

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**.

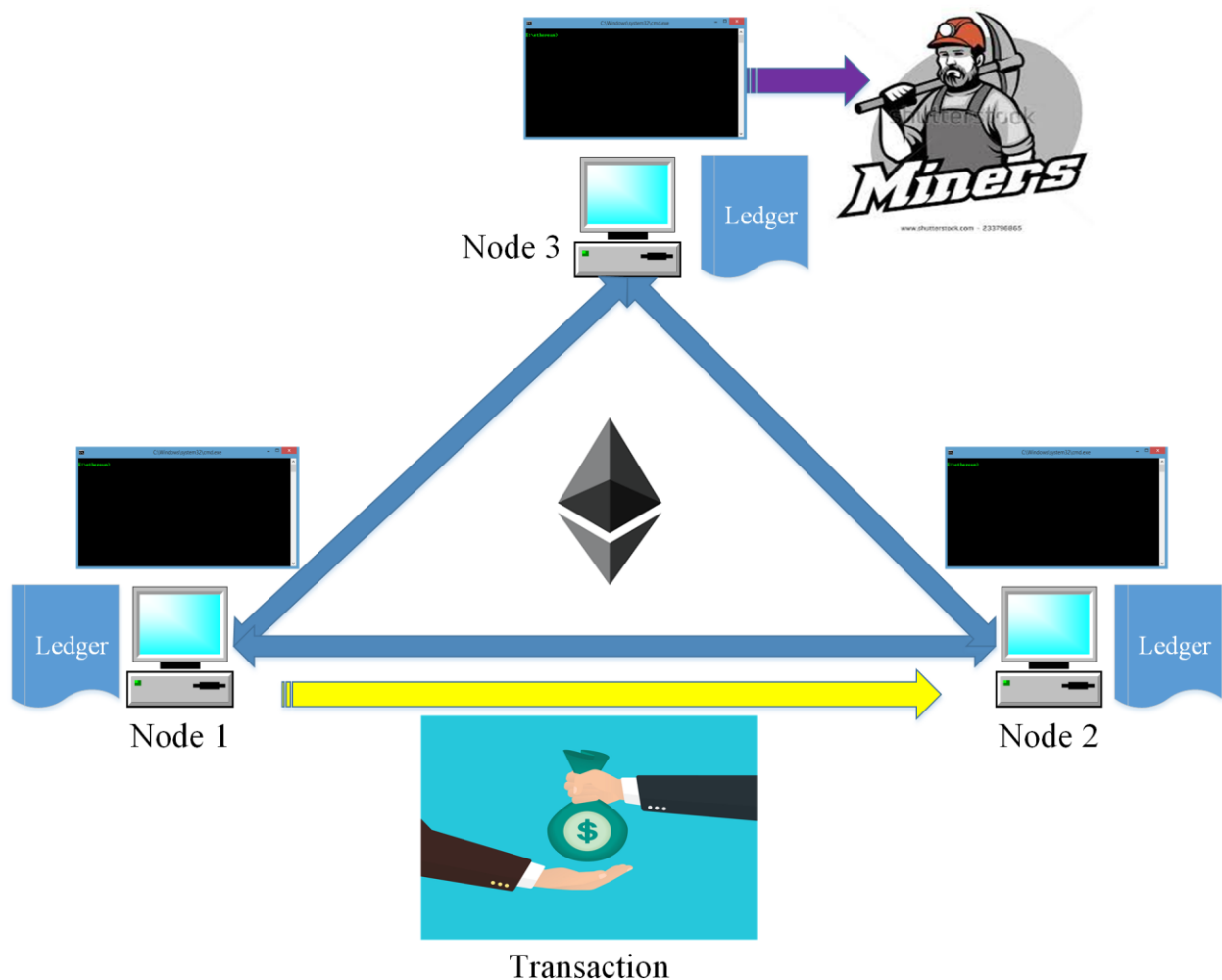


Figure 1. A scenario of sending transaction by Geth client

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_transaction” folder as shown in Figure 2.

