以太坊学习：

目录

用geth搭建以太坊私链

因为公共网络的区块数量太多，同步耗时太长，我们为了方便快速了解 Geth，可以试着用它来搭一个 只属于自己的私链。

1.编写创世区块

首先，我们需要创建网络的"创世"（genesis）状态，这写在一个小小的 JSON 文件里（例如，我们 将其命名为 genesis.json）：新建一个文件 /usr/

```
 1  {
 2      "config": {
 3          "chainId": 15
 4      },
 5      "difficulty": "2000",
 6      "gasLimit": "2100000",
 7      "alloc": {
 8          "7df9a875a174b3bc565e6424a0050ebc1b2d1d82": { "balance": "300000" },
 9          "f41c74c9ae680c1aa78f42e5647a62f353b7bdde": { "balance": "400000" }
10      }
11  }
12  //config：区块的信息
13  //chainId：你这个测试网络的Id，不要和主网等其他测试网络相同
14  //difficulty：挖矿难度
15  //gasLimit：一个块最多含有多少gas的上限
16  //alloc：分配：在创世时，就给该地址多少代币
```


0x8cbf968154223254306C41C03F0110FC97B200BA：是我的MetaMask地址

2.初始化创世区块

```
[root@Q1nChanCentos GethProjects]# geth --datadir . init genesis.json
INFO [07-12|00:35:53.916] Maximum peer count                      ETH=50 LES=0 total=50
INFO [07-12|00:35:53.916] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm:
 directory"
INFO [07-12|00:35:53.918] Allocated cache and file handles         database=/usr/local/GethProjects/g
cache=16.00MiB handles=16
INFO [07-12|00:35:53.922] Persisted trie from memory database      nodes=1 size=147.00B time=9.378µs
ze=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [07-12|00:35:53.922] Successfully wrote genesis state         database=chaindata
hash=8d464d…6f7d43
INFO [07-12|00:35:53.922] Allocated cache and file handles         database=/usr/local/GethProjects/g
data cache=16.00MiB handles=16
INFO [07-12|00:35:53.925] Persisted trie from memory database      nodes=1 size=147.00B time=4.418µs
ze=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [07-12|00:35:53.925] Successfully wrote genesis state         database=lightchaindata
hash=8d464d…6f7d43
[root@Q1nChanCentos GethProjects]#
```

```
[root@Q1nChanCentos GethProjects]# ll
总用量 12
-rw-r--r--. 1 root root  180 7月  12 00:31 genesis.json
drwxr-xr-x. 4 root root 4096 7月  12 00:34 geth
drwx------. 2 root root 4096 7月  12 00:34 keystore
[root@Q1nChanCentos GethProjects]# cd geth
[root@Q1nChanCentos geth]# ll
总用量 8
drwxr-xr-x. 2 root root 4096 7月  12 00:35 chaindata
drwxr-xr-x. 2 root root 4096 7月  12 00:35 lightchaindata
[root@Q1nChanCentos geth]# cd lightchaindata/
[root@Q1nChanCentos lightchaindata]# ll
总用量 24
-rw-r--r--. 1 root root  706 7月  12 00:35 000002.ldb
-rw-r--r--. 1 root root  166 7月  12 00:35 000003.log
-rw-r--r--. 1 root root   16 7月  12 00:35 CURRENT
-rw-r--r--. 1 root root   16 7月  12 00:35 CURRENT.bak
-rw-r--r--. 1 root root    0 7月  12 00:34 LOCK
-rw-r--r--. 1 root root 1105 7月  12 00:35 LOG
-rw-r--r--. 1 root root  239 7月  12 00:35 MANIFEST-000004
```

