Wiki文章题目： truechain\_testing\_network\_setup, True链开发实战篇-安装系统环境

作者：作者:大野 审阅:rectinajh

**TrueChain Tutorials - Testing**

**Network Setup**

this tutorials provides an introduction to Testing Network Setup . it demonstrates how to build,run, and other node jion network.

**System requirements**

**Note: this article operates under the OSX system**

1. golang
2. gcc

**step 1**

install the requirements in terminal

brew install gcc golang

**step 2**

set golang environment

* GOPATH
* GOROOT

echo "GOPATH=your\_working\_dir" >> .bash\_profile

echo "GOROOT=go\_install\_dir>" >> .bash\_profile

finish its, we will have a test where to get the environments

go env



**Build TrueChain Project**

get the TrueChain Engineering Project

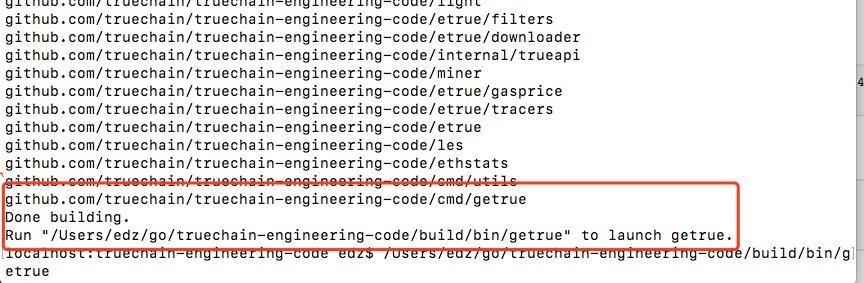
in your working dir

go get <https://github.com/truechain/truechain-engineering-code>

build it

entry into the **truechain-engineering-code** directory

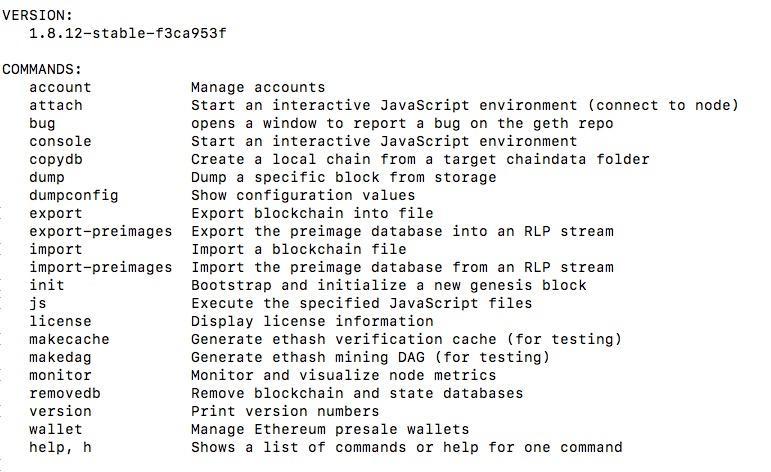
make getrue

When compiled, the path for the. Getrue command is the project directory / build/bin/. Like diagram 

**The getrue command**

getrue view command help, as shown in figure

geture —help



when we did this step it means that successful built the project and run it gracefully

**Deploy TrueChain Test Network**

step1. Create an Test Directory like Test

step2. Cmd/getrue reference source code in the genesis.json file, the contents of the file as follows.

{

"config": {

"chainId": 10,

"homesteadBlock": 0,

"eip155Block": 0,

"eip158Block": 0

},

"alloc" : {

"0x970e8128ab834e8eac17ab8e3812f010678cf791" : { "balance" : "90000000000000000000000"},

"0x68f2517b6c597ede0ae7c0559cdd4a84fd08c928" : { "balance" : "10000000000000000000000"}

},

"coinbase" : "0x0000000000000000000000000000000000000000",

"difficulty" : "0x200",

"extraData" : "",

"gasLimit" : "0x2fefd8",

"nonce" : "0x0000000000000042",

"mixhash" : "0x0000000000000000000000000000000000000000000000000000000000000000",

"parentHash" : "0x0000000000000000000000000000000000000000000000000000000000000000",

"timestamp" : "0x00"

