.io 51235 //目前测试的节点

```
[validation_seed]
snVNTbPZwkNPNYoFmMSYg6FbaZmF7

[validators_file]
validators.txt

[validation_quorum]
3

[rpc_startup]
{ "command": "log_level", "severity": "warning" }

[ssl_verify]
0
```

编辑 validators.txt 文件

```
# Public keys of the validators that this sdchaind instance trusts.
[validators]
n9M2JkDT2WWAo1iaPTag9rxjGQDwC7dnnvyxUfVsZNrAe2CPnu6p
```

5 启动运行

5.1 普通模式启动

```
./sdchaind
```

首次启动，选用这种模式，会从六域链 SDChain-Core 区块链的其他网络上节点同步初始化历史账本信息。

5.2 加载启动模式

```
./sdchaind -load
```

再次启动，选用这种启动模式，首先会从本地初始化历史账本信息，然后在和网络上同步。

5.3 单机启动模式

```
./sdchaind -a
```

单机调试，选用这种模式。不会连接到六域链 SDChain-Core 其他的公开节点网络上。

5.4 关闭服务

```
./sdchaind stop
```

5.5 验证是否启动成功

```
./sdchaind
```

5.6 验证

执行如下命令：

```
./sdchaind peers
```

返回如下响应信息：

```
{
  "id" : 1,
  "result" : {
    "cluster" : {},
    "peers" : [
      {
        "address" : "45.76.69.152:44951",
        "complete_ledgers" : "1 - 115670",
        "inbound" : true,
        "latency" : 319,
        "load" : 170,
        "public_key" : "n9LFnDTLjwAnHF9zUtkkATW5cPRMqKJS7u6x6zcmtzdSjKtszsf2",
        "uptime" : 16,
        "version" : "sdchaind-0.60.2"
      }
    ],
    "status" : "success"
  }
}
```

注释：

如果 `peers` 中有其它的 SDChain-Core 节点服务器的信息数据，说明已经成功连接上区块链公共服务网络。