GethProjects目录下的geth是初始化之后生成的

3.运行私链

```
[root@Q1nChanCentos GethProjects]# geth --datadir . --networkid 15
WARN [07-12|00:41:35.006] Sanitizing cache to Go's GC limits        provided=1024 updated=660
INFO [07-12|00:41:35.007] Maximum peer count                         ETH=50 LES=0 total=50
INFO [07-12|00:41:35.007] Smartcard socket not found, disabling      err="stat /run/pcscd/pcscd.comm:
directory"
INFO [07-12|00:41:35.009] Starting peer-to-peer node                 instance=Geth/v1.9.8-stable-d62e9
/go1.13.4
INFO [07-12|00:41:35.009] Allocated trie memory caches               clean=165.00MiB dirty=165.00MiB
INFO [07-12|00:41:35.009] Allocated cache and file handles           database=/usr/local/GethProjects/
cache=330.00MiB handles=2048
INFO [07-12|00:41:35.016] Opened ancient database                    database=/usr/local/GethProjects/
ancient
INFO [07-12|00:41:35.016] Initialised chain configuration            config="{ChainID: 15 Homestead: <
 DAOSupport: false EIP150: <nil> EIP155: <nil> EIP158: <nil> Byzantium: <nil> Constantinople: <nil>
l> Istanbul: <nil> Engine: unknown}"
INFO [07-12|00:41:35.016] Disk storage enabled for ethash caches     dir=/usr/local/GethProjects/geth/
INFO [07-12|00:41:35.016] Disk storage enabled for ethash DAGs       dir=/root/.ethash
INFO [07-12|00:41:35.016] Initialising Ethereum protocol             versions="[64 63]" network=15 dbv
WARN [07-12|00:41:35.016] Upgrade blockchain database version        from=<nil> to=7
INFO [07-12|00:41:35.017] Loaded most recent local header            number=0 hash=8d464d…6f7d43 td=20
INFO [07-12|00:41:35.017] Loaded most recent local full block        number=0 hash=8d464d…6f7d43 td=20
INFO [07-12|00:41:35.017] Loaded most recent local fast block        number=0 hash=8d464d…6f7d43 td=20
INFO [07-12|00:41:35.017] Regenerated local transaction journal      transactions=0 accounts=0
INFO [07-12|00:41:35.020] Allocated fast sync bloom                  size=330.00MiB
INFO [07-12|00:41:35.022] Initialized fast sync bloom                items=1 errorrate=0.000 elapsed=1
INFO [07-12|00:41:35.026] New local node record                      seq=1 id=f95d151b9d9fe5ba ip=127.
tcp=30303
INFO [07-12|00:41:35.026] Started P2P networking                     self=enode://24830741ffa6b57ffab7
c16d134b6bcca9bf9a3ee5fc92a00116d867a810a3b13c5da4b9517680ae0eaee054091e369178f2b0eede71c1619@127.0.
INFO [07-12|00:41:35.027] IPC endpoint opened                        url=/usr/local/GethProjects/geth.
INFO [07-12|00:41:39.359] New local node record                      seq=2 id=f95d151b9d9fe5ba ip=120.
```

geth控制台操作

- geth --datadir . --networkid 15 console：进入控制台

- web3：调出web3对象

```
Welcome to the Geth JavaScript console!

instance: Geth/v1.9.8-stable-d62e9b28/linux-amd64/go1.13.4
at block: 0 (Thu, 01 Jan 1970 08:00:00 CST)
 datadir: /usr/local/GethProjects
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpoo

> INFO [07-12|13:18:22.783] New local node record                    seq=4 id=f95d151b9d9fe5ba
37556 tcp=30303
> web3
{
  admin: {
    datadir: "/usr/local/GethProjects",
    nodeInfo: {
      enode: "enode://24830741ffa6b57ffab7b28382c6ad35da1c16d134b6bcca9bf9a3ee5fc92a00116d867
ae0eaee054091e369178f2b0eede71c1619@120.84.11.13:30303?discport=37556",
      enr: "enr:-J-4QGIqfGSyBN3rqLgyytfjKjMiDO5UVhAeQCHiX7HZHFZrVUAaHw4-LR4g1vezKgnOHKGkgElBT
GhHENWW6AgmlkgnY0gmlwhHhUCw2Jc2VjcDI1NmsxoQMkgwdB_6a1f_q3soOCxq012hwW0TS2vMqb-aPuX8kqAINOY3CC
8",
      id: "f95d151b9d9fe5bad619d0ae5b74f1ec96be62f5ead26e7bf22810e7b3d05454",
      ip: "120.84.11.13",
      listenAddr: "[::]:30303",
      name: "Geth/v1.9.8-stable-d62e9b28/linux-amd64/go1.13.4",
      ports: {
        discovery: 37556,
        listener: 30303
      },
      protocols: {
        eth: {...}
```

