前言

EOS官方推荐使用Docker来安装EOS，所以我们接下来的演示环境也切换到了Docker下操作。

1、启动本地私有链

$ sudo docker run --rm --name eosio -d -p 8888:8888 -p 9876:9876 -v /tmp/work:/work -v /tmp/eosio/data:/mnt/dev/data -v /tmp/eosio/config:/mnt/dev/config eosio/eos-dev /bin/bash -c "nodeos -e -p eosio --plugin eosio::wallet_api_plugin --plugin eosio::wallet_plugin --plugin eosio::producer_plugin --plugin eosio::history_plugin --plugin eosio::chain_api_plugin --plugin eosio::history_api_plugin --plugin eosio::http_plugin -d /mnt/dev/data --config-dir /mnt/dev/config --http-server-address=0.0.0.0:8888 --access-control-allow-origin=* --contracts-console"

722167fd21f24f315e3e00a66faeedfbc212db4ed7aa84774417a058ca2482b3
#看到节点ID就表示启动正常了

从上面命令可以看出，我们启动了本地节点，并加载了钱包、钱包API、区块生产等插件。

2、创建钱包

假设当前EOS安装目录为~/eos，最后一行为钱包密码。

$ cd ~/eos

$ cleos wallet create -n simon

"/usr/local/bin/keosd" launched

Creating wallet: simon

Save password to use in the future to unlock this wallet.

Without password imported keys will not be retrievable.

"PW5Kdgdd2ZMmBXeeHxECFkjQiY71ATrDmtvM1JaqKPpnXMpSbaeaA"

3、查看钱包

刚创建的钱包默认是打开+解锁状态的，如钱包名称旁边的*号。

```
$ cleos wallet listWallets:[ "simon *" # ???????900s??15????
??? ]
```

4、创建公私钥

公私钥用于管理账户，账户归属谁，最终认私钥。

$ cleos create key

Private key: 5Jrnorj8fNbQdiQAELDiwJzh7BTATntbYEjLDGgw7psQXKs6TNg

Public key: EOS7je26gtYggvufrSEY38egVk9ssBQF9Go5FrKtXqQgLNDt7rQDY
#ouyang账户

$ cleos create key

Private key:
5JJjmQKveVZmHNGPW9j7prWR1dxSUcg87hePBdZjQgMkLyWpyyg

Public key:
EOS7jfrangqG5R7Gn5YvEk3Um7t1hBDaTPFPacjpSEcWbyGARS7P6
#yangxm账户

5、查看当前钱包导入的私钥对应的公钥

因为当前EOS是V1.0.2，会有个默认的公钥，V1.0.5后去除了，为空才是正常情况
。

```
$ cleos wallet keys[ "EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8Bh
tHuGYqET5GDW5CV"]
```

6、用公钥创建账户

钱包必须处于解锁状态才能创建账户。

创建ouyang，作为合约创建账户。

$ cleos create account eosio ouyang EOS7je26gtYggvufrSEY38egVk9ssBQF9Go5FrKtXqQgLNDt7rQDY

executed transaction: 690a40054f8cbbef3057a801dd9b6b82543961b21bdc4e880a32d8b2ed05e6f9 200 bytes 411 us

eosio <= eosio::newaccount {"creator":"eosio","name":"ouyang","owner":{"threshold":1,"keys":[{"key":"EOS7je26gtYggvufrSEY38egVk...

创建yangxm，作为合约发行方账户，用于接收代币。

```
$ cleos create account eosio yangxm EOS7jfrangqG5R7Gn5YvEk3Um7t1hBDaTPFPacjpSEcWbyGARS7P6executed transaction: 3a8fc91632188e0c30211ee1fcc39e4c08325dee39a11a278d9ad9dd7f4db100 200 bytes 182 us# eosio <= eosio::newaccount {"creator":"eosio","name":"yangxm","owner":{"threshold":1,"keys":[{"key":"EOS7jfrangqG5R7Gn5YvEk3Um7...
```

创建vsiryxm，作为普通用户账户，用于转账测试。

$ cleos create account eosio vsiryxm EOS7jfrangqG5R7Gn5YvEk3Um7t1hBDaTPFPacjpSEcWbyGARS7P6

executed transaction: ead5c5708857cee68c084aae971fc020d9502024088539b34aa10e9e62b46574 200 bytes 642 us

eosio <= eosio::newaccount {"creator":"eosio","name":"vsiryxm","owner":{"threshold":1,"keys":[{"key":"EOS7jfrangqG5R7Gn5YvEk3Um...

7、将私钥导入钱包

试图绕过将私钥导入钱包这一步直接创建EOS合约，发现是报错的，必须将私钥导入钱包之后才能操作。

```
$ cleos set contract ouyang build/contracts/eosio.token/ -p
ouyangReading WAST/WASM from build/contracts/eosio.token/eos
io.token.wasm...Using already assembled WASM...Publishing co
ntract...Error 3090003: provided keys, permissions, and dela
ys do not satisfy declared authorizationsEnsure that you hav
e the related private keys inside your wallet and your walle
t is unlocked.Error Details:transaction declares authority '
{"actor":"ouyang","permission":"active"}', but does not have
 signatures for it.
```

