本地搭建SAVE网络

SAVE Network 本地开发环境配置(SOLO共识)

> 本配置用于说明如何在本地环境搭建SAVE网络，为了方便测试，主链采用solo共识。同时为了简化网络拓扑，不引入NAT代理节点。本文档是在Mac OS系统下运行, 其他操作系统可能在个别命令会有差别。

1.代码编译

运行环境

- go version go1.16.5
- 如果不使用 `Go Module` 则输入命令 `$ go env -w GO111MODULE="off"`， 同时下载各依赖模块
- 使用 `Go Module` ， 修改以下配置

```
$ git config --global url."http://10.0.1.228:3000/saveio".insteadOf  http://10.0.1.228:3000/saveio
$ go env -w GONOPROXY=github.com/saveio
$ go env -w GONOSUMDB=github.com/saveio
$ go env -w GOPRIVATE="github.com/saveio"
$ go env -w GOPROXY="https://goproxy.io,direct"
```

各代码版本

Aa 仓库	CommitId	说明
themis	41ad0ce75fa3d76d6b3ff66f43b57bf29051e0f6	主链
themis-go-sdk	59432db5c30793283349b103c932b4a0eede8755	主链RPC SDK
carrier	9fc4d908c3850ceb1bec6445cc2825df56dbb347	P2P网络
scan	a0b7871af66c983d18c002d20da351df9f02c6a6	DNS
pylons	78a1ef20d8a0dfc16d2301eaae2dc0257cc67ae2	Layer2支付通道
max	139def04befb758949a40fef65b822aa440e6bd6	FS存储
edge	d56fb27056c347b73f5403d18149c72a1291e648	DSP客户端

仓库	CommitId	说明
dsp-go-sdk	a3d6f6d46c55a43ccf2de63487f474878af9c59b	DSP SDK
porter	03916968865b1694f3a8ead0b1bdb0576bd3b6cd	NAT代理

编译

核心代码克隆完成之后，还需要克隆其他依赖代码。各依赖可以在内网编译机器直接下载 （内网机器 10.0.1.138）

依赖所在位置 `~/publish_build/golang/src`

编译参考各项目的说明，基本是`make`方式编译。

2.主链配置

创建钱包

```
$ cd ~/go/src/github.com/saveio/themis
$ go run main.go account add -d
```

```
输出:

Index:1
Label:
Address:AaK1g8jcH8zRMmN8F8oDaVteMXpit42pjh
Id(For PoC):7163600173146005825
Public key:03977eda4ea3db5757ad1977eaca6172ab3f197c19f06d2361769a6621941c48e1
Signature scheme:SHA256withECDSA
Create account successfully.
```

启动

```
$ cd ~/go/src/github.com/saveio/themis
$ go run main.go --testmode -p pwd
```

查看余额(另起新的窗口)

```
$ cd ~/go/src/github.com/saveio/themis
$ go run main.go asset balance 1
```

```
输出：

BalanceOf:AaK1g8jcH8zRMmN8F8oDaVteMXpit42pjh
  USDT:100000000
```

3.DNS节点配置

> 由于有数据库文件产生，这里需要先编译成可执行文件

创建钱包

```
$ cd ~/go/src/github.com/saveio/scan
$ make
$ ./scan accound add -d
```

```
输出:

Index:1
Label:
Address:AYFzNEzNH4KCGw7XS9N9C9oD6VbYobK5f7
Id(For PoC):5654908485478275393
Public key:03ce349d0f6d6913e3a22f5191a7c751cbeedda9ab42ceed217ea8d083cc00dba5
Signature scheme:SHA256withECDSA
Create account successfully.
```

获取测试币(另起新的窗口)

到`themis`节点处给 `dns` 节点转账

```
$ cd ~/go/src/github.com/saveio/themis
$ go run main.go asset transfer --from=AaK1g8jcH8zRMmN8F8oDaVteMXpit42pjh --to=AYFzNEzNH4KCGw7X
S9N9C9oD6VbYobK5f7 --amount=100000
```

```
输出：

Transfer USDT
  From:AaK1g8jcH8zRMmN8F8oDaVteMXpit42pjh
  To:AYFzNEzNH4KCGw7XS9N9C9oD6VbYobK5f7
  Amount:100000
  TxHash:edf053f2855c3890f3ca6f67952e21346e25e6ad66430fbf46fcf4bea7792e9c

Tip:
  Using './themis info status edf053f2855c3890f3ca6f67952e21346e25e6ad66430fbf46fcf4bea7792e9c'
to query transaction status.
```

