

PENROSE部署方案

一、节点部署

创世节点部署

1. 下载源码

```
1 $ git clone https://github.com/EOS-Mainnet/eos.git
2 $ git fetch
3 $ git checkout -b mainnet-1.4.1
4 $ git submodule update --init --recursive
```

2. 编译源码

```
1 $ ./eosio_build.sh
2 $ sudo ./eosio_install.sh
```

注意：自动化编译过程可能会因为主机物理内存小于7G，boost库，mongodb等下载失败或其他原因而编译失败，请查找原因或修改脚本自行解决。下载失败可多次重新尝试解决。

安装成功后查看版本

```
1 $ nodeos -v
2 mainnet-1.4.1
```

3. 创世节点配置修改

在命令行先运行nodeos与keosd，让其产生各自的配置文件。

配置文件路径：~/local/share/eosio/nodeos/config/config.ini

配置后需开放端口9876，8888，4321（按需开放端口）

```
1 # 设置一个自己的bp名字
2 agent-name = "EOS Shen Si"
3 # 创世节点名字必须为eosio
4 producer-name = eosio
5 # 创建一对密钥 创世节点不能修改私钥
6 signature-provider =
  EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV=KEY:5KQwrPbwdL6PhXuj
  xW37FSSQZ1JiwsST4cqQzDeyXtP79zkvFD3
7 # 创世节点才需要，其他bp不需要开启
8 enable-stale-production = true
9
10 # 本机ip
11 p2p-server-address = 192.168.154.157:9876
```

```

12 # 添加一些稳定的其他bp节点
13 # p2p-peer-address =
14
15 #添加插件
16 plugin = eosio::chain_api_plugin
17 plugin = eosio::history_plugin
18 plugin = eosio::history_api_plugin
19 plugin = eosio::producer_plugin

```

4. 启动创世节点

```
1 $ nodeos
```

如果此时是创世节点，此时已开始出块。

5. 导入创世账户eosio的私钥到钱包

创建钱包

```
1 $ cleos wallet create --to-console
```

记得保存返回的钱包解锁密码，丢失无法找回钱包内数据

创建密钥

```
1 $ cleos wallet create_key
```

导入密钥到钱包

```
1 $ cleos wallet import -n default --private-key
5KQwrPbwdL6PhXuJxW37FSSQZ1JiwsST4cqQzDeyXtP79zkvFD3
```

查看钱包密钥

```
1 $ cleos wallet unlock
2 $ cleos wallet private_keys
```

输入钱包解锁密码查看

6. 创建系统账户

```

1 $ cleos create account -f eosio eosio.bpay
EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
2 $ cleos create account -f eosio eosio.msigs
EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
3 $ cleos create account -f eosio eosio.names
EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
4 $ cleos create account -f eosio eosio.ram
EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
5 $ cleos create account -f eosio eosio.ramfee
EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
6 $ cleos create account -f eosio eosio.saving
EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
7 $ cleos create account -f eosio eosio.stake
EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
8 $ cleos create account -f eosio eosio.token
EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV

```

```
9 $ cleos create account -f eosio eosio.vpay
EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV
```

7. 部署系统合约

```
1 $ cleos --wallet-url http://127.0.0.1:8900 --url http://127.0.0.1:8888 set
  contract eosio.token ~/eos/build/contracts/eosio.token/
2 $ cleos --wallet-url http://127.0.0.1:8900 --url http://127.0.0.1:8888 set
  contract eosio.msig ~/eos/build/contracts/eosio.msig/
3 $ cleos --wallet-url http://127.0.0.1:8900 --url http://127.0.0.1:8888
  push action eosio.token create '["eosio", "1000000000.0000 SYS"]' -p
  eosio.token
4 $ cleos --wallet-url http://127.0.0.1:8900 --url http://127.0.0.1:8888
  push action eosio.token issue '["eosio", "1000000000.0000 SYS", "memo"]' -
  p eosio
5 $ cleos --wallet-url http://127.0.0.1:8900 --url http://127.0.0.1:8888 set
  contract eosio ~/eos/build/contracts/eosio.system/
6 $ cleos --wallet-url http://127.0.0.1:8900 --url http://127.0.0.1:8888
  push action eosio setpriv '["eosio.msig", 1]' -p eosio@active
```

