部署合约并调用智能合约的函数

之前的教程中以及创建了一个私链。我们接下来进入控制台部署合约并调用它的函数。

1. 首先进入之前创建的私链的控制台中

* sudo geth --datadir "./" --nodiscover --networkid 528 --ipcdisable --port 3333 --rpc --rpcport 4444 --rpcapi="db,eth,net,web3,personal" --allow-insecure-unlock --rpccorsdomain "\*" console



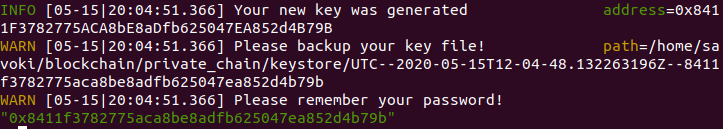
1. 创建我们自己的用户

//要用这个用户来部署合约和调用函数

* personal.newAccount("123456")

//括号内是设的密码，返回“0x......”为创建出来的用户的地址。之后要用到。

Password："0x8411f3782775aca8be8adfb625047ea852d4b79b"（账户地址）



* eth.accounts

//查询用户

* eth.getBalance(eth.accounts[0])

// 通过eth.accounts[0]查询账户余额。若私链上有多个用户，第一个用户为eth.accounts[0]，第二个为eth.accounts[1]。以此类推。

* eth.getBalance("0x......账户地址")

// 通过地址查询出账户地址

1. 挖矿赚钱

我们创建的账户没有钱，想要部署合约和调用函数就得有钱。所以我们挖矿来赚钱。

挖矿所得的奖励会进入矿工的账户。这个账户叫做 coinbase，默认情况下 coinbase 是本地账户中的第一个账户，即eth.accounts[0]。

* eth.coinbase

// 第一种查看矿工的账户地址的方式，该地址默认为第一个账户地址

* web3.eth.coinbase

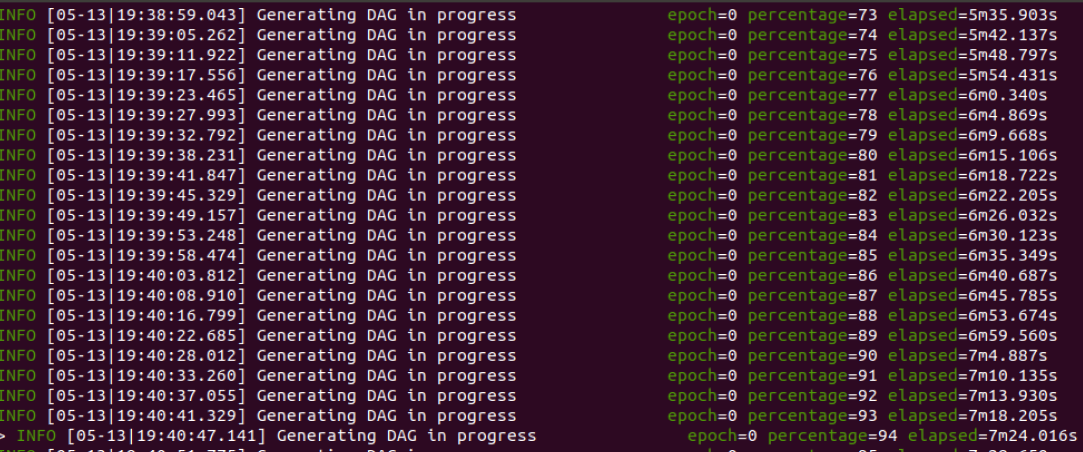
// 第二种查看方式矿工地址的方式

* miner.setEtherbase(eth.accounts[0])

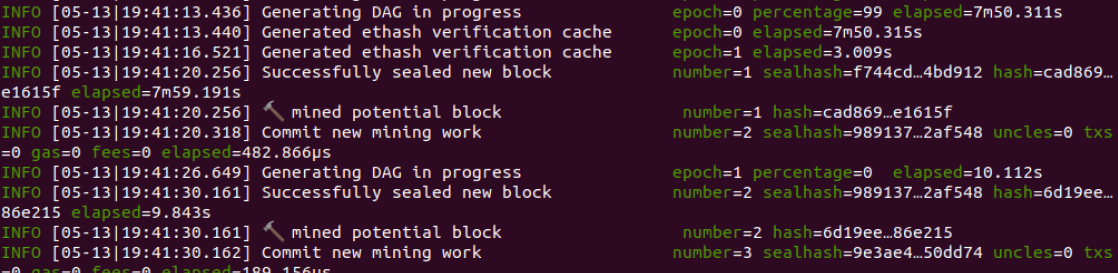
//设置矿工用户。矿工的账户地址即奖励地址，括号里面填写用户地址。这里设置eth.accounts[0]为挖矿的账户地址。

* miner.start()

//开始挖矿，等待，第一次挖矿要好久哦，下面这种情况就是还在准备。



Percentage=100的时候就开挖了



挖一会就有钱了。

* miner.stop()