- admin：和权限相关的东西

```
1  admin
2  {
3    datadir: "/usr/local/GethProjects",
4    nodeInfo: {
5      //nodeInfo：我们启动一个节点的完整的信息
6      enode: "enode://248
7      enr: "enr:-J-4QGIqf
```

```
 8    id: "f95d151b9d9fe5bad619d0ae5b74f1ec96be62f5ead26e7bf22810e7b3d05454", 9 │    ip: "120.84.11.13",
10    listenAddr: "[::]:30303",
11    //listenAddr：监听的地址
12    name: "Geth/v1.9.8-stable-d62e9b28/linux-amd64/go1.13.4",
13    ports: {
14      discovery: 37556,
15      listener: 30303
16    },
17    protocols: {
18      eth: {
19        config: {...},
20        difficulty: 2000,
21        genesis: "0x8d464df7c77e9e659073ca4a6dbb7bce4814e754e0b43370806f519c036f7d43",
22        head: "0x8d464df7c77e9e659073ca4a6dbb7bce4814e754e0b43370806f519c036f7d43",
23        network: 15
24      }
25    }
26    },
27    peers: [],
28    addPeer: function(),
29    addTrustedPeer: function(),
30    clearHistory: function(),
31    exportChain: function(),
32    getDatadir: function(callback),
33    getNodeInfo: function(callback),
34    getPeers: function(callback),
35    importChain: function(),
36    removePeer: function(),
37    removeTrustedPeer: function(),
38    sleep: function github.com/ethereum/go-ethereum/console.(*bridge).Sleep-fm(),
39    sleepBlocks: function github.com/ethereum/go-ethereum/console.(*bridge).SleepBlocks-fm(),
40    startRPC: function(),
41    startWS: function(),
42    stopRPC: function(),
43    stopWS: function()
44 }
```

- personal属性：控制用户账户

  -
```
 1 │ {
 2    listAccounts: [],
 3    listWallets: [],
 4    deriveAccount: function(),
 5    ecRecover: function(),
 6    getListAccounts: function(callback),
 7    getListWallets: function(callback),
 8    importRawKey: function(),
 9    initializeWallet: function(),
10    lockAccount: function(),
11    newAccount: function github.com/ethereum/go-ethereum/console.(*bridge).NewAccount-fm(),
12    //newAccount：用于创建账户
13    openWallet: function github.com/ethereum/go-ethereum/console.(*bridge).OpenWallet-fm(),
14    sendTransaction: function(),
15    sign: function github.com/ethereum/go-ethereum/console.(*bridge).Sign-fm(),
16    signTransaction: function(),
17    unlockAccount: function github.com/ethereum/go-ethereum/console.(*bridge).UnlockAccount-fm(),
18    unpair: function()
19 }
```

- eth

  -
```
 1 │ WARN [07-12|13:30:50.950] Served eth_coinbase                    reqid=69 t=25.278µs err="etherbase must
 2 │ {
 3    accounts: [],
 4    blockNumber: 0,
 5 │ WARN [07-12|13:30:50.953] Served eth_coinbase                    reqid=80 t=7.173µs  err="etherbase must
 6    coinbase: undefined,
 7    compile: {
```

LEVI_104   👍 0   👎   💬 0   ⭐ 0

```
 8        lll: function(), 9    |    serpent: function(),
10      solidity: function()
11    },
12    defaultAccount: undefined,
13    defaultBlock: "latest",
14    gasPrice: 1000000000,//我们设定的gasLimited
15    hashrate: 0,
16    mining: false,
17    pendingTransactions: [],
18    protocolVersion: "0x40",
19    syncing: false,
20    call: function(),
21    chainId: function(),
22    contract: function(abi),
23    estimateGas: function(),
24    fillTransaction: function(),
25    filter: function(options, callback, filterCreationErrorCallback),
26    getAccounts: function(callback),
27    getBalance: function(),
28    getBlock: function(),
29    getBlockByHash: function(),
30    getBlockByNumber: function(),
31    getBlockNumber: function(callback),
32    getBlockTransactionCount: function(),
33    getBlockUncleCount: function(),
34    getCode: function(),
35    getCoinbase: function(callback),
36    getCompilers: function(),
37    getGasPrice: function(callback),
38    getHashrate: function(callback),
39    getHeaderByHash: function(),
40    getHeaderByNumber: function(),
41    getMining: function(callback),
42    getPendingTransactions: function(callback),
43    getProof: function(),
44    getProtocolVersion: function(callback),
45    getRawTransaction: function(),
46    getRawTransactionFromBlock: function(),
47    getStorageAt: function(),
48    getSyncing: function(callback),
49    getTransaction: function(),
50    getTransactionCount: function(),
51    getTransactionFromBlock: function(),
52    getTransactionReceipt: function(),
53    getUncle: function(),
54    getWork: function(),
55    iban: function(iban),
56    icapNamereg: function(),
57    isSyncing: function(callback),
58    namereg: function(),
59    resend: function(),
60    sendIBANTransaction: function(),
61    sendRawTransaction: function(),
62    sendTransaction: function(),
63    sign: function(),
64    signTransaction: function(),
65    submitTransaction: function(),
66    submitWork: function()
67  }
```

- providers: 简单理解为服务提供商

- txpool: 交易池(目前什么都没)

  -
    ```
    1  {
    2    content: {
    3      pending: {},
    4      queued: {}
    5    },
    6    inspect: {
    ```

LEVI_104   👍 0   👎   💬 0   ⭐ 0