同步节点部署

1. 下载源码
2. 编译源码
3. 普通节点配置
4. 创建钱包，导入密钥

BP节点部署

BP节点注册

二、命令行工具

三、RPC调用

EOS官方文档: <https://developers.eos.io/eosio-nodeos/reference>

Chain相关API

get_info

功能：获取最新区块信息

参数：无

```
1 curl -X POST --url http://127.0.0.1:8888/v1/chain/get_info
```

get_block

功能：获取指定块信息

参数: block_num_or_id

```
1 curl -X POST --url http://127.0.0.1:8888/v1/chain/get_block -d  
  '{"block_num_or_id":191}'
```

get_block_header_state

功能: 获取区块头状态

参数: block_num_or_id

```
1 curl -X POST --url http://127.0.0.1:8888/v1/chain/get_block_header_state -  
  d '{"block_num_or_id":191}'`
```

get_account

功能: 获取账户信息

参数: account_name

```
1 curl -X POST --url http://127.0.0.1:8888/v1/chain/get_account -d  
  '{"account_name":"tester"}'
```

get_abi

功能: 获取ABI信息

参数: account_name

```
1 curl -X POST --url http://127.0.0.1:8888/v1/chain/get_abi -d  
  '{"account_name":"eosio.token"}'
```

get_code

功能: 获取代码

参数: account_name, code_as_wasm

```
1 curl -X POST --url http://127.0.0.1:8888/v1/chain/get_code -d  
  '{"code_as_wasm":"true","account_name":"eosio.token"}'
```

get_table_rows

功能: 获取表格行数

参数: scope, code, table, json, lower_bound, upper_bound, limit

```
1 curl --request POST \  
2   --url http://127.0.0.1:8888/v1/chain/get_table_rows \  
3   --data  
  '{"scope":"eosio","code":"eosio.token","table":"test_table","json":"true",  
  "upper_bound":8}'
```

get_currency_balance

功能: 获取账户余额

参数: code, account, symbol

```

1 curl --request POST --url
  'http://127.0.0.1:8888/v1/chain/get_currency_balance' -d
  '{"code":"eosio.token", "account":"user", "symbol":"'
2 EOS"}'

```

abi_json_to_bin

功能：将JSON的智能合约转换成二进制

参数：code, action, args

args参数位JSON格式

```

1 curl --request POST --url http://127.0.0.1:8888/v1/chain/abi_json_to_bin -
  -data '{"code":"eosio.token", "action":"transfer", "args":"'
  {'\from\':"user\","\to\':"tester\","\quantity\':"40 EOS\"}'

```

abi_bin_to_json

功能：将二进制代码转换成JSON格式的智能合约

参数：code, action, binargs

```

1 curl --request POST \
2   --url http://127.0.0.1:8888/v1/chain/abi_bin_to_json \
3   --data
  '{"code":"eosio.token", "action":"issue", "binargs":"'000000000587203..."}'

```

get_required_keys

功能：返回用于给一个交易签名的keys

参数：transaction, available_keys

如：有transaction (JSON格式)

```

1 {
2   "ref_block_num": "100",
3   "ref_block_prefix": "137469861",
4   "expiration": "2017-09-25T06:28:49",
5   "scope": ["initb", "initc"],
6   "actions": [{
7     "code": "currency",
8     "type": "transfer",
9     "recipients": ["initb", "initc"],
10    "authorization": [{
11      "account": "initb",
12      "permission": "active"
13    }],
14    "data": "000000000041934b0000000008041934be803000000000000"
15  }],
16  "signatures": [],
17  "authorizations": []
18 }

```

以及 available_keys (数组格式)

15- push_transactions

功能：发送一组交易

同上

Wallet相关API

create

作用：新建钱包

参数：直接用钱包名

返回：钱包密码

```
1 curl --request POST --url http://127.0.0.1:8888/v1/wallet/create --data  
  "guqianfeng"
```