}

step3. Create creation blocks. Execute the following command

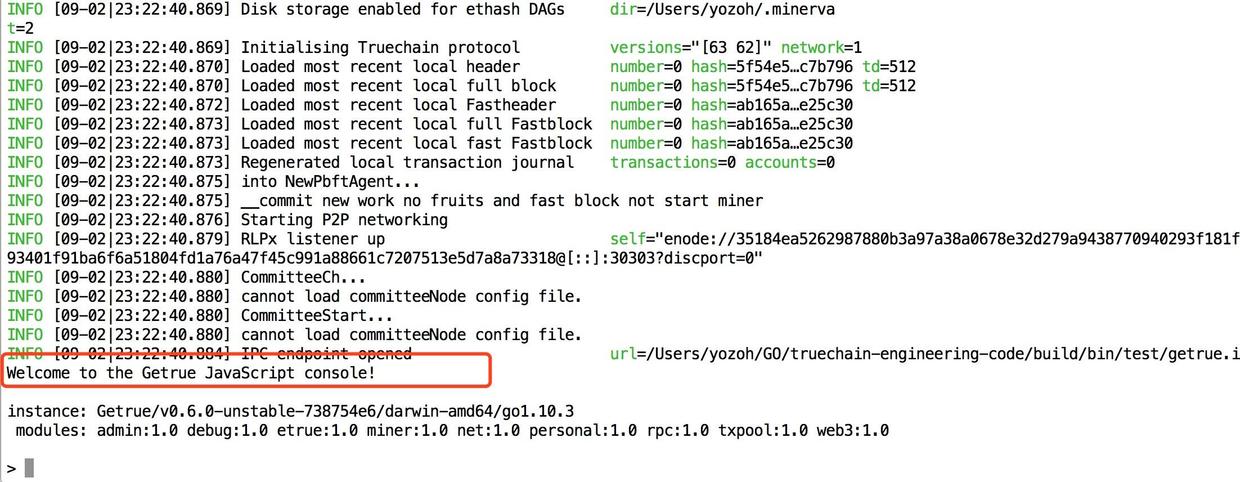
getrue --datadir Test init path/to/cmd/getrue/genesis.json

parameter --datadir,Used to specify a directory.After the above command is executed, there are two folders, **getrue** and **keystore**, in the directory. The **getrue** folder is used to store the relevant data of the chain, and the user information of the chain is stored in the **keystore** folder as shown in the figure.

step4. Start the test chain start node. Execute the following command:

getrue --datadir Test --nodiscover console

After starting successfully, as shown in the following figure



**Useage**

**step1. Create a new account**

> personal.newAccount()

Passphrase:

Repeat passphrase:

"0xce0f1ee66b1695e33d5e8cb9dd79525764d68a48"

>

Enter the password twice and create a new account. Next, you can view the account

> etrue.accounts

["0xce0f1ee66b1695e33d5e8cb9dd79525764d68a48"]

>

**step2. Query account balance**

> etrue.getBalance(etrue.accounts[0])

0

>

**step3. Start mining**

miner.start(1)

INFO [09-02|23:42:27.187] Updated mining threads threads=1

INFO [09-02|23:42:27.188] Transaction pool price threshold updated price=18000000000

INFO [09-02|23:42:27.188] Etherbase automatically configured address=0xCE0F1EE66B1695E33D5e8cb9Dd79525764D68A48

INFO [09-02|23:42:27.188] Starting mining operation

null

**Stop mining**

miner.stop

**Add additional nodes to the test network**

**step1.Separate start**

1.Committee node startup parameter

./build/bin/getrue --datadir ./data --networkid 1004 --testnet --etherbase 0x8a45d70f096d3581866ed27a5017a4eeec0db2a1 console --singlenode --nodiscover --bftkeyhex "c1581e25937d9ab91421a3e1a2667c85b0397c75a195e643109938e987acecfc"

2.Digging block startup parameters

./build/bin/getrue --datadir ./data --networkid 1004 --testnet --nodiscover --etherbase 0x8a45d70f096d3581866ed27a5017a4eeec0db2a1 console --mine

3.Digging fruit start parameter

./build/bin/getrue --datadir ./data --networkid 1004 --testnet --nodiscover --etherbase 0x8a45d70f096d3581866ed27a5017a4eeec0db2a1 console --mine --minefruit

**step2. In other host local consoles, use admin.nodeInfo.enode to get enode information.**

admin.nodeInfo.enode "enode://35184ea5262987880b3a97a38a0678e32d279a9438770940293f181f4790738011f93401f91ba6f6a51804fd1a76a47f45c991a88661c7207513e5d7a8a73318@[::]:30303?discport=0"

**step3. The ip address and encoe information of other host nodes must be filled in when admin.addPeer (), is connected to other host nodes in their own node console so that the nodes of other hosts can be connected.**

admin.addPeer("enode://35184ea5262987880b3a97a38a0678e32d279a9438770940293f181f4790738011f93401f91ba6f6a51804fd1a76a47f45c991a88661c7207513e5d7a8a73318@[47.92.224.44]:30303?discport=0")

**step4. Test whether the two nodes are connected successfully as shown in figure**

**step5. After a successful connection between two nodes, the next two nodes will automatically synchronize the transaction as shown in the figure**