**Geth 部署**

**1、Go下载安装**

|  |
| --- |
| wget https://dl.google.com/go/go1.15.4.linux-amd64.tar.gz |

|  |
| --- |
| tar -C /usr/local -xzf go1.15.4.linux-amd64.tar.gz #解压到指定目录  vim /etc/profile  添加  export PATH=$PATH:/usr/local/go/bin  source /etc/profile  go version #查看版本 |

**2 、go-ethereum下载安装**

下载<https://github.com/ethereum/go-ethereum/releases> 再tags页签下载版本包

**<https://geth.ethereum.org/downloads/> (版本最好是最新的，才能同步完整,此链接下载的直接可以用不用编译，不用安装go)**

|  |
| --- |
| #cd /usr/local/src  #wget <https://github.com/ethereum/go-ethereum/archive/refs/tags/v1.10.11.tar.gz>  #tar xf v1.10.11.tar.gz  #mv go-ethereum-1.10.11 /usr/local/  #cd /usr/local/  #ln -sfv go-ethereum-1.10.11 go-ethereum  #cd go-ethereum  #make geth |

|  |
| --- |
| #vim /etc/profile  添加  export PATH=$PATH:/usr/local/go-ethereum/build/bin  #source /etc/profile  #geth version |

|  |
| --- |
| **另外一种安装方式 只适合eth经典 不适用eth 避免踩坑**  Core-Geth 是适用于多种以太网络的 Go 语言客户端。用户可以使用 Core-geth 运行以太坊经典，以太坊或相关的 testnet 节点。大多数用户想使用 EVM 网络提供的服务就需要安装自己的节点。我们将在这篇文章中帮助您安装 Core-Geth ****使用二进制发行版进行安装**** 二进制发行版发布在 [https://github.com/etclabscore/core-geth/releases](https://github.com/etclabscore/core-geth/releases" \t "https://www.chainnews.com/articles/_blank)。  wget <https://github.com/etclabscore/core-geth/releases/download/v1.11.22/core-geth-linux-v1.11.22.zip>  unzip core-geth-linux-v1.11.22.zip -d /bin/  geth --help # check install, view usage & commands  提示：运行 Core-geth 时，请使用--classic 指定以太坊经典主网标识。 |

**3创建私链**

|  |
| --- |
| mkdir /home/privatechain/data0  cd /home/privatechain/ |

vim genesis.json #新建创世区块描述文件genesis.json

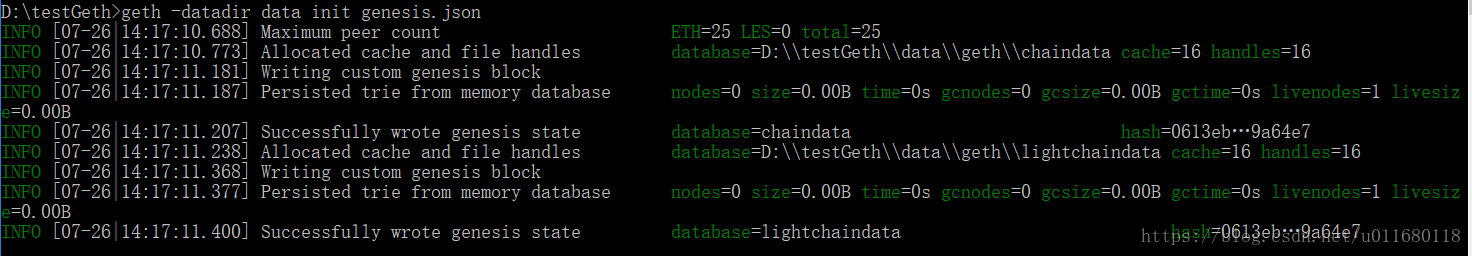
genesis.json是创世区块配置文件，通过此配置让我们很容易挖矿以方便测试

|  |
| --- |
| {  "config": {  "chainId": 666,  "homesteadBlock": 0,  "eip150Block": 0,  "eip150Hash": "0x0000000000000000000000000000000000000000000000000000000000000000",  "eip155Block": 0,  "eip158Block": 0,  "byzantiumBlock": 0,  "constantinopleBlock": 0,  "petersburgBlock": 0,  "istanbulBlock": 0,  "ethash": {}  },  "nonce": "0x0",  "timestamp": "0x5ddf8f3e",  "extraData": "0x0000000000000000000000000000000000000000000000000000000000000000",  "gasLimit": "0x47b760",  "difficulty": "0x00002",  "mixHash": "0x0000000000000000000000000000000000000000000000000000000000000000",  "coinbase": "0x0000000000000000000000000000000000000000",  "alloc": { },  "number": "0x0",  "gasUsed": "0x0",  "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000"  } |

