SDChain-Core节点搭建

1. **环境要求**

操作系统：Ubuntu 14.04\_64、 Ubuntu 16.04\_64

CPU：3GHz+ processor with 4 cores

内存：2G+

硬盘：40G+

编译环境：gcc version 5.4.1

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 |

1. **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

libprotobuf.so.8.0.0

libstdc++.so.6

start.sh

start-first.sh

1. **安装部署**
   1. **部署可执行程序目录**

mkdir /usr/local/ sdchaind

mv sdchaind /usr/local/ sdchaind

* 1. **部署配置文件目录**

mkdir /etc/opt/ sdchaind

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

* 1. **3.3部署数据库文件目录**

mkdir /var/lib/sdchaind/db

* 1. **部署日志文件目录**

mkdir /var/log/ sdchaind

* 1. **部署依赖库文件目录**

mv libprotobuf.so.8 /usr/lib/x86\_64-linux-gnu/libprotobuf.so.8

mv libprotobuf.so.8.0.0 /usr/lib/x86\_64-linux-gnu/libprotobuf.so.8.0.0

mv libstdc++.so.6 /usr/lib/x86\_64-linux-gnu/libstdc++.so.6

1. **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  [sntp\_servers]  time.windows.com  time.apple.com  time.nist.gov  pool.ntp.org  [ips]  test-node1.sdchain.io 51266 //目前测试的节点  [validation\_seed]  shqaNQbXwytxSfA9Suh6nFrhB169b  [validators\_file]  validators.txt  [validation\_quorum]  3  [rpc\_startup]  { "command": "log\_level", "severity": "warning" }  [ssl\_verify]  0 |

编辑validators.txt文件

|  |
| --- |
| [validators]  n9MLZmN5AJhfE3okEtRtMYNAKAmjWfTYPYDNdCQfZwTkDYursM5j |

1. **启动运行**
   1. **普通模式启动**

./sdchaind –net

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。

* 1. **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。

* 1. **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。

* 1. **脚本启动模式**

使用chmod +x start-first.sh start.sh命令添加可执行权限给启动脚本start-first.sh和start.sh。

第一次启动使用./start-first.sh命令执行脚本。再次启动使用./start.sh命令执行脚本

* 1. **关闭服务**

./sdchaind stop

* 1. **验证是否启动成功**

执行如下命令：

./sdchaind peers

返回如下响应信息：

|  |
| --- |
| {  "id" : 1,  "result" : {  "cluster" : {},  "peers" : [  {  "address" : "217.69.3.57:51266",  "complete\_ledgers" : "1 - 609456",  "latency" : 9,  "ledger" : "D08105558B6261A6237DDBADCAA1B64E5D8C0F13F475C6B962940DAEF486AE41",  "load" : 18,  "public\_key" : "n9Mq2U2BPesJkr3K4s1g9GuTKfv44NxxYzbs7RoGCfVXArUynMnr",  "uptime" : 130601,  "version" : "sdchaind-0.3.10-b"  }  ],  "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。