Ethereum Development Tutorial

Written by: Pham Hoai Luan

Last revision: May, 2018

How to send transaction by Geth client on Ethereum network

In the previous tutorial, we have successfully completed creating the local private multi-node Ethereum network. To comprehend deeply Ethereum protocol, this tutorial focuses on transmitting the transaction on Ethereum network. I suppose that there is a scenario, as shown in Figure 1, like: "There are 3 nodes on Ethereum network. An account of node 1 send a transaction for the account of node 2. The account of node 3 acts as miner to write the transaction into ledger". Note that in this tutorial we have **not** related to **smart contract.**

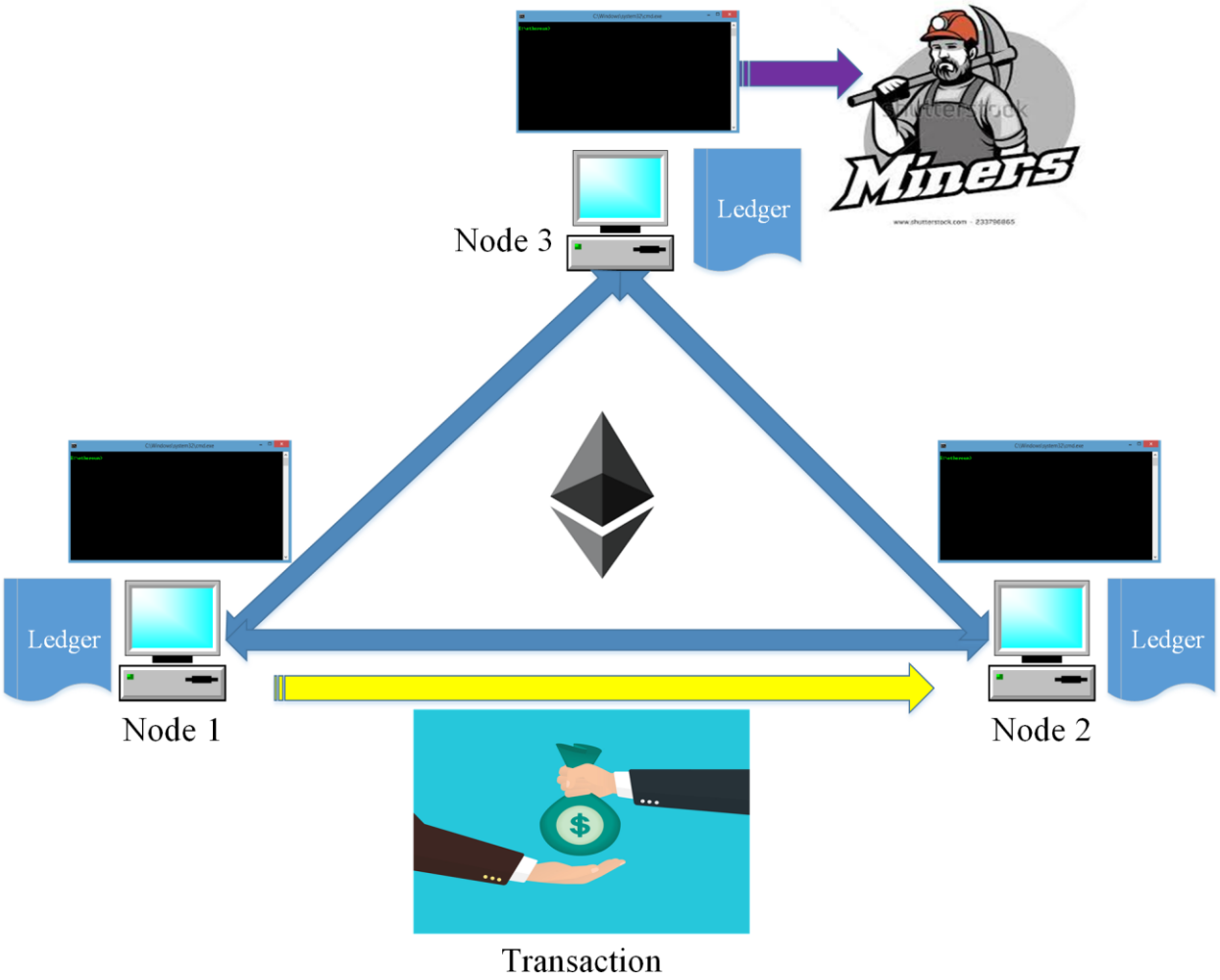

Figure 1. A scenario of sending transaction by Geth client

Now, let’s get started.

Create a network with 3 nodes

I will not explain too much details in this step because it is guided entirely in the previous tutorial. I will go quickly through images.

Starting at node 1

Create a genesis.json file like:

```
{
  "config": {
    "chainId": 1985,
    "homesteadBlock": 0,
    "eip155Block": 0,
    "eip158Block": 0
  },
  "difficulty": "4000",
  "gasLimit": "2100000",
  "alloc": {}
}
```

And put it in an “ethereum_transation” folder as shown in Figure 2.

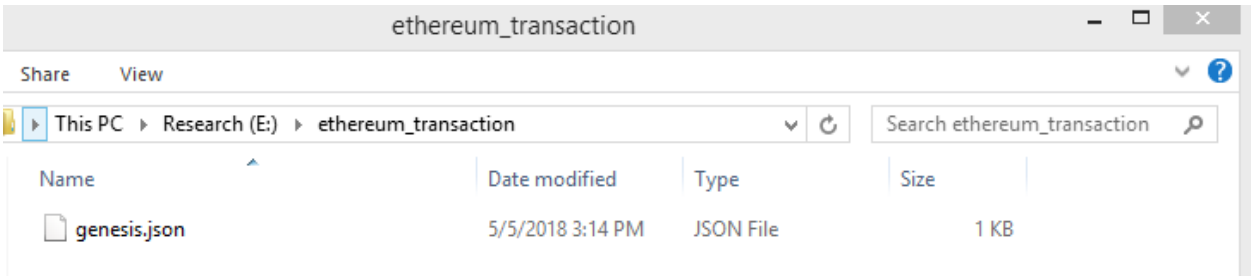

Figure 2. Genesis file

Right-click and choose “Open with Command Prompt as shown in Figure 3.

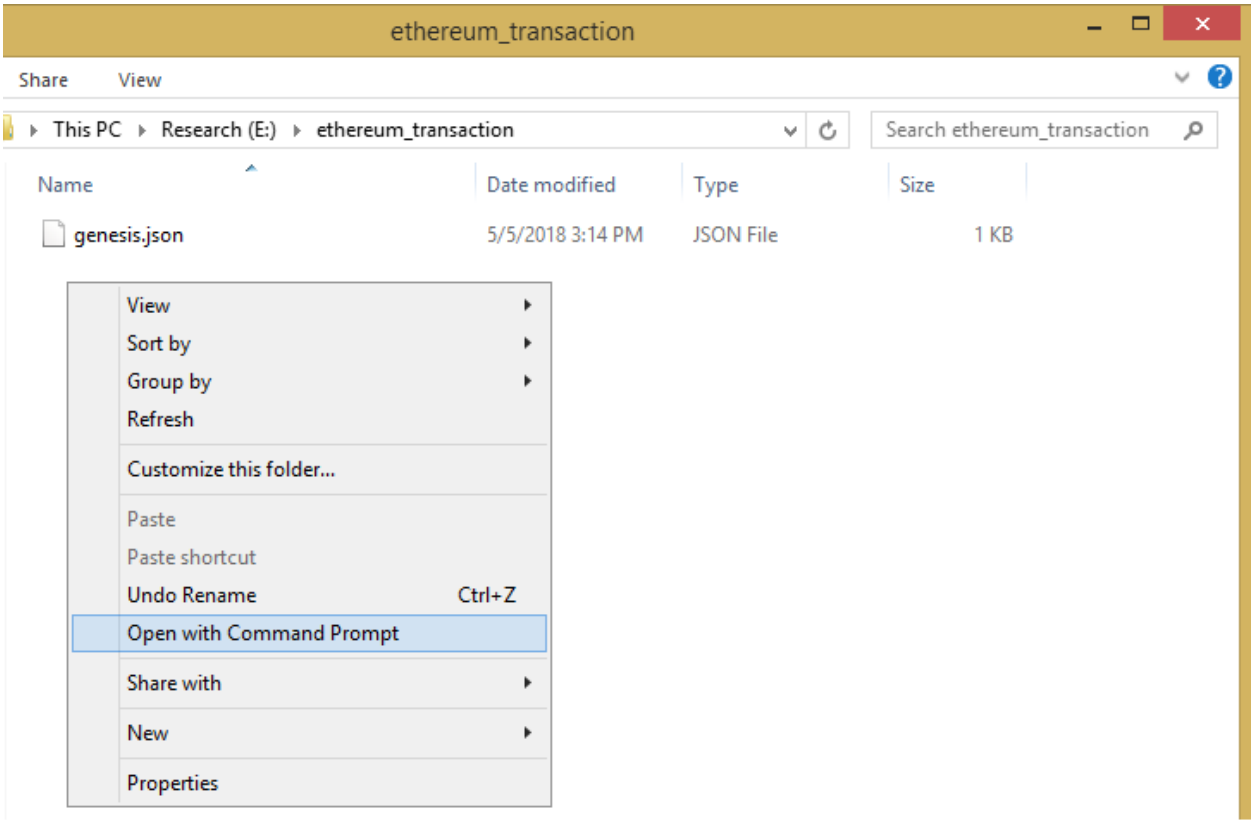

Figure 3. Open with Command Prompt

Then the terminal window will appear as Figure 4.