将两个私钥分别导入钱包：

```
$ cleos wallet import -n simon 5Jrnorj8fNbQdiQAELDiwJzh7BTAT
ntbYEjLDGgw7psQXKs6TNgimported private key for: EOS7je26gtYg
gvufrSEY38egVk9ssBQF9Go5FrKtXqQgLNDt7rQDY #???ouyang??$ cleo
s wallet import -n simon 5JJjmQKveVZmHNGPW9j7prWR1dxSUcg87he
PBdZjQgMkLyWpyygimported private key for: EOS7jfrangqG5R7Gn5
YvEk3Um7t1hBDaTPFPacjpSEcWbyGARS7P6 #???yangxm?vsiryxm??
```

再来查看一下导入情况：

```
$ cleos wallet keys[ "EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8Bh
tHuGYqET5GDW5CV", "EOS7je26gtYggvufrSEY38egVk9ssBQF9Go5FrKtX
qQgLNDt7rQDY", #??? "EOS7jfrangqG5R7Gn5YvEk3Um7t1hBDaTPFPacj
pSEcWbyGARS7P6" #???]
```

8、使用ouyang账户部署eosio.token合约

eosio.token合约为系统自带基础合约，eosio为命名空间，token为eosio下的一个合约类。

我们可以通过查看 contracts/eosio.token/eosio.token.hpp
文件代码得知。详细解读>>

```
$ cleos set contract ouyang build/contracts/eosio.token/ -p
ouyangReading WAST/WASM from build/contracts/eosio.token/eos
io.token.wasm...Using already assembled WASM...Publishing co
ntract...executed transaction: bd0ae1133c9b2cd7fd56fbd66ed6c
a15143de99f3669676ce120647f959f88ab 8104 bytes 2379 us# eosi
```

```
o <= eosio::setcode {"account":"ouyang","vmtype":0,"vmversio
n":0,"code":"0061736d01000000017e1560037f7e7f0060057f7e7e7f7
...# eosio <= eosio::setabi {"account":"ouyang","abi":"0e656
f73696f3a3a6162692f312e30010c6163636f756e745f6e616d65046e616
d6505087...
```

9、创建代币

代币名称CBT，发行总量10亿。代币归属谁，最终只认账户，而账户归属谁是认密钥的。

$ cleos push action ouyang create '["eosio", "1000000000.0000 CBT", 0, 0, 0]' -p ouyang

executed transaction: 248a3f612983c42c0fbdaaccb2243861073c675a442bf7bb049bc1f210a1d9d1 120 bytes 835 us

ouyang <= ouyang::create {"issuer":"eosio","maximum_supply":"1000000000.0000 CBT"}

10、将代币总量转给发行方

```
$ cleos push action ouyang issue '["yangxm", "1000000000.000
0 CBT", "ouyang?????yangxm"]' -p eosioexecuted transaction:
e0e3cbed0512309ea1215a06531809425699b32864d0e459cfcd9eac5b46
a6b8 144 bytes 2008 us# ouyang <= ouyang::issue {"to":"yangx
m","quantity":"1000000000.0000 CBT","memo":"ouyang?????yangx
m"}# ouyang <= ouyang::transfer {"from":"eosio","to":"yangxm
","quantity":"1000000000.0000 CBT","memo":"ouyang?????yangxm
"}# eosio <= ouyang::transfer {"from":"eosio","to":"yangxm",
"quantity":"1000000000.0000 CBT","memo":"ouyang?????yangxm"}
# yangxm <= ouyang::transfer {"from":"eosio","to":"yangxm","
quantity":"1000000000.0000 CBT","memo":"ouyang?????yangxm"}$
 cleos get currency balance ouyang yangxm CBT1000000000.0000
 CBT
```

11、转账

给vsiryxm普通用户账户转账。

$ cleos push action ouyang transfer '["yangxm","vsiryxm","10000.0000 CBT","yangxm给vsiryxm转账1000个CBT"]' -p yangxm

executed transaction: c4bf101bd0e2763bd5c14c78d6e273bc71cb696e9c4832b71ee0b2b607283117 160 bytes 1912 us

ouyang <= ouyang::transfer {"from":"yangxm","to":"vsiryxm","quantity":"10000.0000 CBT","memo":"yangxm给vsiryxm转账10000个CB...

yangxm <= ouyang::transfer {"from":"yangxm","to":"vsiryxm","quantity":"10000.0000 CBT","memo":"yangxm给vsiryxm转账10000个CB...

vsiryxm <= ouyang::transfer {"from":"yangxm","to":"vsiryxm","quantity":"10000.0000 CBT","memo":"yangxm给vsiryxm转账10000个CB...

12、查看余额

ouyang为创建合约账户，yangxm为发行方账户，vsiryxm为普通用户账户。

```
$ cleos get currency balance ouyang yangxm CBT #??yangxm??99999990000.0000 CBT$ cleos get currency balance ouyang vsiryxm CBT #??vsiryxm??10000.0000 CBT
```

小结

1、account账户是EOS网络的基本单位，可以理解成现实中的法人，最终是保存在EOS公链上的，而公私钥、钱包、钱包密码是保存在本地的。

2、一个account账户至少可以用两个权限组owner、active来进行管理，同一个公钥拥有这两个权限组，也可以是不同公钥拥有这两个权限组，当然这样的权限还可以扩展成多个，只要你需要，把你的亲友团都加入到这个账户中来，一旦密钥被盗，不用担心，亲友团都可以帮助你找回密钥，37天的审核期，让你有足够的时间来操作，但前提是要设计好这个权限组成员和阀值。

3、钱包的解锁时间默认为900s，超过了要重新解锁，解锁状态为加了*星号。