初始化

|  |
| --- |
| geth --datadir data0 init genesis.json |

此时在testGeth目录下会生成data目录，data目录又包含geth和keystore目录，geth目录存储区块数据，keystore目录则保存账户信息。命令执行结果如下：



**启动节点连接私有测试网络**

|  |
| --- |
| geth --syncmode fast --networkid 666 --nodiscover --datadir "/home/privatechain/data0" --rpc --rpcapi "db,eth,net,web3,personal,admin,miner" --rpcaddr 0.0.0.0 --rpccorsdomain "\*" --port 10070 --ws --wsaddr "0.0.0.0" --wsapi "db,eth,net,web3,personal" --allow-insecure-unlock console 2>>geth.log  console 2>> 后面不能有空格 否则会启动不了 |

--nodiscover 关闭p2p网络的自动发现，需要手动添加节点，这样有利于我们隐藏私有网络

--datadir 区块链数据存储目录

--networkid 网络标识，私有链取一个大于4的随意的值

--rpc 启用ipc服务，默认端口号8545

--rpcapi 表示可以通过ipc调用的对象 --rpcaddr ipc监听地址，默认为127.0.0.1，只能本地访问

console 打开一个可交互的javascript环境

identity 区块链的标示，用于标示目前网络的名字

init 指定创世块文件的位置，并创建创世块

datadir 当前区块数据存放的位置

**port 网络监听端口，默认30303**

rpc 启动rpc通信

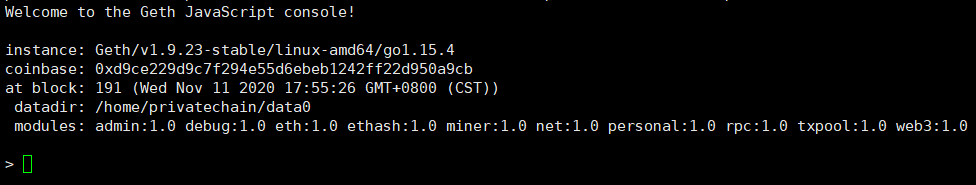
rpcapi 设置允许连接rpc的客户端，一般为db,eth,net,web3

rpccorsdomain 指定什么url能连接到你的节点执行rpc定制端任务，如果输入的是“\*”，则任何url都可以连接到你的rpc实例。

rpcaddr 连接rpc的地址，默认为localhost rpcport 连接rpc的端口，默认为8545

networkid 设置当前区块链的网络ID，用于区分不同的网络，是一个数字 console 启动命令行模式，可以在Geth中执行命令

连接测试网进入控制台，此时我们已经进入geth测试网的交互式控制台，窗口也显示「Welcome to the Geth JavaScript console」成功提示！



这是一个交互式的Javascript执行环境，在这里面可以执行Javascript代码，其中>是命令提示符。在这个环境里也内置了一些用来操作以太坊的Javascript对象，可以直接使用这些对象。这些对象主要包括：

|  |
| --- |
| eth：包含一些跟操作区块链相关的方法  net：包含以下查看p2p网络状态的方法  admin：包含一些与管理节点相关的方法  miner：包含启动&停止挖矿的一些方法  personal：主要包含一些管理账户的方法  txpool：包含一些查看交易内存池的方法  web3：包含了以上对象，还包含一些单位换算的方法  其中常用命令有：  personal.newAccount()：创建账户；  personal.unlockAccount()：解锁账户；  eth.accounts：枚举系统中的账户；  eth.getBalance()：查看账户余额，返回值的单位是 Wei（Wei - 是以太坊中最小货币面额单位，类似比特币中的聪，1 ether = 10^18 Wei）；  eth.blockNumber：列出区块总数；  eth.getTransaction()：获取交易；  eth.getBlock()：获取区块；  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；  admin.addPeer()：连接到其他节点； |