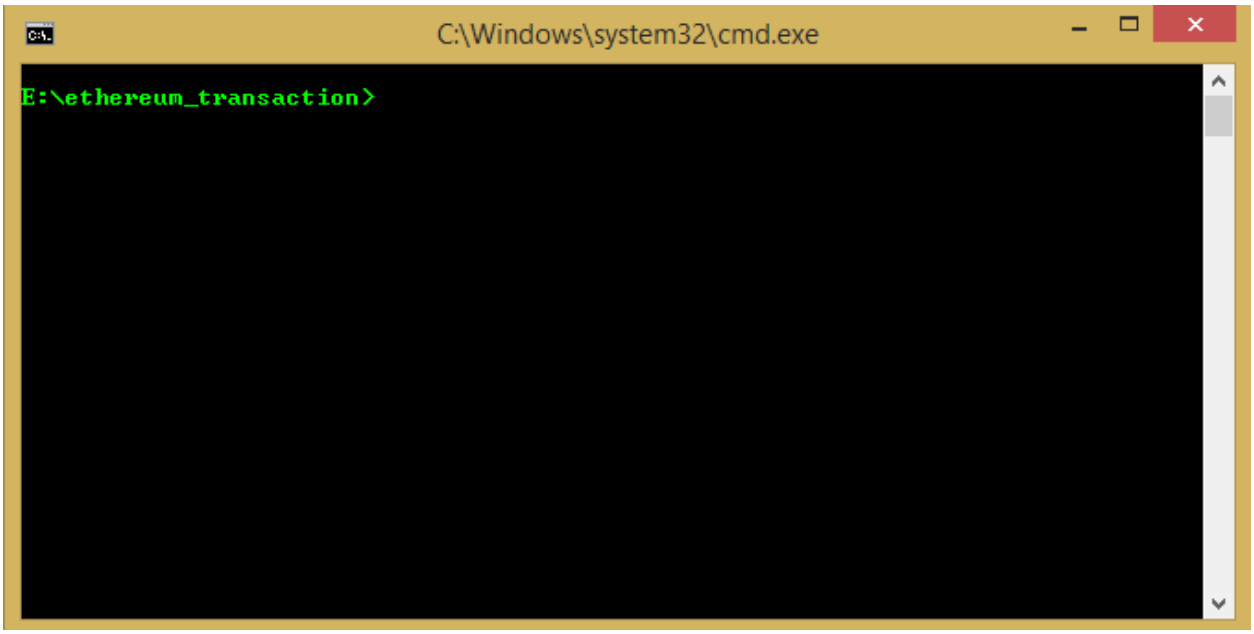

Figure 4. Command Prompt terminal

Type "*geth --datadir "./node01" init "genesis.json"*"" in Command Prompt terminal like Figure 5.

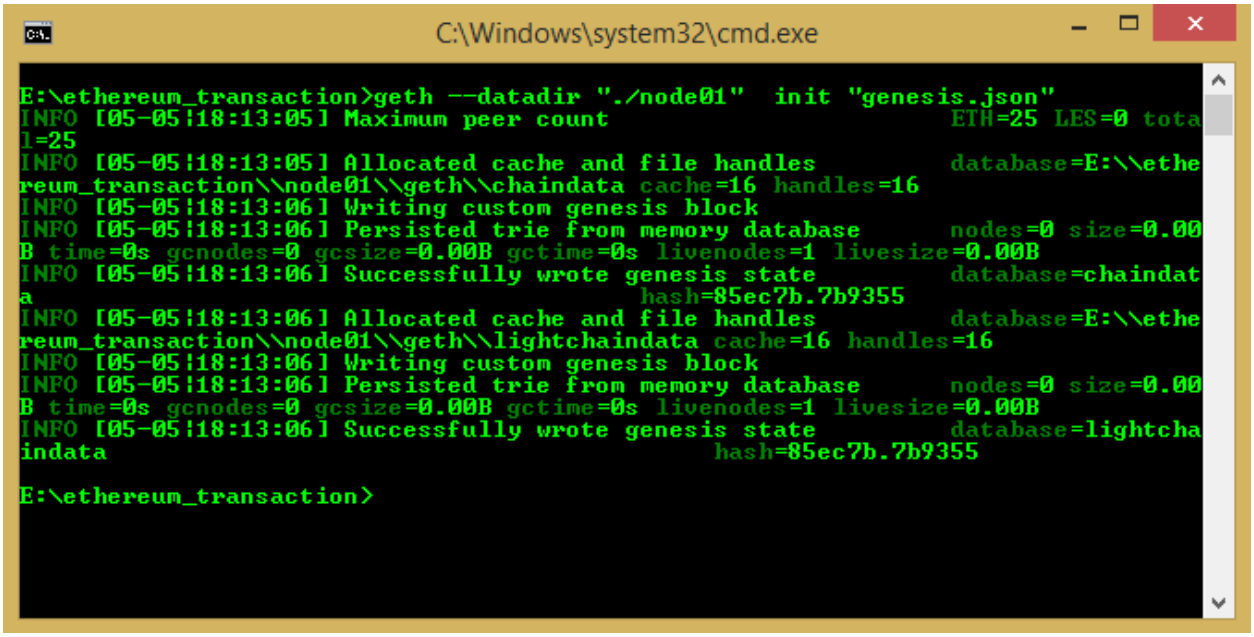

Figure 5. Create genesis block at node 1

Then type "*geth --datadir "./node01" --port 30301 --networkid 1234 --rpc --rpcport "8545" --rpccorsdomain "*" console 2>console.log*" in Command Prompt terminal like Figure 6.

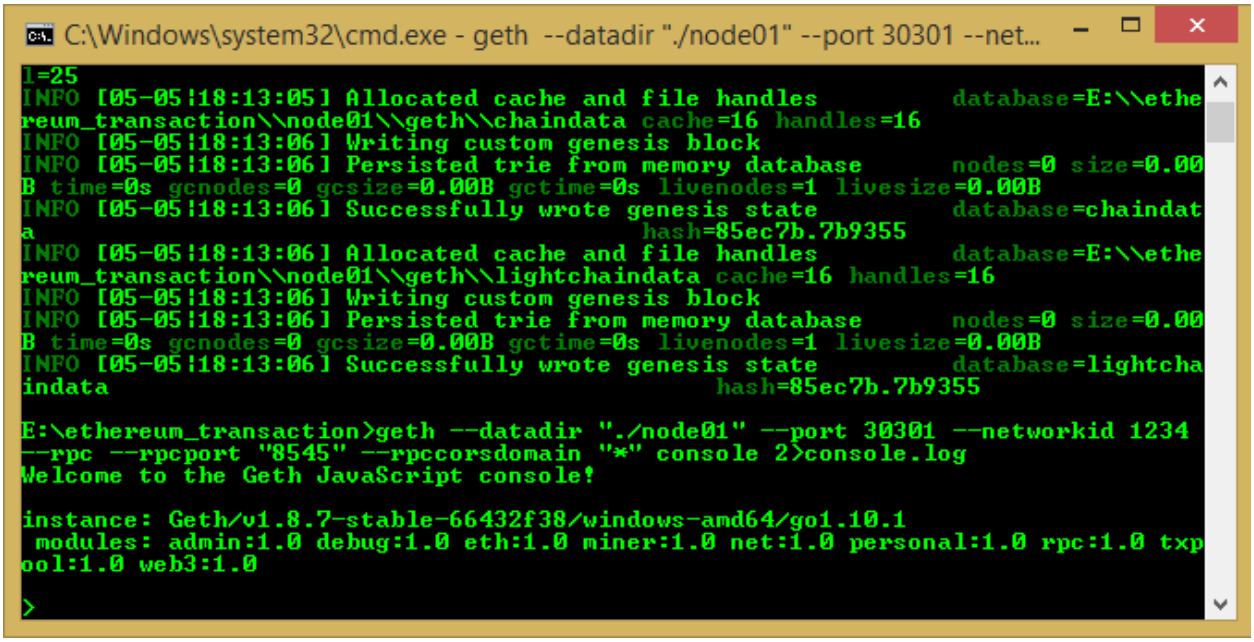

Figure 6. Open Geth console at node 1

Starting at node 2

Right-click and choose “Open with Command Prompt as shown in Figure 3.Then the terminal window will also appear as Figure 4. Type “*geth --datadir "./node02" init "genesis.json"*” in Command Prompt terminal like Figure 7.