等待1个区块高度，查询是否到账

```
$ cd ~/go/src/github.com/saveio/themis
$ go run main.go asset balance AYFzNEzNH4KCGw7XS9N9C9oD6VbYobK5f7
```

```
输出:

BalanceOf:AYFzNEzNH4KCGw7XS9N9C9oD6VbYobK5f7
  USDT:100000
```

配置文件

```
$ cd ~/go/src/github.com/saveio/scan
$ vim scan_config.json
```

```
{
        "Base": {
                "DumpMemory": false,
                "BaseDir": ".",
                "LogLevel": 0,
                "PublicIP": "127.0.0.1",
                "IntranetIP": "127.0.0.1",
                "PortBase": 10000,
                "LocalRpcPortOffset": 337,
                "EnableLocalRpc": false,
                "JsonRpcPortOffset": 336,
                "EnableJsonrpc": true,
                "HttpRestPortOffset": 335,
                "HttpCertPath": "",
                "HttpKeyPath": "",
                "EnableRest": true,
                "ChainRpcAddr": "http://127.0.0.1:20336",
                "ChainRestAddr": "http://127.0.0.1:20334",
                "DisableChain": true,
                "ChannelNetworkId": 1565267317,
                "ChannelPortOffset": 338,
                "ChannelProtocol": "tcp",
                "ChannelClientType": "rpc",
                "ChannelRevealTimeout": "20",
                "ChannelSettleTimeout": "50",
                "ChannelDBPath": "./ChannelDB",
                "DnsNetworkId": 1565511150,
                "DnsProtocol": "tcp",
                "DnsPortOffset": 339,
                "DnsGovernDeposit": 10000000000,
                "DnsChannelDeposit": 1000000000,
                "AutoSetupDNSRegisterEnable": true,
                "TrackerProtocol": "udp",
                "TrackerPortOffset": 6369,
                "TrackerPeerValidDuration": "-24h",
                "NATProxyServerAddr": "",
                "DBPath": "DB",
                "DnsNodeMaxNum": 100,
                "SeedInterval": 10,
                "IgnoreConnectDNSAddrs": [
                ]
        }
}
```

配置文件说明

配置项	说明	示例
DumpMemory	是否开启Profile进行性能记录	false
BaseDir	数据库文件所在目录(windows系统，请修改此项)	.
LogLevel	日志等级	0
PublicIP	固定外部IP	127.0.0.1
IntranetIP	内网IP	127.0.0.1
PortBase	各服务端口基准端口，如下面本地端口为10000+337即10337	10000
LocalRpcPortOffset	本地RPC端口	337
EnableLocalRpc	是否开启本地RPC	false
JsonRpcPortOffset	JSON-RPC端口	336
EnableJsonrpc	是否使用JSON-RPC	true
HttpRestPortOffset	REST 端口	335
HttpCertPath	HTTPS 证书路径	""
HttpKeyPath	HTTPS key 证书路径	""
EnableRest	是否开启 REST 服务	true
ChainRpcAddr	主链RPC地址，如果DNS为全节点， RPC地址为本地JSON-RPC地址	http://127.0.0.1:20336