****配置参数含义****

<https://www.cnblogs.com/wanghui-garcia/p/10256520.html>

**常用命令**

****1 创建账号****

创建一个新账号，密码为123456，账号的地址为：0x9cac40f650e2cbe459dcb32c7c23103497134467。

> personal.newAccount("123456")

"0x9cac40f650e2cbe459dcb32c7c23103497134467"

查看所有账号

> eth.accounts

["0x9cac40f650e2cbe459dcb32c7c23103497134467", "0x762a2e28c5dbab9cd31369db5f3cbb48f421c0e3"]

### **2、查看账户余额**

|  |  |
| --- | --- |
| 1  2  3  4 | > eth.getBalance("0x675fda0eac2af486c0b8778ce386150af5140458")  0  > eth.getBalance(eth.accounts[1])  0 |

目前两个账户的以太币余额都是0，要使账户有余额，可以从其他账户转账过来，或者通过挖矿来获得以太币奖励。

### **3、启动&停止挖矿**

* 通过miner.start()来启动挖矿
* 通过miner.stop()来停止挖矿

|  |
| --- |
| 使用JavaScript方法打印所有的余额：  function checkAllBalances() {  var i =0;  eth.accounts.forEach( function(e){  console.log(" eth.accounts["+i+"]: " + e + " \tbalance: " + web3.fromWei(eth.getBalance(e), "ether") + " ether");  i++;  })  };  可以这样调用：  > checkAllBalances();  eth.accounts[0]: 0xd1ade25ccd3d550a7eb532ac759cac7be09c2719 balance: 63.11848 ether  eth.accounts[1]: 0xda65665fc30803cb1fb7e6d86691e20b1826dee0 balance: 0 ether  eth.accounts[2]: 0xe470b1a7d2c9c5c6f03bbaa8fa20db6d404a0c32 balance: 1 ether  eth.accounts[3]: 0xf4dd5c3794f1fd0cdc0327a83aa472609c806e99 balance: 6 ether  由于这个功能在geth重启后会消失，把常用的方法记录下来以便以后调用。loadScript使得这样做很容易。  首先把checkAllBalances()保存至电脑上的一个文件中。例如： /tmp/gethload.js。接着从交互式控制台上导入文件： > loadScript("/tmp/gethload.js") true |

参考文档

<https://www.cnblogs.com/WPF0414/p/10046481.html>

--datadir "xxxx" 指定数据目录，用来存放区块链数据，状态数据，keystore数据等。如果不加这个参数这些数据在不同的系统会放到不同的位置。这个目录占用空间是比较大的，我一般会指定一个目录，并保证这个目录所在分区有足够的磁盘空间。

--cache value 分配给内部缓存的内存MB数量，默认为 128，最后设置大一点儿，起码 1024吧，这个值设大一些可以提高数据同步效率。

--rpc 启用HTTP-RPC服务器

--rpcaddr value HTTP-RPC服务器接口地址(默认值:“localhost”)，默认只允许本地连接，设置为 0.0.0.0 可以接收任何地址发来的连接请求

--rpcport value HTTP-RPC服务器监听端口(默认值:8545)，可以改为不同的端口

--ws 启用WS-RPC服务器，几乎所有第三方节点都不启动这个服务，而要监听以太坊事件又必须启动这个服务

--wsaddr value WS-RPC服务器监听接口地址(默认值:“localhost”)

--wsport value WS-RPC服务器监听端口(默认值:8546)

[](https://metamask.io/index.html)

Passwd:mytestmetamask

avocado feed grain bomb wet glide crisp basket extra broccoli half pool

**问题**

err="only replay-protected (EIP-155) transactions allowed over RPC"

https://blog.csdn.net/cljdsc/article/details/115255048?utm\_medium=distribute.pc\_aggpage\_search\_result.none-task-blog-2~aggregatepage~first\_rank\_v2~rank\_aggregation-1-115255048.pc\_agg\_rank\_aggregation&utm\_term=geth+%E5%8F%91%E9%80%81%E4%BA%A4%E6%98%93&spm=1000.2123.3001.4430