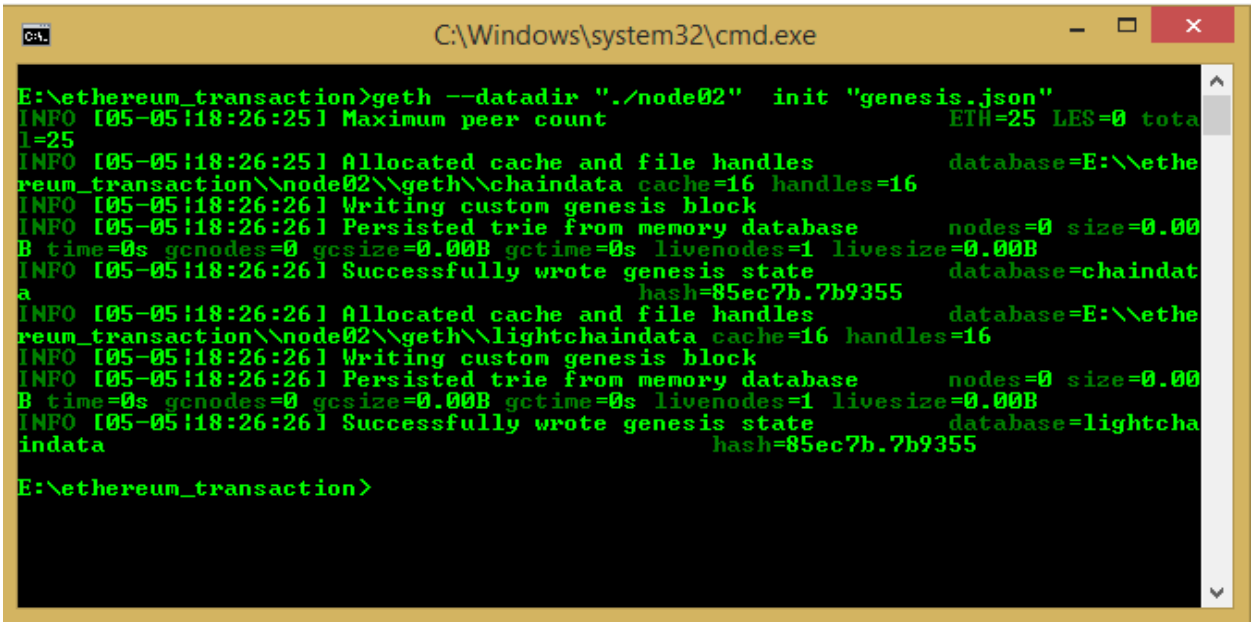

Figure 7. Create genesis block at node 2

Then type “*geth --datadir "./node02" --port 30302 --ipcdisable --networkid 1234 console 2> console2.log*” in Command Prompt terminal like Figure 8.

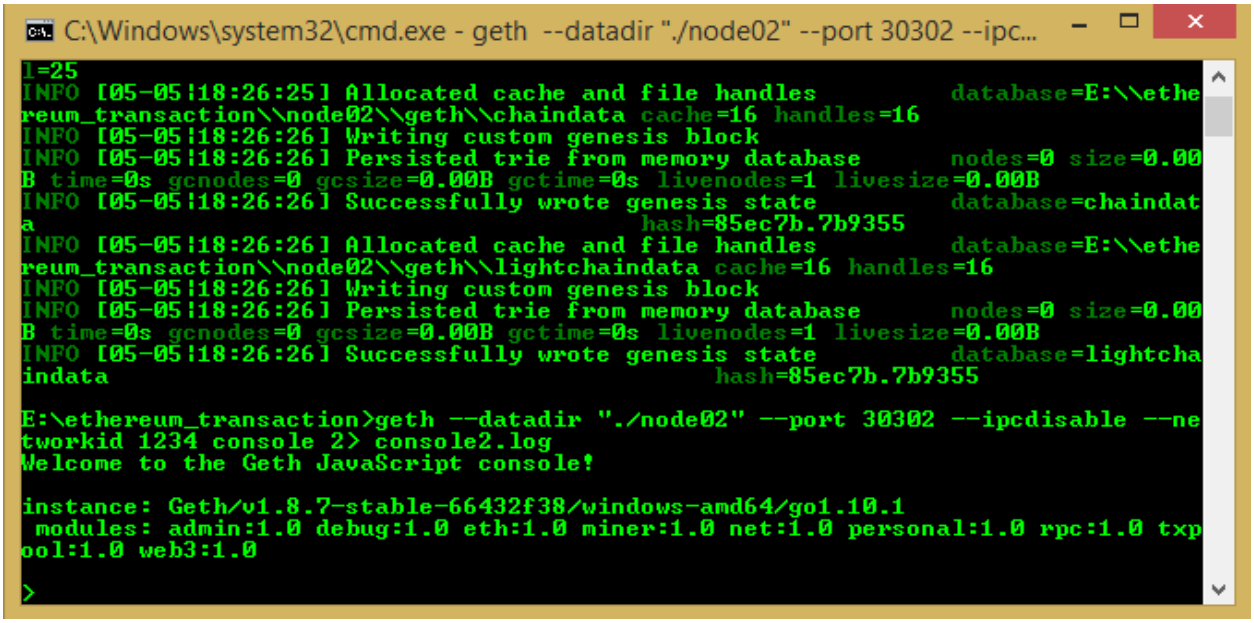

Figure 8. Open Geth console at node 2

Starting at node 3

Right-click and choose “Open with Command Prompt as shown in Figure 3.Then the terminal window will also appear as Figure 4. Type “*geth --datadir "./node03" init "genesis.json"*”” in Command Prompt terminal like Figure 9.

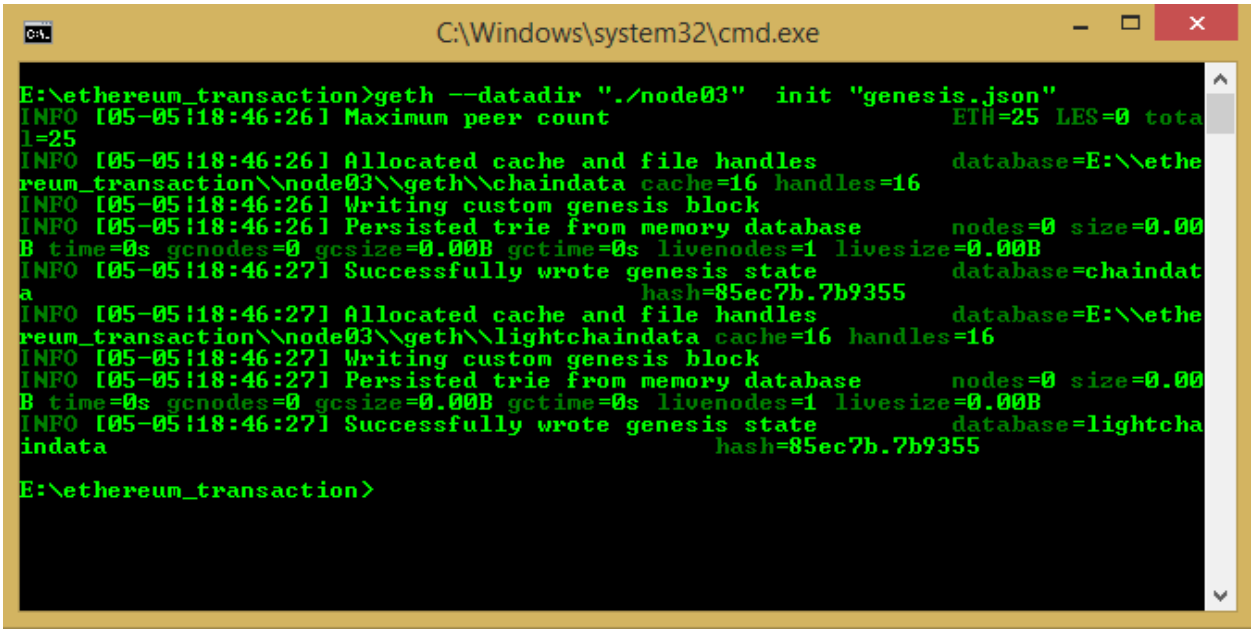

Figure 9. Create genesis block at node 3

Then type “*geth --datadir "./node03" --port 30303 --ipcdisable --networkid 1234 console 2> console3.log*” in Command Prompt terminal like Figure 10.