```
 7 |     pending: {}, 8 |       queued: {}
 9 |   },
10 |   status: {
11 |     pending: 0,
12 |     queued: 0
13 |   },
14 |   getContent: function(callback),
15 |   getInspect: function(callback),
16 |   getStatus: function(callback)
17 | }
```

- miner

```
 1 | {
 2 |   getHashrate: function(),
 3 |   setEtherbase: function(),
 4 |   setExtra: function(),//写进区块的附加信息
 5 |   setGasPrice: function(),
 6 |   setRecommitInterval: function(),
 7 |     //start，stop：开始、停止挖矿的选项
 8 |   start: function(),
 9 |   stop: function()
10 | }
```

Geth Console 是一个交互式的 JavaScript 执行环境，里面内置了一些用来操作以太坊的 JavaScript 对象，我们可以直接调用这些对象来获取区块

eth：主要包含对区块链进行访问和交互相关的方法；

net：主要包含查看 p2p 网络状态的方法；

admin：主要包含与管理节点相关的方法；

miner：主要包含挖矿相关的一些方法；

personal：包含账户管理的方法；

txpool：包含查看交易内存池的方法；

web3：包含以上所有对象，还包含一些通用方法。

常用命令有： personal.newAccount()：创建账户；

personal.unlockAccount()：解锁账户；

eth.accounts：列出系统中的账户；

eth.getBalance()：查看账户余额，返回值的单位是 Wei；

eth.blockNumber：列出当前区块高度；

eth.getTransaction()：获取交易信息；

eth.getBlock()：获取区块信息；

miner.start()：开始挖矿；

miner.stop()：停止挖矿；

web3.fromWei()：Wei 换算成以太币；

web3.toWei()：以太币换算成 Wei；

txpool.status：交易池中的状态；

(1)

eth.accounts没有东西：因为只有在geth下创建的用户才存在keystore文件夹里面，而创世区块的信息是不在里面的

```
> eth.accounts
[]
> eth.getBalance("0x8cbf968154223254306C41C03F0110FC97B200BA")
3000000000000000
> web3.fromWei(eth.getBalance("0x8cbf968154223254306C41C03F0110FC97B200BA"
... )
0.003
> web3.from(eth.getBalance("0x8cbf968154223254306C41C03F0110FC97B200BA"),'
TypeError: 'from' is not a function
    at <anonymous>:1:1

> web3.fromWei(eth.getBalance("0x8cbf968154223254306C41C03F0110FC97B200BA"
0.003
>
```

(2) 创建区块

```
> personal.newAccount()
Password:
Repeat password:
INFO [07-12|13:52:40.987] Your new key was generated          address=0xA95DeF0F6890F2c79Bb2f6c
17
WARN [07-12|13:52:40.987] Please backup your key file!         path=/usr/local/GethProjects/keys
07-12T05-52-39.285998661Z--a95def0f6890f2c79bb2f6ca006c18ef0ad56417
WARN [07-12|13:52:40.987] Please remember your password!
"0xa95def0f6890f2c79bb2f6ca006c18ef0ad56417"
> eth.accounts
["0xa95def0f6890f2c79bb2f6ca006c18ef0ad56417"]
```

```
[root@Q1nChanCentos GethProjects]# ll
总用量 16
-rw-r--r--. 1 root root  180 7月  12 00:31 genesis.json
drwxr-xr-x. 5 root root 4096 7月  12 13:18 geth
-rw-------. 1 root root  385 7月  12 13:54 history
drwx------. 2 root root 4096 7月  12 13:52 keystore
[root@Q1nChanCentos GethProjects]# cd keystore/
[root@Q1nChanCentos keystore]# ll
总用量 4
-rw-------. 1 root root 491 7月  12 13:52 UTC--2022-07-12T05-52-39
6417
[root@Q1nChanCentos keystore]# cat UTC--2022-07-12T05-52-39.285998
{"address":"a95def0f6890f2c79bb2f6ca006c18ef0ad56417","crypto":{"c
532df4fd45a54d2fd60c09cecb8c540114a4884718743544a1","cipherparams"
"scrypt","kdfparams":{"dklen":32,"n":262144,"p":1,"r":8,"salt":"b9
c3bfd2f94bbfc"},"mac":"e8644f545528183f7fefeb3e54c6e3b127cb12ff015
8a78-c47c27ffc191","version":3}[root@Q1nChanCentos keystore]#
```

crypto：cipher是加密方式，ciphertext是加密内容

文章知识点与官方知识档案匹配，可进一步学习相关知识

CS入门技能树　Linux入门　初识Linux　7348 人正在系统学习中

"相关推

😠 非常没帮助　　😕 没帮助

LEVI_104　　　　　👍 0　👎　💬 0　⭐ 0