配置项	说明	示例
ChainRestAddr	主链REST地址，如果DNS为全节点， REST地址为本地REST地址	http://127.0.0.1:20334
DisableChain	是否启动全节点， 如果关闭全节点功能，上面的主链RPC地址需要配成远端RPC节点的地址	true
ChannelNetworkId	Channel 网络ID	1565267317
ChannelPortOffset	Channel网络端口	338
ChannelProtocol	Channel网络协议	tcp
ChannelClientType	Channel请求主链方式	rpc
ChannelRevealTimeout	Channel Reveal Timeout， 目前建议使用200个区块高度	200
ChannelDBPath	Channel 数据库路径	./ChannelDB
DnsNetworkId	DNS网络ID	1565511150
DnsProtocol	DNS网络协议	tcp
DnsPortOffset	DNS端口	339
DnsGovernDeposit	DNS治理抵押，如果默认钱包余额充足，会自动抵押；否则，可使用命令手动抵押.	1000000000
DnsChannelDeposit	DNS通道抵押	1000000000
AutoSetupDNSRegisterEnable	是否开启DNS自动注册	true
TrackerNetworkId	Tracker网络ID	1567481543
TrackerProtocol	Tracker协议	tcp
TrackerPortOffset	Tracker端口	6369
TrackerPeerValidDuration	Tracker文件信息失效时间	-24h
NATProxyServerAddr	NAT proxy地址， 目前DNS有固定外部IP, 不使用NAT服务	""
DBPath	DNS数据库路径	DB
DnsNodeMaxNum	保留字段	100
SeedInterval	保留字段	10
DnsChannelDeposit	DNS通道抵押的资产数量	1000000000
AutoSetupDNSRegisterEnable	是否开启DNS自动注册	true
AutoSetupDNSChannelsEnable	是否自动创建通道	false
InitDeposit	保留字段	1000000000
ChannelDeposit	保留字段	1000000

配置项	说明	示例
IgnoreConnectDNSAddrs	忽略连接其他DNS的钱包地址	[]

启动

```
$ cd ~/go/src/github.com/saveio/scan
$ ./scan --config scan_config.json -p pwd
```

查询节点是否成功注册（可跳过)

> 如果查询为空，则重启scan节点，重试几次。 如果注册失败，则排查DNS节点钱包是否有余额

```
$ cd ~/go/src/github.com/saveio/scan
$ ./scan dns getRegInfo --all
```

```
输出

{
   "WalletAddr": "AYFzNEzNH4KCGw7XS9N9C9oD6VbYobK5f7",
   "IP": "127.0.0.1",
   "Port": "10338",
   "InitDeposit": 10000000000,
   "PeerPubKey": "03ce349d0f6d6913e3a22f5191a7c751cbeedda9ab42ceed217ea8d083cc00dba5"
}
```

4.存储节点配置

> 为了方便测试，创建一个文件夹 node1

创建存储节点文件夹

```
$ cd ~/go/src/github.com/saveio/edge
$ mkdir node1
```

编译

```
$ cd ~/go/src/github.com/saveio/edge
$ make
$ cd node1
$ ln -s ../edge .
```

创建钱包并获取测试币

```
$ cd ~/go/src/github.com/saveio/edge/node1
$ ./edge account add -d
```

```
输出

Index:1
Label:
Address:AHgGsdkCyxFpy3uV5kxQ9ChHZLy41FmSw2
Public key:024617e072438af2e19929ab3079d474499e9991cda3a52daea4b657ac788a98ee
Signature scheme:SHA256withECDSA
Create account successfully.
```

```
$ cd ~/go/src/github.com/saveio/themis
$ go run main.go asset transfer --from=AaK1g8jcH8zRMmN8F8oDaVteMXpit42pjh --to=AHgGsdkCyxFpy3uV
5kxQ9ChHZLy41FmSw2 --amount=100000
```

```
输出

Transfer USDT
  From:AaK1g8jcH8zRMmN8F8oDaVteMXpit42pjh
  To:AHgGsdkCyxFpy3uV5kxQ9ChHZLy41FmSw2
  Amount:100000
  TxHash:1acd3137a1ed9bf16fe9b4aa7766d921e3b7e9422a57542c453e02798a6b0b14

Tip:
  Using './themis info status 1acd3137a1ed9bf16fe9b4aa7766d921e3b7e9422a57542c453e02798a6b0b14'
to query transaction status.
```

等一个区块高度后，查询是否到账

```
$ cd ~/go/src/github.com/saveio/themis
$ go run main.go asset balance AHgGsdkCyxFpy3uV5kxQ9ChHZLy41FmSw2
```

```
输出

BalanceOf:AHgGsdkCyxFpy3uV5kxQ9ChHZLy41FmSw2
  USDT:100000
```