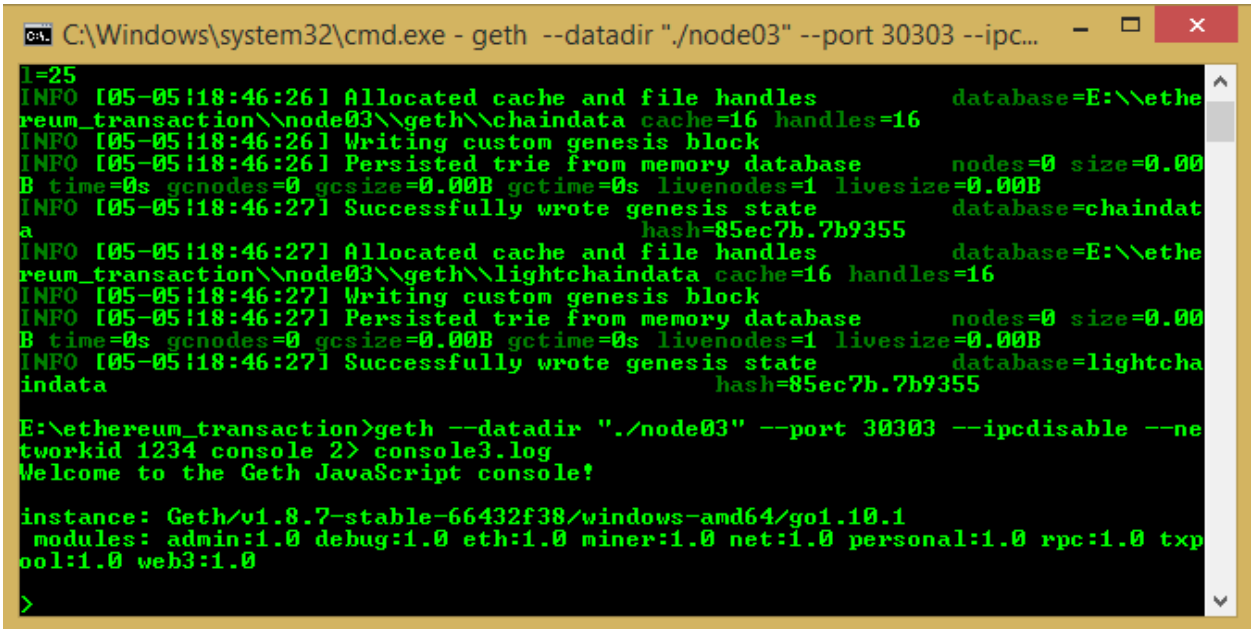

Figure 10. Open Geth console at node 3

Linking 3 nodes

At **node 2,** we type “admin.nodeInfo.enode” like Figure 11.

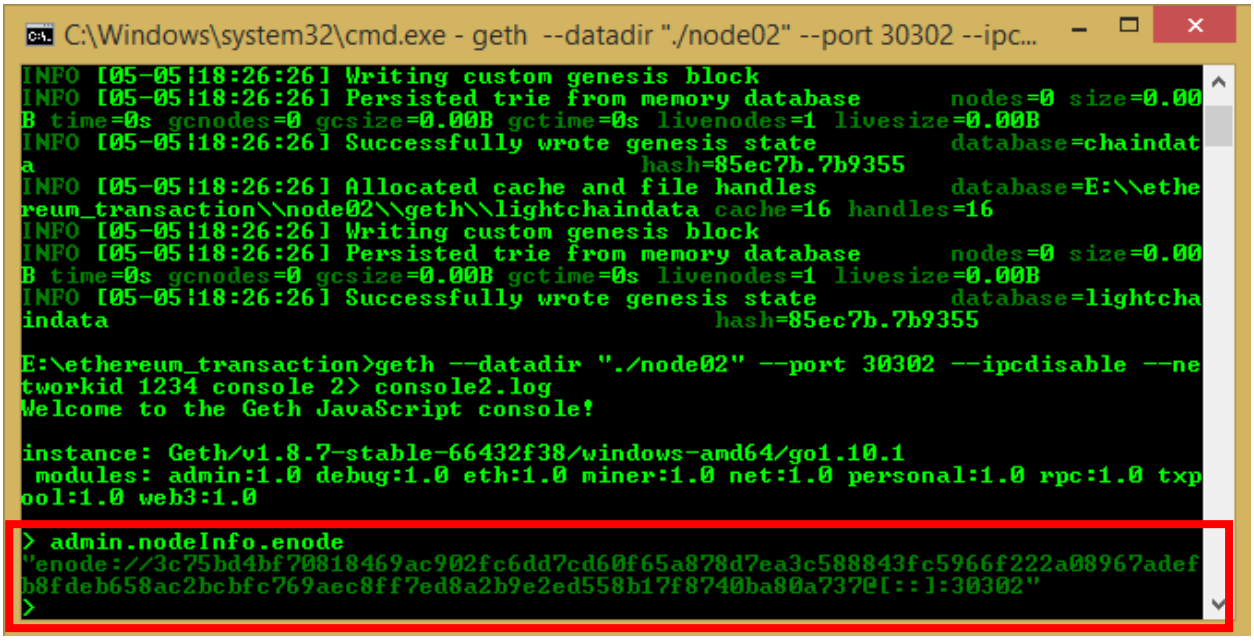

Figure 11. Get enode address at node 2

At **node 1,** we type “admin.addPeer(enode address **node 2**) like Figure 12.

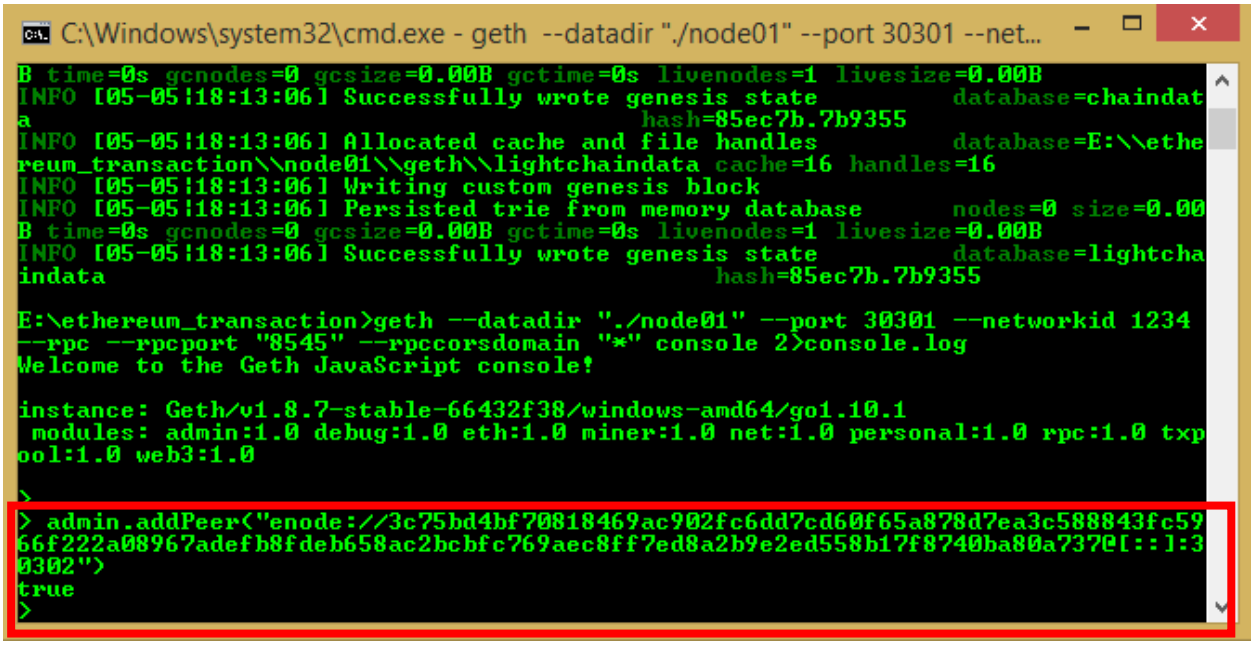

Figure 12. Add node 2 at node 1

At **node 3,** we type “admin.nodeInfo.enode” like Figure 13.