//输入此命令停止挖矿。你在输入此命令时可能会发现。你边输入电脑一边还在挖矿。但是不用担心它回打断你的输入。你只要一股脑的输入miner.stop()回车就完事了。打断了不要重新输，要继续输入。回车就停下来了。

* eth.getBalance(eth.accounts[0])

//再查账户，发现自己突然有钱了。

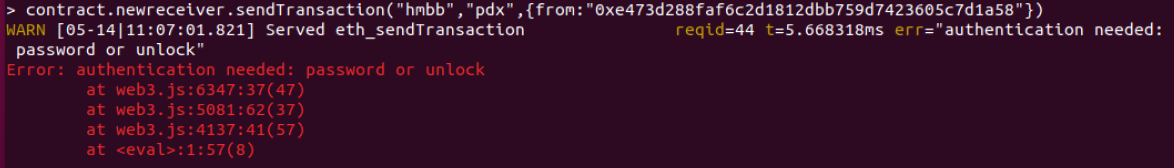
账户有钱啦，就可以用这个账户部署和调用合约辣。

1. 部署合约

* personal.unlockAccount(eth.coinbase)

//解锁账户。若打开geth节点的命令没有--allow-insecure-unlock的话这一步即使输对密码了也报错。出现这种情况，建议exit然后重新按照文档第1点中的命令进入控制台。

/这个命令在调用函数的时候时不时就要用到。



//不解锁就会报错

* var code=”合约的bytecode”

//录入bytecode。Bytecode在文档底部。去复制就好了。双引号建议自己在终端输入。

* var abi=JSON.parse(‘ABI’)

//录入ABI。括号的单引号建议自己在终端输入。ABI在文档底部，复制进去就好了。

* eth.estimateGas({data: code})

//这个命令是用来估计部署合约要用多少gas的。你挖矿之后应该就有钱了。这点gas不在话下

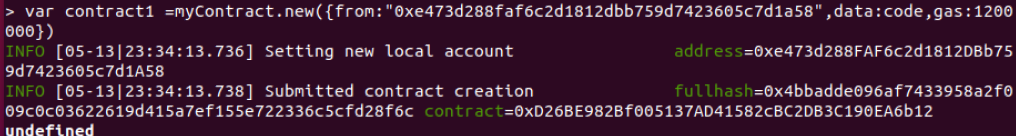
* var myContract = eth.contract(abi)

//创建类

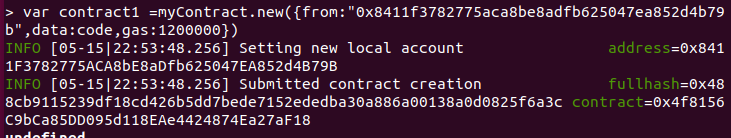
* var contract1 =myContract.new({from:"0xe473d288faf6c2d1812dbb759d7423605c7d1a58",data:code,gas:1200000})

//创建合约实例。括号内部的from后填写的是你账户的地址。gas后填写你愿意支付的gas。原则上大于刚刚估计的值。

//wps这个排版我也是服。在终端这行命令是下面这样的，实际都是一个空格。



Zzx：



* contract1

//输入刚刚合约实例的名字。可以查看合约实例的信息。可以看到address合约地址那一栏没有东西。因为这个私链上现在就只有我们自己这一个用户。没有其它矿工来挖矿把这个合约写入链上。所以我们要自立更生，挖一下矿把合约写入链上。

* txpool.status

//这个命令可以看到有没有待挖矿的交易。我们可以看到有一个。

* miner.start()
* miner.stop()

//挖一会矿再停下来。

* txpool.status

//待挖矿交易变为0。就说明合约已经写入链上了。

* contract1.address

//调用这个可以看到我们部署到链上的合约的地址了。这个值保存下来。Web3里面要用。

1. 如何调用已经部署好的合约

重新启动geth控制台或者使用其他包时，调用原来部署好的智能合约要通过合约地址。

没有写入链上的变量在经历重启geth或者重启web3程序时都会没有了。但是如果部署好的智能合约，并且写入数据时运用的交易都挖矿写入链中了。那么再次通过地址调用时关于合约的数据就都还保留着。

下面示例在geth中调用合约的方法。首先还是要用合约地址实例化一个合约出来。

* var abi=JSON.parse(‘文档中的abi’)

//录入ABI。括号的单引号建议自己在终端输入。ABI在文档底部，复制进去就好了。

* myContract = eth.contract(abi)

//创建类

* contract = myContract.at(address)

//创建合约实例，这里要指名部署过的合约地址。地址先前已经保存。

* personal.unlockAccount(eth.coinbase)

//记得先解锁账户

* contract.newsender.sendTransaction("tmf123456","1","2","3","4",{from:"0xe473d288faf6c2d1812dbb759d7423605c7d1a58"})

//上面是在geth控制台调用transactionn类型（要付费的）函数的方法。括号内部前部分写好参数，最后一项大括号内from要写付费的用户的地址（换成自己的）。

* contract.fetchhyperpv.call("tmf123456")

