

土豆链节点快速配置手册

2018-08-01

本文档主要介绍土豆链节点编译安装和配置。

注意：因为命令和配置参数很多，这里只用实例介绍，详细的命令参数说明请查看nodeos和cleos命令文档。

- 土豆链节点快速配置手册
 - 1. 名词解释
 - 2. 节点代码编译、程序安装
 - 3. 默认配置文件所在目录
 - 4. 创建钱包和公私钥
 - 5. 通用配置
 - 6. 创世节点配置
 - 6.1. 启动参数配置
 - 6.2. 部署系统合约
 - 7. 矿工节点配置
 - 1. 注册矿工账号
 - 2. 启动参数配置
 - 8. 备份节点配置
 - 9. 建议

1. 名词解释

- 创世节点(BIOS)用于创建初始区块，加载基础合约，供其它节点同步连接。
- 矿工节点(BP)主要用于生产同步区块，计算合约产生的费用归矿工账号所有。
- 备份节点 不产生区块，同步区块信息，提供区块信息查询功能。
- 私钥 用来进行签名操作，私钥可以生成唯一一对应公钥。
- 公钥 用来对私钥的签名进行验证。
- 钱包 生成和保存私钥的地方，当需要进行签名操作时，会从钱包读取私钥列表，进行签名。
- 带宽 带宽分为cpu带宽和net带宽两种，每次执行合约都会消耗一定的带宽（从账号中扣取）。
- 合约 一段在链上可执行代码，绑定在账号上，每个账号只能绑定一份合约。
- 账号 存储用户信息，包括余额、带宽、合约（如果有）等。
- 总票数 总票数=总发行的货币数量。
- 投票 投票者抵押货币投票给矿工，当矿工投票数大于总票数的15%时创世节点停止产生区块，转由矿工生产区块。

2. 节点代码编译、程序安装

注意：后面编译依赖GIT，所以务必保证代码根目录的.git完整，以及系统安装有git。

以下以ubuntu系统为例，其它系统步骤类似。

- 先安装GIT，如果已经安装则忽略这一步

```
sudo apt update && sudo apt install git -y
```

- 从GIT上克隆最新代码

```
git clone http://192.168.1.2:8090/Potato-Token/potato.git
```

- 使用pcoin_build.sh自动编译

脚本会检查系统依赖库，并自动安装，请注意管理员权限提示，如果下载依赖库时中断请检查网络状态，然后再次执行。

```
cd potato
./pcoin_build.sh
```

- 安装程序到系统

编译成功会出现'POTATO'的ASCII图样提示，请执行以下操作，把程序安装到系统，如果出现编译错误，请提交错误提示给相关人员查看解决。

```
cd build
sudo make install
```

3. 默认配置文件所在目录

Linux: ~/local/share/pcoin/nodeos/config

Mac: ~/Library/Application Support/pcoin/nodeos/config

4. 创建钱包和公私钥

- 启动钱包服务

```
keosd --unlock-timeout 3600
```

- 创建钱包

请保存输出的钱包密码，下次打开钱包时用的到。

```
cleos wallet create -n default
```

打开钱包

```
cleos wallet open -n default
```

- 创建公私钥

```
cleos create key
```

- 把上步生成的私钥导入钱包

```
cleos wallet import -n default --private-key [private key]
```

5. 通用配置

参数名	示例	说明
agent-name	"potato bios"	p2p网络中标识自己的节点的名字
producer-name	"pcoin"	矿工的账号名
signature-provider	PC6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV =KEY:5KQwrPbwdL6PhXujxW37FSSQZ1JiwsST4cqQzDeyXtP79zkvFD3	矿工的公钥私钥，用于签名
p2p-server-address	"0.0.0.0:9876"	假如要成为BP节点，ip地址需要是公网地址。假如不想成为BP，这个地址可以不填写。
http-server-address	"127.0.0.1:8888"	http服务器地址
enable-stale-production	true	启动后立即开始生产块。如果不是BIOS节点，这里填false。
p2p-peer-address	"192.168.1.2:9876"	其它节点的p2p同步地址，用于同步数据，此参数可以有多个，即连接到多个节点。

6. 创世节点配置

6.1. 启动参数配置

- 创建pcoin系统账号的公私钥，参见[4. 创建钱包和公私钥](#)
- 修改genesis.json的initial_key为上一步生成的公钥
- 启动节点，注意填写第1步生成的公私钥

```
nodeos --agent-name "potato bios" --producer-name "pcoin" \  
--signature-provider [public key]=KEY:[private key] \  
--p2p-server-address "0.0.0.0:9876" --http-server-address "127.0.0.1:8888" --enable-stale-production
```

6.2. 部署系统合约

- 创建系统账号

OwnerKey与ActiveKey相同，是[6.1](#)生成的公钥

```
cleos create account pcoin pcoin.token [OwnerKey] [ActiveKey] -p pcoin  
cleos create account pcoin pcoin.msigs [OwnerKey] [ActiveKey] -p pcoin  
cleos create account pcoin pcoin.bpay [OwnerKey] [ActiveKey] -p pcoin  
cleos create account pcoin pcoin.names [OwnerKey] [ActiveKey] -p pcoin  
cleos create account pcoin pcoin.ram [OwnerKey] [ActiveKey] -p pcoin  
cleos create account pcoin pcoin.ramfee [OwnerKey] [ActiveKey] -p pcoin  
cleos create account pcoin pcoin.saving [OwnerKey] [ActiveKey] -p pcoin  
cleos create account pcoin pcoin.stake [OwnerKey] [ActiveKey] -p pcoin  
cleos create account pcoin pcoin.vpay [OwnerKey] [ActiveKey] -p pcoin
```