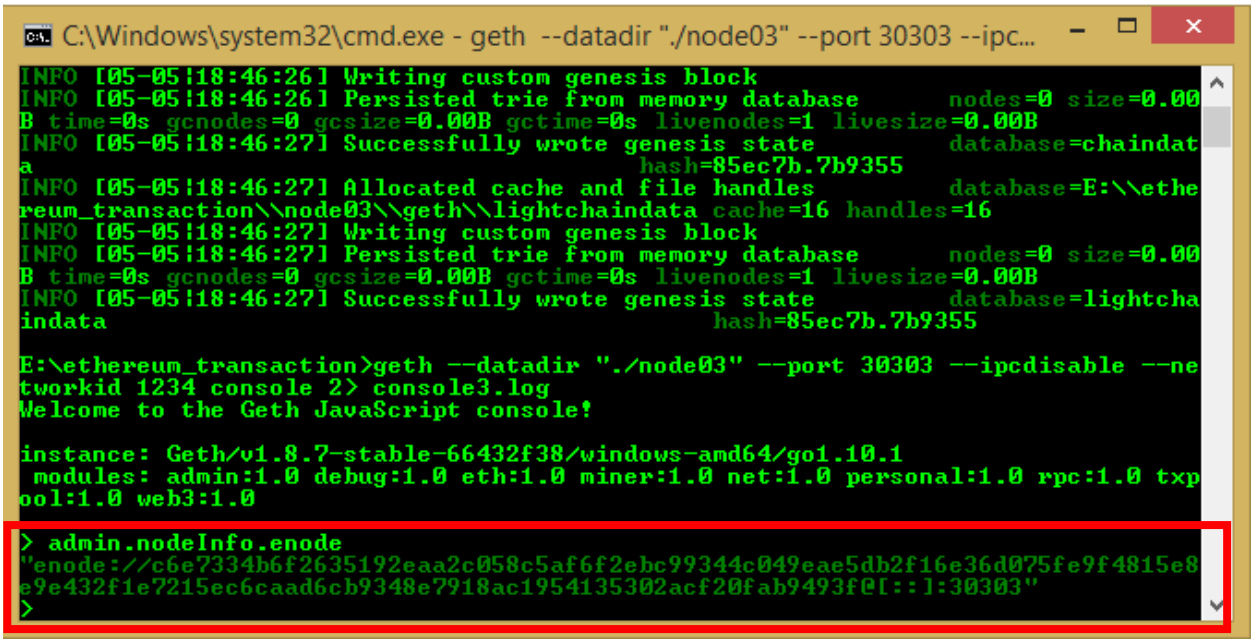

Figure 13. Get enode address at node 2

At **node 1,** we type “admin.addPeer(enode address **node 3**) like Figure 14.

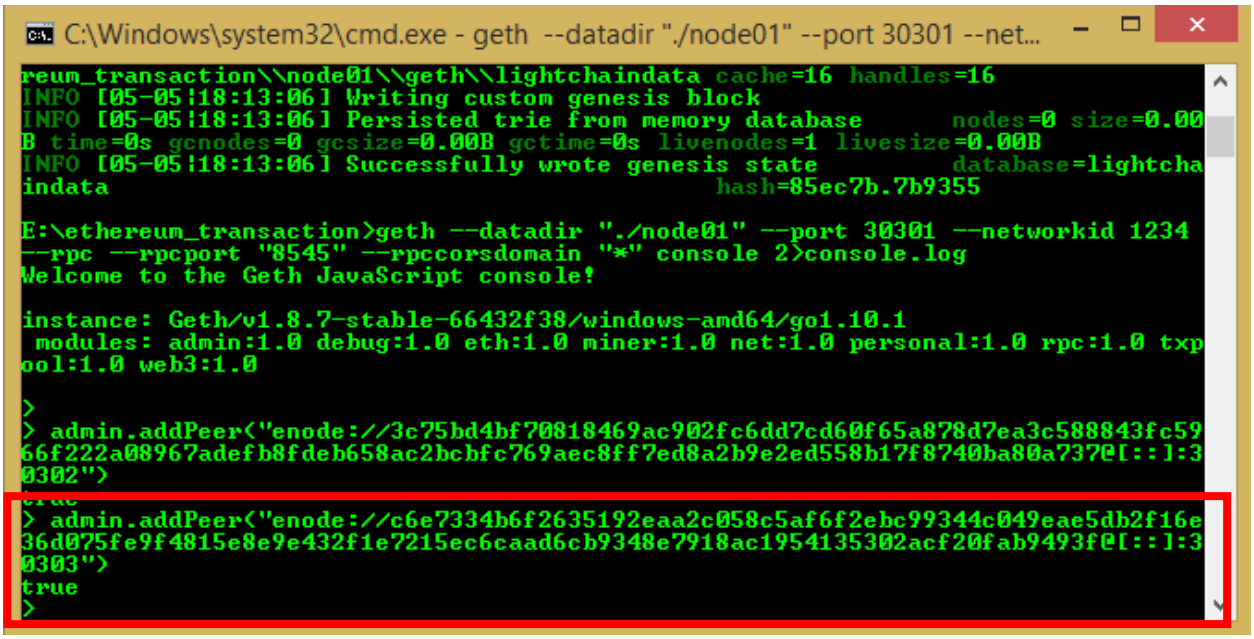

Figure 14. Add node 3 at node 1

Now, you can check that **node 1** sees **node 2** and **node 3** as its **peers** by typing "net.peerCount" and "admin.peers" as shown in Figure 15.

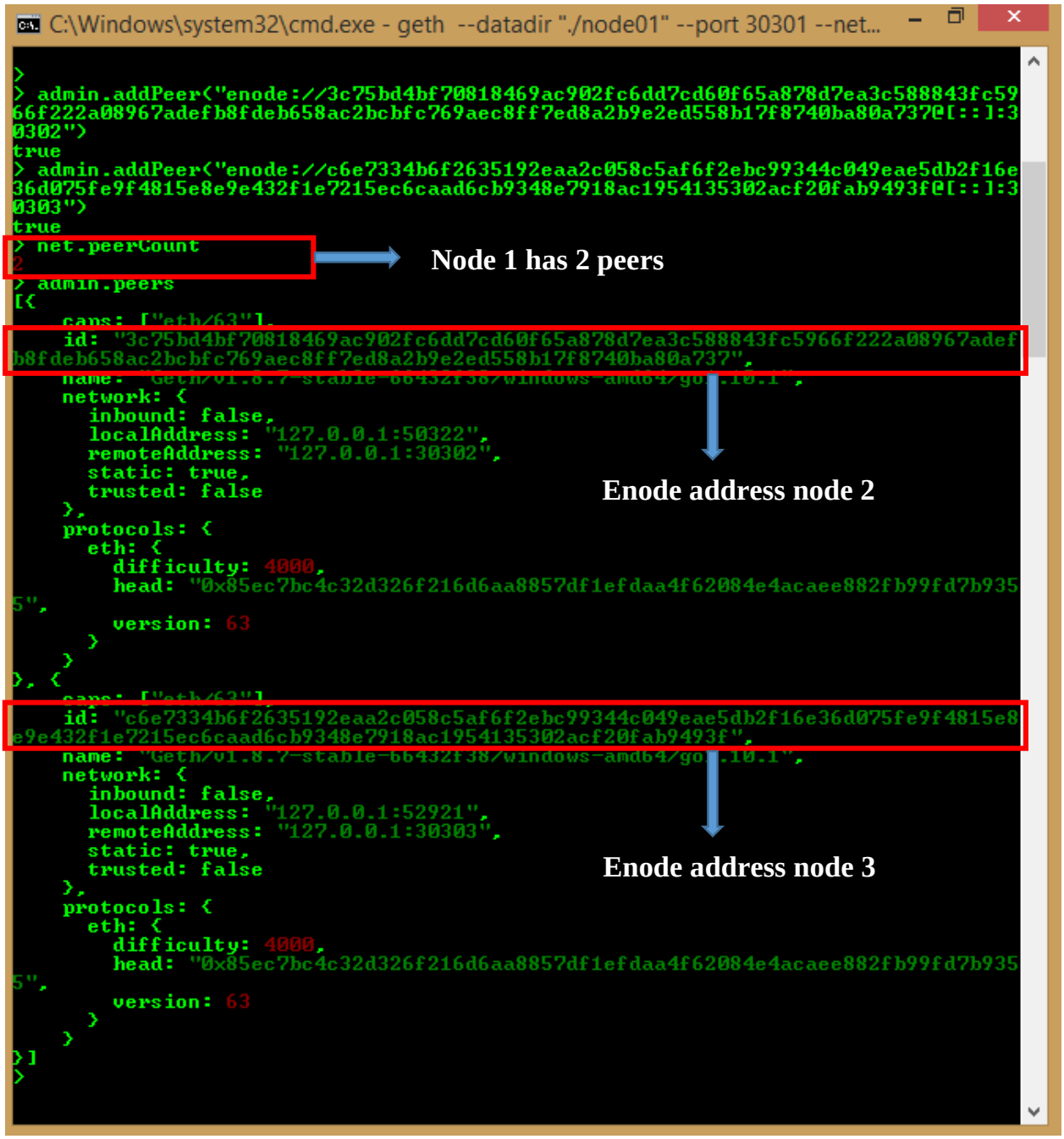

Figure 15. Check peer-to-peer network

Send transaction & miner

Create account

To be able to send transactions as well as mine, we must create an account at each node. We type "*personal.newAccount()*" at each node to create the account like Figure 16.