//调用call类型函数的方法。（不要付费的）

1. 关于函数调用

如果web3.py调用智能合约函数发生错误时。建议在geth控制台试试，具体在geth上怎么调用参见本教程和在群里发的《私链上合约的相关信息和函数2.0》。如果geth控制台调用成功。那么可能是web3.py调用函数格式错误。如果geth控制台调用不了或者和想象中的结果有偏差，那么直接微信问唐梦菲。

调用要收取gas的函数的时候一共三个（下图橙色的三个），调用完了要记得挖矿。因为只有一个用户，自己不挖就没人挖了。挖了矿这些函数写入的数据才能真正被写到链上。下次调用时在链上的数据才能被找到。不然都是临时的。

Python将rsa保存为文件并读取！！：

<https://blog.csdn.net/sinat_25449961/article/details/78039656>

<https://www.jb51.net/article/109884.htm>

先把文件把存在本地

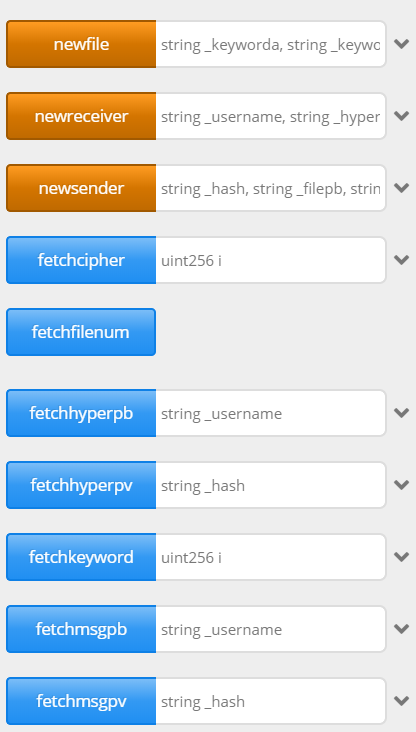
传给智能合约的是对应的文件名

这样传过来我也能在本地查找

然后读取内容

（要安全性考虑的话 需要在用户注册之后 直接把私钥文件下载到用户的电脑上 然后公钥留在这边的服务器上就可以了）

（但是由于我们没有真实用户 所以可以暂时把所有文件都存在我这边）



附：

合约的ABI：

[{\"constant\":false,\"inputs\":[{\"name\":\"\_keyworda\",\"type\":\"string\"},{\"name\":\"\_keywordb\",\"type\":\"string\"},{\"name\":\"\_cipher\",\"type\":\"string\"}],\"name\":\"newfile\",\"outputs\":[{\"name\":\"\",\"type\":\"bool\"}],\"payable\":false,\"stateMutability\":\"nonpayable\",\"type\":\"function\"},{\"constant\":false,\"inputs\":[{\"name\":\"\_username\",\"type\":\"string\"},{\"name\":\"\_hyperbolapb\",\"type\":\"string\"},{\"name\":\"\_filepb\",\"type\":\"string\"}],\"name\":\"newreceiver\",\"outputs\":[{\"name\":\"\",\"type\":\"bool\"}],\"payable\":false,\"stateMutability\":\"nonpayable\",\"type\":\"function\"},{\"constant\":false,\"inputs\":[{\"name\":\"\_hash\",\"type\":\"string\"},{\"name\":\"\_filepb\",\"type\":\"string\"},{\"name\":\"\_filepv\",\"type\":\"string\"},{\"name\":\"\_hyperbolapb\",\"type\":\"string\"},{\"name\":\"\_hyperbolapv\",\"type\":\"string\"}],\"name\":\"newsender\",\"outputs\":[{\"name\":\"\",\"type\":\"bool\"}],\"payable\":false,\"stateMutability\":\"nonpayable\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[{\"name\":\"i\",\"type\":\"uint256\"}],\"name\":\"fetchcipher\",\"outputs\":[{\"name\":\"\",\"type\":\"string\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[],\"name\":\"fetchfilenum\",\"outputs\":[{\"name\":\"\",\"type\":\"uint256\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[{\"name\":\"\_username\",\"type\":\"string\"}],\"name\":\"fetchhyperpb\",\"outputs\":[{\"name\":\"\",\"type\":\"string\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[{\"name\":\"\_hash\",\"type\":\"string\"}],\"name\":\"fetchhyperpv\",\"outputs\":[{\"name\":\"\",\"type\":\"string\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[{\"name\":\"i\",\"type\":\"uint256\"}],\"name\":\"fetchkeyword\",\"outputs\":[{\"name\":\"\",\"type\":\"string\"},{\"name\":\"\",\"type\":\"string\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[{\"name\":\"\_username\",\"type\":\"string\"}],\"name\":\"fetchmsgpb\",\"outputs\":[{\"name\":\"\",\"type\":\"string\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[{\"name\":\"\_hash\",\"type\":\"string\"}],\"name\":\"fetchmsgpv\",\"outputs\":[{\"name\":\"\",\"type\":\"string\"}],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"}]

合约的bytecode：