- 部署合约

```
cleos set contract pcoin.token build/contracts/pcoin.token -p pcoin.token  
cleos set contract pcoin.msigs build/contracts/pcoin.msigs -p pcoin.msigs
```

- 创建货币

```
cleos push action pcoin.token create '["pcoin", "1000000000.0000 PC", 0, 0, 0]' -p pcoin.token  
cleos push action pcoin.token issue '["pcoin", "1000000000.0000 PC", "issue"]' -p pcoin
```

- 部署系统合约

```
cleos set contract pcoin build/contracts/pcoin.system -p pcoin  
cleos push action pcoin setpriv '["pcoin.msigs", 1]' -p pcoin@active
```

7. 矿工节点配置

1. 注册矿工账号

- 创建矿工账号的公私钥，参见[4. 创建钱包和公私钥](#)
- 创建矿工账号

```
cleos system newaccount --stake-net [quantity] --stake-cpu [quantity] --buy-ram-kbytes 8192 \  
[creator] [name] [OwnerKey] [ActiveKey] -p [creator]  
# 参数说明  
# creator 矿工账号的创建者账号。  
# name 矿工账号的名字。  
# quantity 购买带宽的资源，从创建者账号扣取。  
# OwnerKey与ActiveKey相同，矿工账号的公钥，即上步生成的公钥。  
  
# 示例  
cleos system newaccount --stake-net "50.0000 PC" --stake-cpu "50.0000 PC" --buy-ram-kbytes 888888 pcoin pcbpa \  
PC7n1U9Z2NqeVEvQZyJHcedNXXRVWshmmuGH2j3r6bD4c8fH4U8QL PC7n1U9Z2NqeVEvQZyJHcedNXXRVWshmmuGH2j3r6bD4c8fH4U8QL -p pcoin
```

- 转账给矿工账号

```
cleos transfer [from] [recipient] [amount] [memo]  
# 参数说明  
# from 转账账号  
# recipient 接收转账账号  
# amount 金额  
# memo 注释  
  
# 示例  
cleos transfer pcoin pcbpa "2500000.0000 PC" "trans to pcbpa"
```

- 注册成矿工

```
cleos system regproducer [name] [ActiveKey] [url]
# 参数说明
# name 矿工账号
# ActiveKey 矿工账号的公钥
# url 矿工的网站

#示例
cleos system regproducer pcbpa PC7n1U9Z2NqeVEvQZYjHCedNXRVWshmmuGH2j3r6bD4c8fH4U8QL https://202.46.36.57:8888
```

5. 抵押相应的资源

```
cleos system delegatebw [name] [name] [stake_net_quantity] [stake_cpu_quantity] --transfer
# 参数说明
# name 矿工账号
# stake_net_quantity 抵押的网络带宽资源
# stake_cpu_quantity 抵押的CPU带宽资源

# 示例
cleos system delegatebw pcbpa pcbpa '12000000.0000 PC' '12000000.0000 PC' --transfer
```

6. 投票给自己

```
cleos system voteproducer prods [voter] [producers] -p [voter]
# 参数说明
# voter 矿工账号
# producers 抵押的网络带宽资源

#示例
cleos system voteproducer prods pcbpa pcbpa
```

2. 启动参数配置

1. 创建pcoin系统账号的公私钥，参见[创建钱包和公私钥](#)
 2. 修改genesis.json的initial_key为创世节点(BIOS)生成的公钥
 3. 启动节点，注意填写第1步生成的公私钥
- 假设已知bios节点202.46.36.56:9876，BP节点202.46.36.58:9876。

```
nodeos --agent-name "potato pb a" --producer-name "pba" \
--signature-provider [public key]=KEY:[private key] \
--p2p-server-address "0.0.0.0:9876" --http-server-address "127.0.0.1:8888" \
--p2p-peer-address "202.46.36.56:9876" --p2p-peer-address "202.46.36.58:9876"
```

8. 备份节点配置

备份节点配置与矿工节点配置类似，只是不注册成矿工，备份节点常用来数据查询。建议开启bnet_plugin和mongo_db_plugin插件。bnet_plugin是一个用来快速同步区块数据的插件，mongo_db_plugin是保存交易信息到mongodb的插件。

假设已知bios节点202.46.36.56:9876，BP节点202.46.36.57:9876、202.46.36.58:9876，并且这三个节点都开启了bnet_plugin插件。

```
nodeos --agent-name "potato pb a" --producer-name "pba" \
--signature-provider [public key]=KEY:[private key] \
--p2p-server-address "0.0.0.0:9876" --http-server-address "127.0.0.1:8888" \
--p2p-peer-address "202.46.36.56:9876" --p2p-peer-address "202.46.36.57:9876" --p2p-peer-address "202.46.36.58:9876" \
--plugin pcoin::mongo_db_plugin --mongodb-uri mongodb://127.0.0.1:27017/POTATO \
--plugin pcoin::bnet_plugin --bnet-endpoint "0.0.0.0:4321" --bnet-threads 4 \
--bnet-connect "202.46.36.56:4321" --bnet-connect "202.46.36.56:4321" --bnet-connect "202.46.36.56:4321"
```

9. 建议

1. 从安全角度来说，不要私钥泄露给他人，知道账户私钥后，可以随意操纵账户。
2. 如果使用助记词生成的私钥，请使用复杂度较高的助记词。
3. 上面传入很多参数都是演示用的，如果正式部署，建议使用配置文件的方式。
4. 建议使用SIGINT信号(CTRL+C)来关闭节点，防止程序产生脏数据。
5. 如果有脏数据产生，请使用--replay-blockchain参数或者--delete-all-blocks参数启动节点。