Figure 16. Create account at node 1

Then we have to unlock account to transmit transaction or mine by typing "*personal.unlockAccount('account','')*" as shown in Figure 17, just like login your account into facebook to post status or chat.

Figure 17. Unlock account at node 1

Send transaction

Accounts created in the previous steps do not have any ether, meaning that the balance is 0. How do we know that? At **node 1**, we type *"eth.getBalance(eth.coinbase)"* t o show the balance of node 1 as shown in Figure 18.

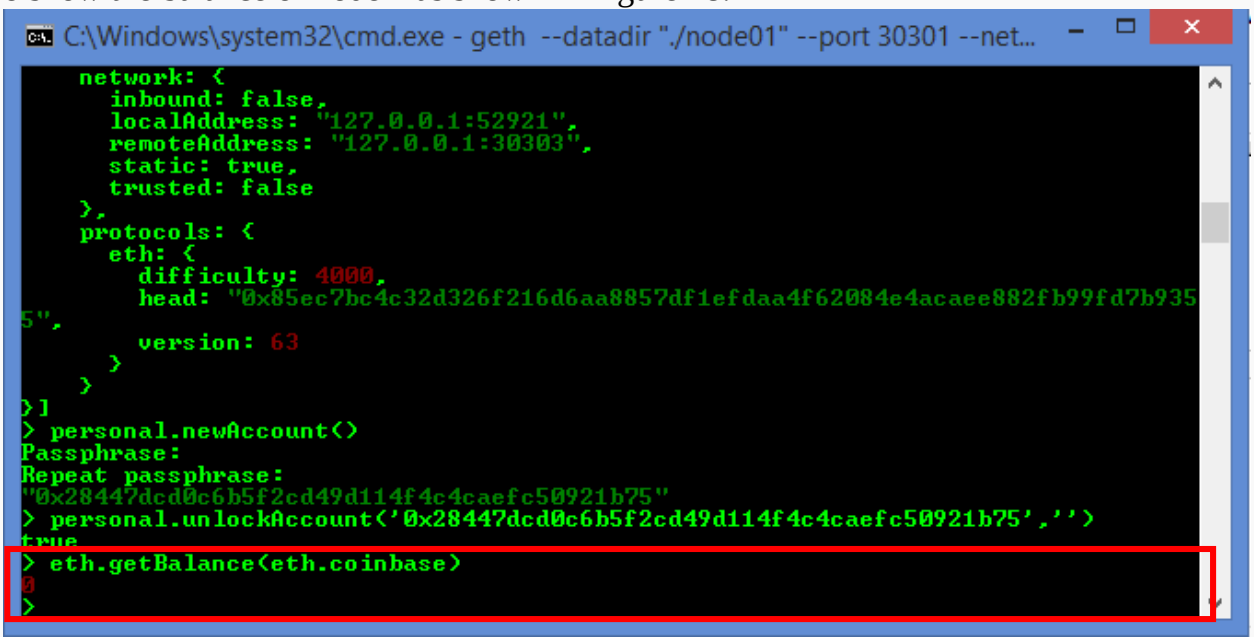

Figure 18. Unlock account at node 1

If your account doesn't have any ethers, you cann't totally send the transaction. So I have a little trick to get some ethers as follow: You type "*miner.start (1)"* in few minutes as shown in Figure 19. Then type "*miner.stop()*".

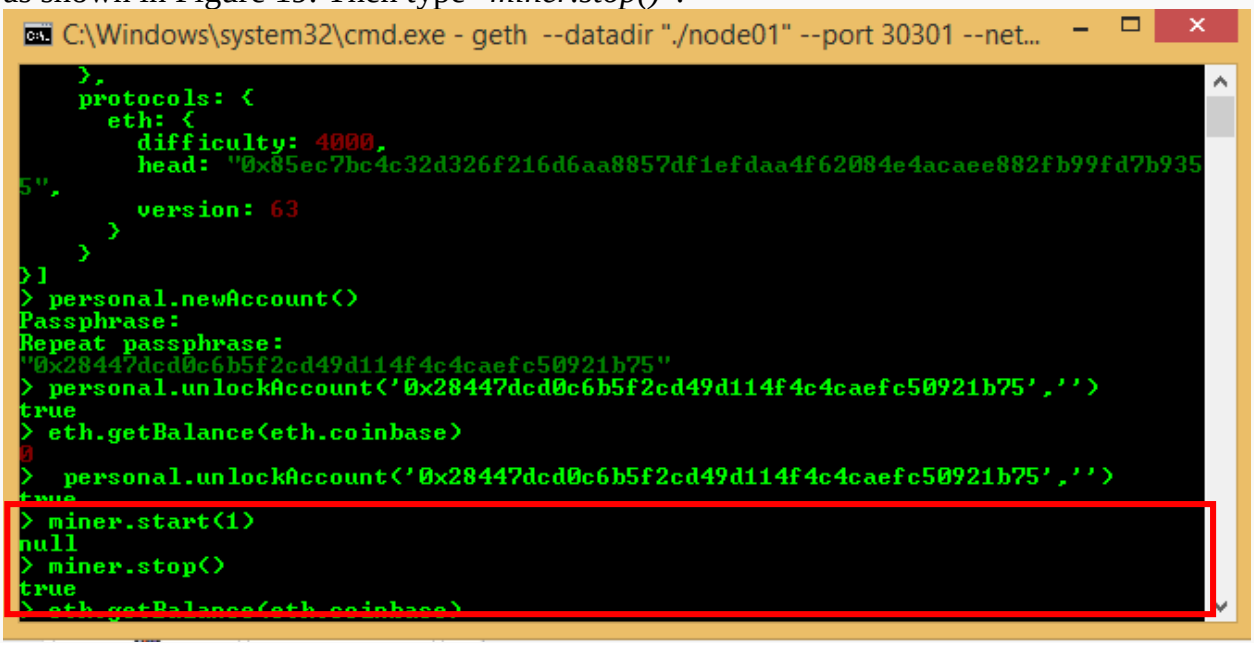

Figure 19. Get ether at node 1

Now we can enjoy our results by typing "*eth.getBalance (eth.coinbase*)" again. As you can see, we have 215000000000000000000 ethers like Figure 20.

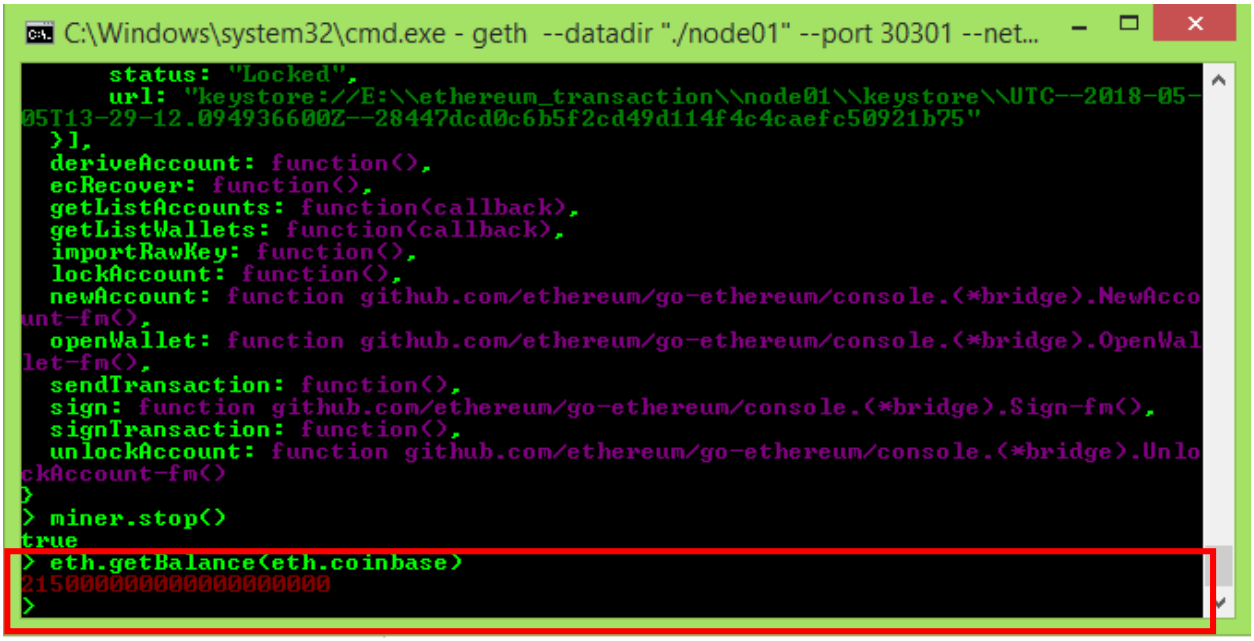

Figure 19. Check ether at node 1

We also type "*eth.getBalance (eth.coinbase*)" to make sure that the balance of node 2 is 0 as shown in Figure 20.