配置文件

```
$ cd ~/go/src/github.com/saveio/edge/node1
$ vim config.json
```

```
{
  "Base": {
    "BaseDir": ".",
    "LogPath": "./Log/",
    "NetworkId": 1567651917,
    "ChainId": "1",
    "BlockTime": 5,
    "PublicIP": "127.0.0.1",
    "IntranetIP": "127.0.0.1",
    "PortBase": 10000,
    "LogLevel": 1,
    "LocalRpcPortOffset": 138,
    "EnableLocalRpc": true,
    "JsonRpcPortOffset": 136,
    "EnableJsonRpc": true,
    "HttpRestPortOffset": 135,
    "HttpCertPath": "",
    "HttpKeyPath": "",
    "RestEnable": true,
    "ChannelProtocol": "tcp",
    "ChannelPortOffset": 501,
    "DBPath": "./DB",
    "ChainRestAddrs": [
      "http://127.0.0.1:20334"
    ],
    "ChainRpcAddrs": [
      "http://127.0.0.1:20336"
    ],
    "NATProxyServerAddrs": "",
    "DspProtocol": "tcp",
    "DspPortOffset": 502,
    "TrackerProtocol": "tcp",
    "TrackerPortOffset": 137,
    "WsPortOffset": 139,
    "ProfilePortOffset": 331,
    "TrackerNetworkId": 1567651917,
    "WalletDir": "./keystore.dat"
  },
  "Fs": {
    "FsRepoRoot": "./FS",
    "FsFileRoot": "./Downloads",
    "FsType": 1,
    "EnableBackup": true,
    "FsGCPeriod": "1h",
    "FsMaxStorage": "1000G"
  },
  "Dsp": {
    "DnsNodeMaxNum": 100,
    "SeedInterval": 600,
    "AutoSetupDNSEnable": false,
    "BlockDelay": "",
    "DnsChannelDeposit": 1000000000,
    "Trackers": [],
    "DNSWalletAddrs": [],
    "ChannelClientType": "rpc",
    "ChannelRevealTimeout": "20",
    "ChannelSettleTimeout": "50",
    "MaxUploadTask": 10000,
    "MaxDownloadTask": 10000,
    "MaxShareTask": 10000
  }
}
```

配置文件说明

配置项	说明	示例
Base.BaseDir	程序根目录	.
LogPath	日志目录相对路径	./Log/
NetworkId	P2P网络ID	1567651917
ChainId	主链ID	1
BlockTime	主链出块时间（秒）	5
PublicIP	固定外部IP	127.0.0.1
IntranetIP	内网IP	127.0.0.1
PortBase	各服务端口基准端口，如下面本地端口为10000+337即10337	10000
LogLevel	日志等级	1
LocalRpcPortOffset	本地RPC端口	138
EnableLocalRpc	是否使用JSON-RPC	true
JsonRpcPortOffset	JSON-RPC 端口	136

配置项	说明	示例
EnableJsonRpc	是否开启RPC	true
HttpRestPortOffset	REST 端口	135
HttpCertPath	HTTPS 证书路径	""
HttpKeyPath	HTTPS key 证书路径	""
RestEnable	是否开启REST	true
ChannelProtocol	Channel通道网络协议	"tcp"
ChannelPortOffset	Channel端口	138
DBPath	DB数据库路径	"./DB"
ChainRestAddrs	主链Rest API地址	["http://127.0.0.1:20334"]
ChainRpcAddrs	主链RPC API地址	["http://127.0.0.1:20336"]
NATProxyServerAddrs	NAT代理地址	""
DspProtocol	DSP网络协议	"tcp"
TrackerProtocol	Tracker网络协议	"tcp"
TrackerPortOffset	Tracker端口	137
WsPortOffset	Websocket端口	139
ProfilePortOffset	性能工具Profile端口	331
TrackerNetworkId	Tracker网络ID	1567651917
WalletDir	钱包地址	"./keystore.dat"
Fs.FsRepoRoot	FS存储路径	"./FS"
FsFileRoot	FS文件下载路径	"./Downloads"
FsType	FS类型 0 客户端 1 服务端 2 兼容	1
EnableBackup	是否开启备份	true
FsGCPeriod	GC时间	"1h"
FsMaxStorage	最大存储容量	"1000G"
Dsp.DnsNodeMaxNum	DNS最大数量	100
SeedInterval	做热点时间间隔	600
AutoSetupDNSEnable	是否开启自动设置DNS	false
BlockDelay	区块落账延迟时间	5
DnsChannelDeposit	DNS通道初始开启充值	1000000000
Trackers	默认Tracker列表	[]
DNSWalletAddrs	默认DNS钱包列表	[]