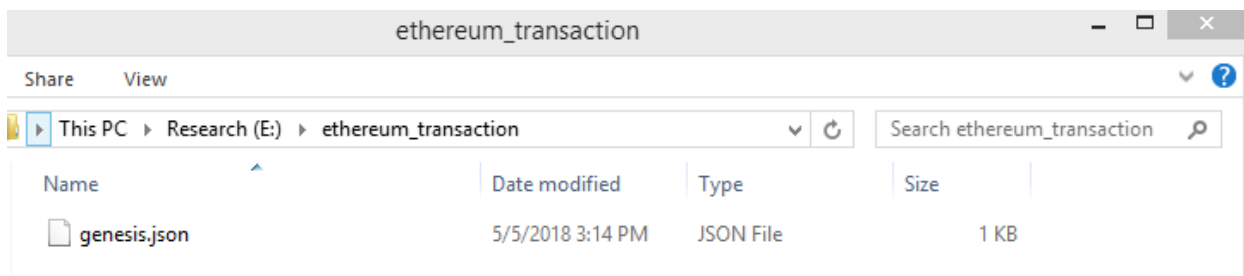


Figure 2. Genesis file

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

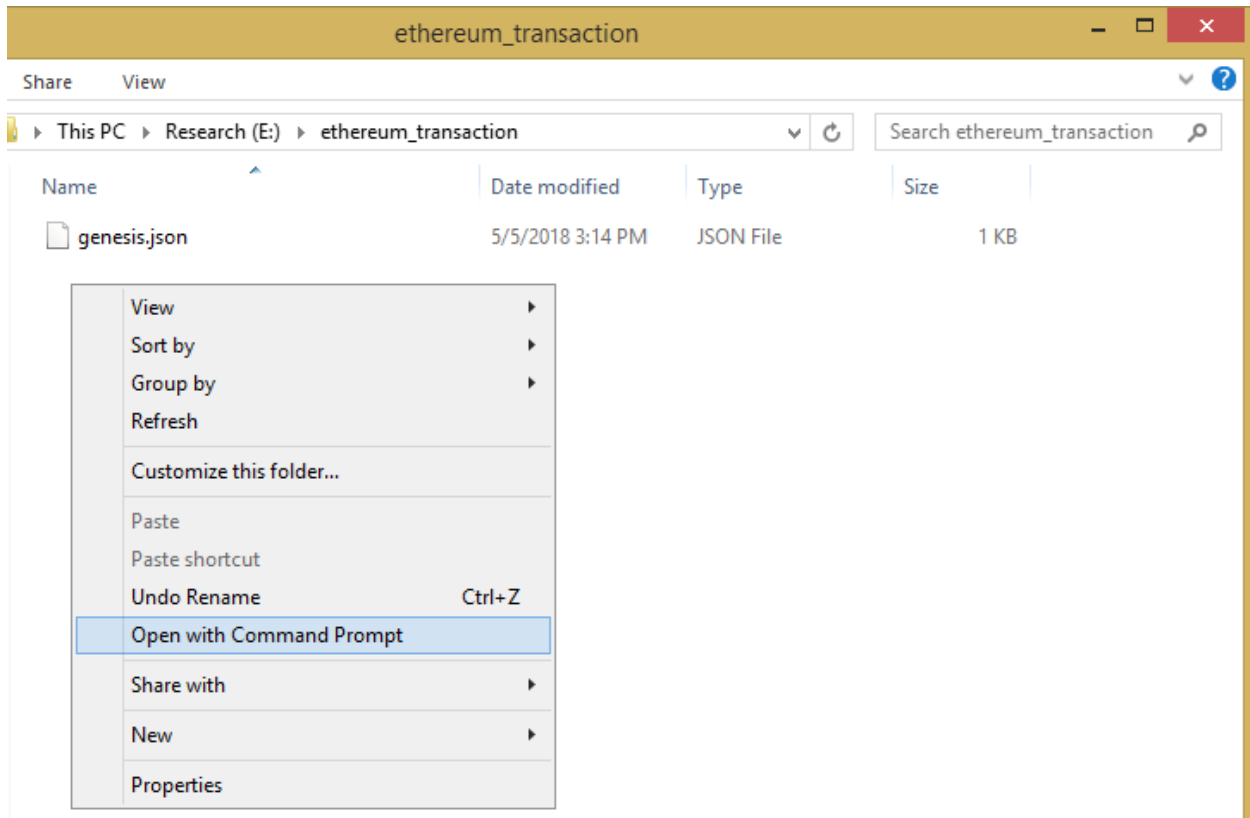


Figure 3. Open with Command Prompt

Then the terminal window will appear as Figure 4.