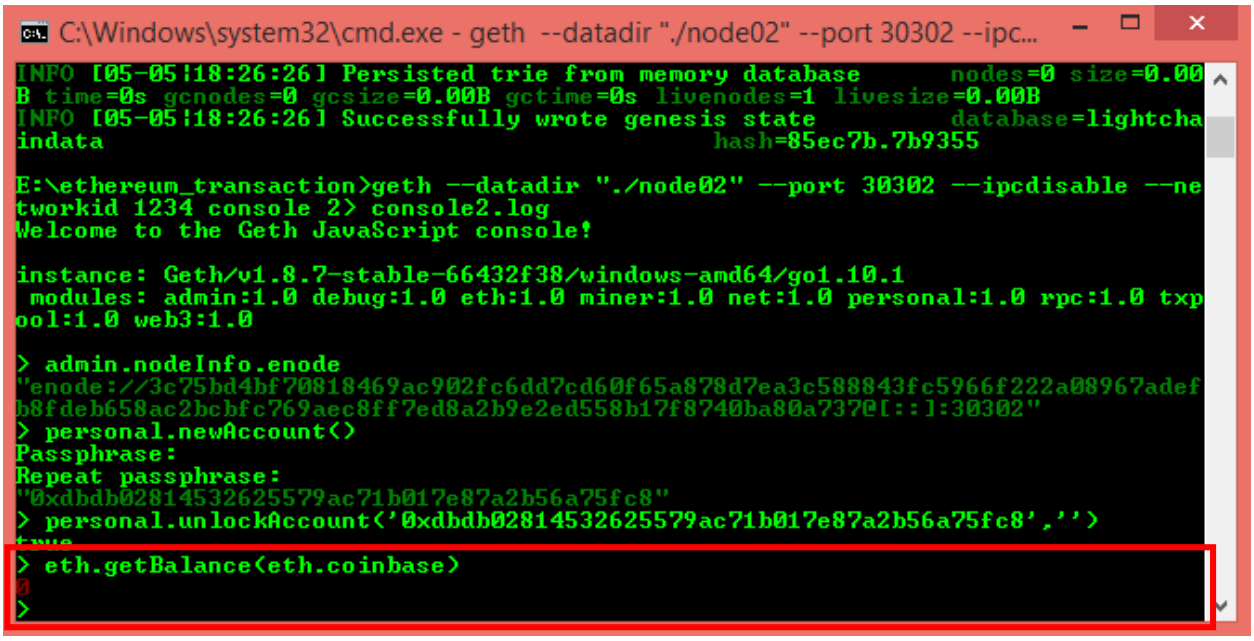

Figure 20. Check ether at node 2

Start sending ether from node 1's account to node 2's account by typing:

eth.sendTransaction({from: eth.coinbase,

to: "0xdbdb02814532625579ac71b017e87a2b56a75fc8", value: ***1000****})*

with *"0xdbdb02814532625579ac71b017e87a2b56a75fc8" is the account of node 2*

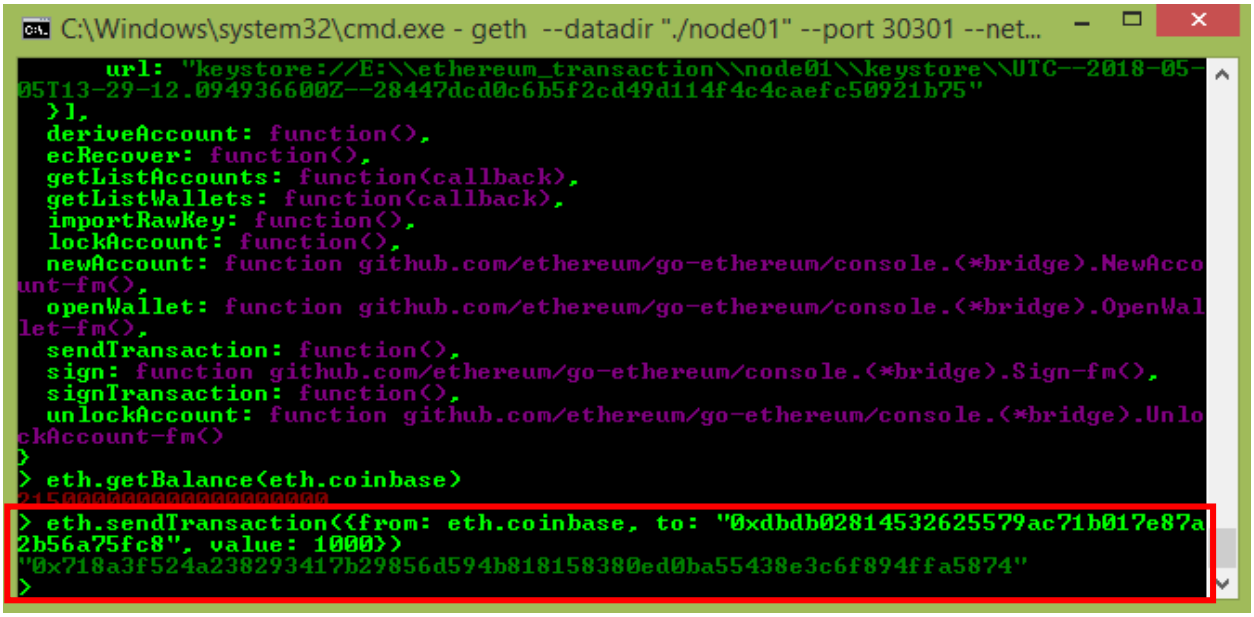

Figure 21. Send transaction at node 1

However, the transaction is not written in the ledger, you can check by typing "eth.pendingTransactions" as shown in Figure 22.

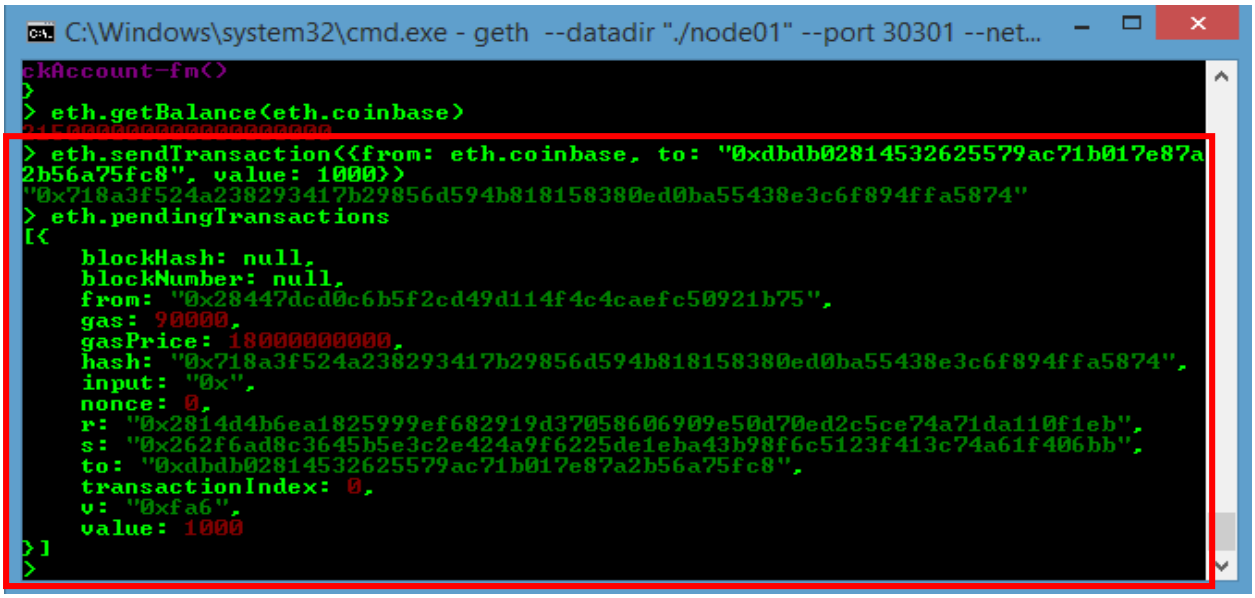

Figure 22. Check pending transaction at node 1

To accomplish the transmitting 1000 ethers from account of node 1 to account of node 2, we will use account of node 3 to mine by typing “miner.start(1)” like Figure 23.