配置项	说明	示例
ChannelClientType	通道客户端与主链API交互类型	"rpc"
ChannelRevealTimeout	ChannelReveal超时时间	20
ChannelSettleTimeout	ChannelSettlel超时时间	50
MaxUploadTask	最大同时上传数量	10000
MaxDownloadTask	最大同时上传数量	10000
MaxShareTask	最大同时分享文件数量	10000

启动存储节点

```
$ cd ~/go/src/github.com/saveio/edge/node1
$ ./edge --config=config.json -p pwd
```

注册存储节点信息(1TB)

> 另起窗口

```
$ cd ~/go/src/github.com/saveio/edge/node1
$ ./edge node register --nodeAddr=tcp://127.0.0.1:10131 --volume=1073741824 --serviceTime=8760
```

```
输出：

{
  "Tx": "c4d8b3c0c9173ce5801872b3ba417d17b98eade90c2cdb2a10c94189b1429dc7"
}
```

查询是否注册成功

```
$ cd ~/go/src/github.com/saveio/edge/node1
$ ./edge node query --walletAddr=AHgGsdkCyxFpy3uV5kxQ9ChHZLy41FmSw2
```

```
输出:
```

```
{
  "Info": {
    "NodeAddr": "tcp://127.0.0.1:10131",
    "Pledge": 1.073741824,
    "Profit": 0,
    "RestVol": 1048576,
    "ServiceTime": 8760,
    "Volume": 1048576,
    "WalletAddr": "AHgGsdkCyxFpy3uV5kxQ9ChHZLy41FmSw2"
  }
}
```

创建存储节点的扇区(1TB)

```
$ cd ~/go/src/github.com/saveio/edge/node1
$ ./edge sector create --sectorId=1 --proveLevel=1 --sectorSize=1073741824
```

```
输出：

{
  "Tx": "bf2ae6d08bb190b13c028f819e190868752fc686707ffd715440056dfee8d665"
}
```

查询是否注册成功

```
$ cd ~/go/src/github.com/saveio/edge/node1
$ ./edge sector getsector --sectorId=1
```

```
输出:

{
  "NodeAddr": "AHgGsdkCyxFpy3uV5kxQ9ChHZLy41FmSw2",
  "SectorID": 1,
  "Size": 1073741824,
  "ProveLevel": 1,
  "FirstProveHeight": 0,
  "NextProveHeight": 0,
  "TotalBlockNum": 0,
  "FileNum": 0,
  "FileList": null
}
```

与DNS节点建立支付通道

```
$ cd ~/go/src/github.com/saveio/edge/node1
$ ./edge channel open --partnerAddr=AYFzNEzNH4KCGw7XS9N9C9oD6VbYobK5f7 --amount=10000
```

等待几分钟

```
输出

{
  "Id": "101"
}
```

再查询一次

```
$ cd ~/go/src/github.com/saveio/edge/node1
$ ./edge channel list
```

```
输出

{
  "Balance": 10000000000000,
  "BalanceFormat": "10000",
  "Channels": [
    {
      "ChannelId": 101,
      "Balance": 10000000000000,
      "BalanceFormat": "10000",
      "Address": "AYFzNEzNH4KCGw7XS9N9C9oD6VbYobK5f7",
      "HostAddr": "tcp://127.0.0.1:10938",
      "TokenAddr": "AFmseVrdL9f9oyCzZefL9tG6UbvhUMqNMV",
      "IsParticipant1Closer": false,
      "IsParticipant2Closer": false,
      "IsDNS": true,
      "IsOnline": true,
      "Selected": true,
      "CreatedAt": 1606366142
    }
  ]
}
```