注意: 通过http_api建立的钱包保存位置, 和~/eosio-wallet/config.ini中设置的目录不一样。http_api钱包保存的位置, 需要在~/local/share/eosio/nodeos/config/config.ini中设置wallet-dir属性。

open

功能：打开钱包

参数：直接用钱包名

返回：如果返回值为空, 则成功打开钱包

```
1 curl --request POST --url http://127.0.0.1:8888/v1/wallet/open --data  
  "guqianfeng"
```

lock

功能：锁定钱包

参数：直接用钱包名

返回：如果返回值为空, 则成功锁定钱包

```
1 curl --request POST --url http://127.0.0.1:8888/v1/wallet/lock --data  
  "guqianfeng"
```

lock_all

功能：锁定所有钱包

参数：无

返回：空值

```
1 curl --request POST --url http://127.0.0.1:8888/v1/wallet/lock_all
```

unlock

功能：解锁钱包

参数：一个json格式的数组，第一项是钱包名，第二项是钱包密码

返回：如果返回值为空，则成功解锁钱包

```
1 curl --request POST --url http://127.0.0.1:8888/v1/wallet/unlock --data  
  '["guqianfeng","PW5HuyuuNftajiTPPvABkXZmRG2AyrFpx3W3cgMmeBmZYXK6Q5KEQ"]'
```

import_key

功能：导入私钥

参数：一个json格式的数组，第一项是钱包名，第二项是私钥（该私钥可以通过其他方式生成，如：cleos create key）

返回：如果返回值为空，则成功导入私钥

```
1 curl --request POST \  
2   --url http://127.0.0.1:8888/v1/wallet/import_key \  
3   --data  
  '["guqianfeng","5J5T3cfraaYYnHTRpKdEwu23SwFLiyNyKAXK4FZnMxvQRE5eqah"]'
```

list_wallets

功能：列出所有钱包

参数：无

返回：钱包数组，如：["guqianfeng *"]，如果有*，则说明已经解锁

```
1 curl --request POST --url http://127.0.0.1:8888/v1/wallet/list_wallets
```

list_keys

功能：暂时无法使用

参数：无

返回：

```
1 curl --request POST --url http://127.0.0.1:8888/v1/wallet/list_keys
```

get_public_keys

功能：列出所有公钥

参数：无

返回：公钥数组

```
1 curl --request POST --url http://127.0.0.1:8888/v1/wallet/get_public_keys
```


set_timeout

功能：设置自动锁定钱包的时间，秒数

参数：秒数

返回：如果返回值为空，则成功设置

```
1 curl --request POST --url http://127.0.0.1:8888/v1/wallet/set_timeout --  
data 1800
```

History 获取历史纪录

get_actions

功能：获取Actions历史

参数：pos, offset, account_name

返回：Actions数组

```
1 curl --request POST \  
2 --url http://127.0.0.1:8888/v1/history/get_actions \  
3 --data '{"pos":1,"offset":20,"account_name":"user"}'
```

get_transaction

功能：返回transaction历史记录

参数：id

返回：该交易的json对象

```
1 curl --request POST \  
2 --url http://127.0.0.1:8888/v1/history/get_transaction \  
3 --data '{"id":1000}'
```

get_key_accounts

功能：返回给定某个公钥下的所有账户

参数：public_key

返回：例如{"account_names":["eosio.hello","eosio.token","tester","user"]}

```
1 curl --request POST \  
2 --url http://127.0.0.1:8888/v1/history/get_key_accounts \  
3 --data  
'{"public_key":"EOS4wM7RcsymSrjTR4S4RixzrRPJpfz1hNrToa85kEW34bSMu7e8Z"}'
```

get_controlled_accounts

功能：获得控制的账户

参数: controlling_account

```
1 curl --request POST \  
2   --url http://127.0.0.1:8888/v1/history/get_controlled_accounts \  
3   --data '{"controlling_account":"guqianfeng"}'
```