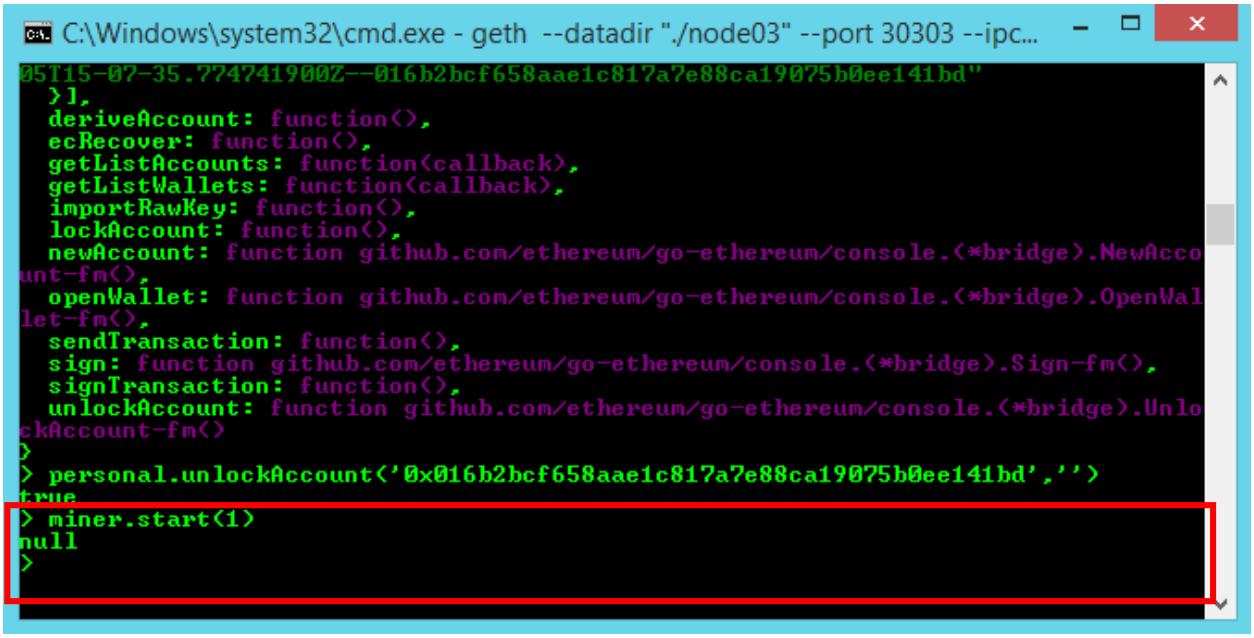

Figure 23. Miner at account of node 3

Now, to verify that the transfer transaction and write to the ledger successfully, we type "eth.pendingTransactions" again like Figure 24. We can see that the "pending transactions" is null. So it indicates the transfer transaction and write to the ledger successfully.

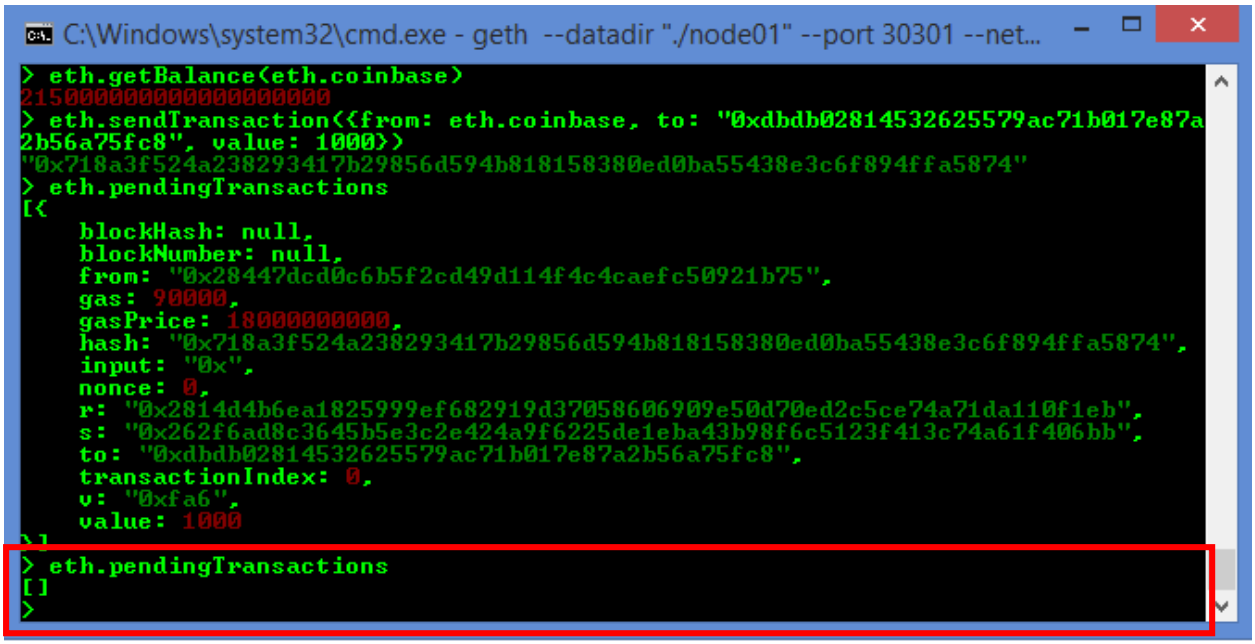

Figure 24. Accomplish sending transaction at node 1

Finally, we check that account of node 2 has received 1000 ethers by typing "eth.getBalance (eth.coinbase)" as shown in Figure 25.

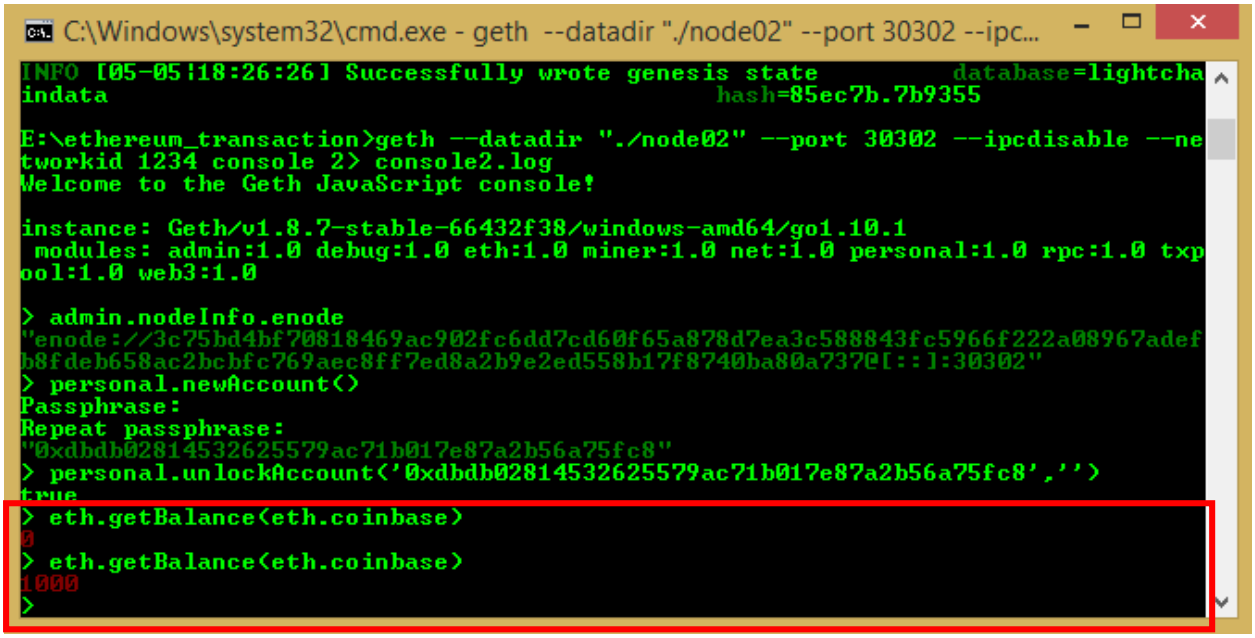

Figure 25. Check ether at account of node 2

Conclusion

I this tutorial, we have successfully completed the transaction and miner by 3 nodes on Ethereal Network.

Reference

https://claudiodangelis.com/ethereum/2018/02/19/exploring-ethereum-platform-accounts.html