5.客户端

启动Seeker

```
$ open seeker
```

设置为本地开发模式

▼ 路径为Mac路径， Windows路径为 %AppData%/Roaming/seek

```
$ cd ~/Library/Application Support/seek
$ vim front-config.json
```

设置 `devEdgeEnable` 为 `true`

```
{"console":false,"devEdgeEnable":true,"maxNumUpload":5,"lang":"en"}
```

创建客户端文件夹

```
$ cd ~/go/src/github.com/saveio/edge
$ mkdir client1
```

链接 `edge`

```
$ cd ~/go/src/github.com/saveio/edge/client1
$ ln -s ../edge .
```

配置文件

```
$ cd ~/go/src/github.com/saveio/edge/client1
$ vim config.json
```

```
{
  "Base": {
    "BaseDir": ".",
    "LogPath": "./Log/",
    "NetworkId": 1567651917,
    "ChainId": "1",
    "BlockTime": 5,
    "PublicIP": "127.0.0.1",
    "IntranetIP": "127.0.0.1",
    "PortBase": 10000,
    "LogLevel": 1,
    "LocalRpcPortOffset": 338,
    "EnableLocalRpc": true,
    "JsonRpcPortOffset": 336,
    "EnableJsonRpc": true,
    "HttpRestPortOffset": 335,
    "HttpCertPath": "",
    "HttpKeyPath": "",
    "RestEnable": true,
    "ChannelProtocol": "tcp",
    "ChannelPortOffset": 301,
    "DBPath": "./DB",
    "PpListenAddr": "",
    "PpBootstrap": null,
    "ChainRestAddrs": [
      "http://127.0.0.1:20334"
    ],
    "ChainRpcAddrs": [
      "http://127.0.0.1:20336"
    ],
    "NATProxyServerAddrs": "",
    "DspProtocol": "tcp",
    "DspPortOffset": 302,
    "TrackerProtocol": "tcp",
    "TrackerPortOffset": 337,
    "WsPortOffset": 339,
    "ProfilePortOffset": 531,
    "TrackerNetworkId": 1567651917,
    "WalletDir": "./keystore.dat"
  },
  "Fs": {
    "FsRepoRoot": "./FS",
    "FsFileRoot": "./Downloads",
    "FsType": 0,
    "EnableBackup": true,
    "FsGCPeriod": "1h",
    "FsMaxStorage": "1000G"
  },
  "Dsp": {
    "DnsNodeMaxNum": 100,
    "SeedInterval": 600,
    "AutoSetupDNSEnable": false,
    "BlockDelay": "",
    "DnsChannelDeposit": 1000000000,
    "Trackers": [],
    "DNSWalletAddrs": [],
    "ChannelClientType": "rpc",
    "ChannelRevealTimeout": "20",
    "ChannelSettleTimeout": "50",
    "MaxUploadTask": 10000,
    "MaxDownloadTask": 10000,
    "MaxShareTask": 10000
  },
  "Bootstraps": []
}
```

创建钱包并获取测试币

> 此操作为命令行操作，也可以使用Seeker操作

```
$ cd ~/go/src/github.com/saveio/edge/client1
$ ./edge account add -d
```

```
输出:

Index:1
Label:
Address:APDvqeWQsPh4m9gFZ7ipmet42pRphLpgBJ
Public key:03806b62e8773fe403aec0d93730b30c15276b8a4e7a0cd858861ec09a383b565c
Signature scheme:SHA256withECDSA
Create account successfully.
```

获取测试币(参考之前的步骤)

启动

```
$ cd ~/go/src/github.com/saveio/edge/client1
$ ./edge --config=config.json -p pwd
```

注册DNS Header

> 说明：注册DNS Header是为了上传文件时，可以生成URL。任意Edge节点都可以注册，仅需注册一次。

```
$ cd ~/go/src/github.com/saveio/edge/client1
$ ./edge dns registerheader --header="oni" --desc="oni" --ttl=100000
```

```
输出：

{
  "Tx": "a4efd633a4342dd0c1796d52e75a8c1a716a41b845f67fbf348d9ab4e11e8cac"
}
```

重启Seeker