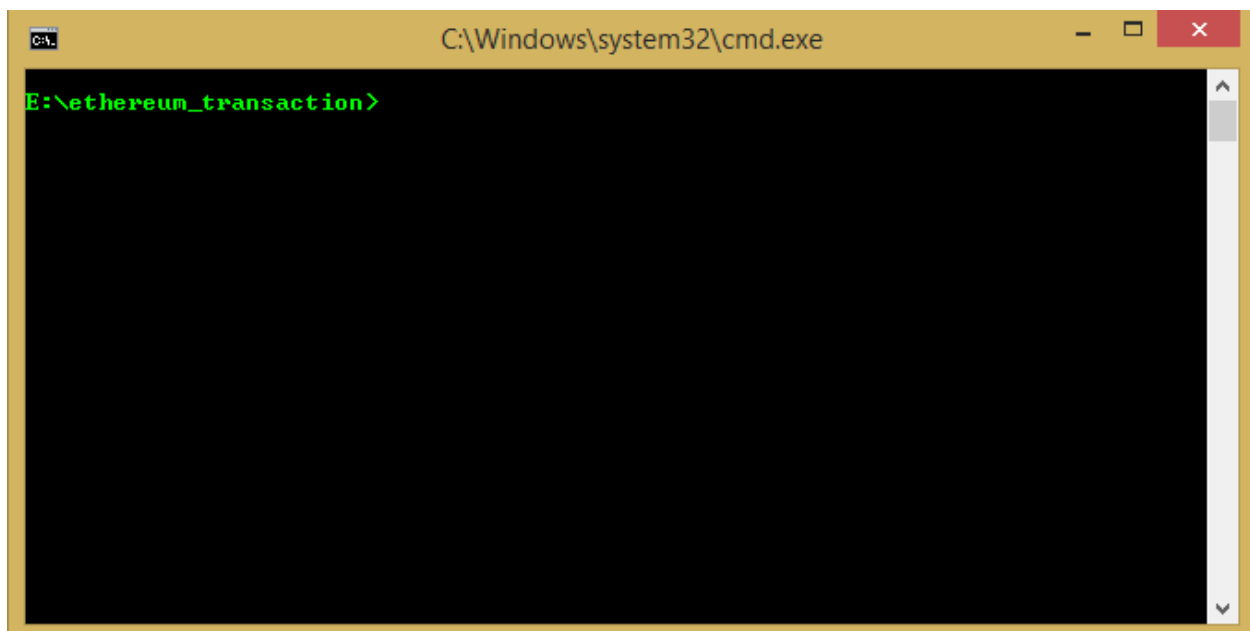
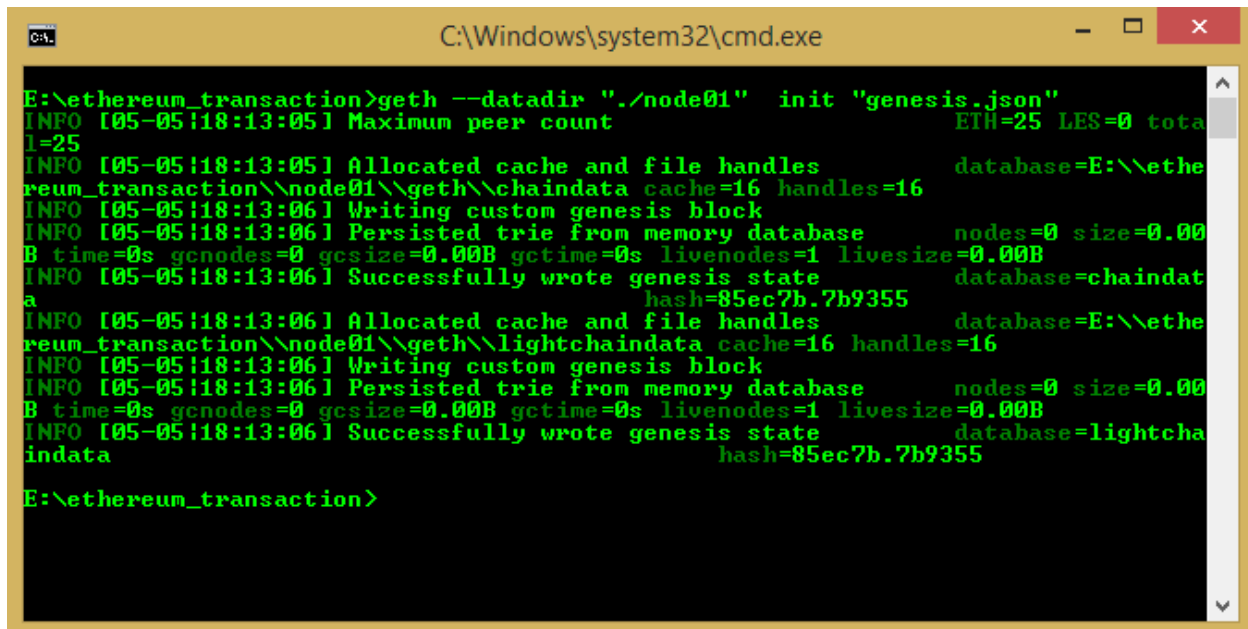


Figure 4. Command Prompt terminal

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



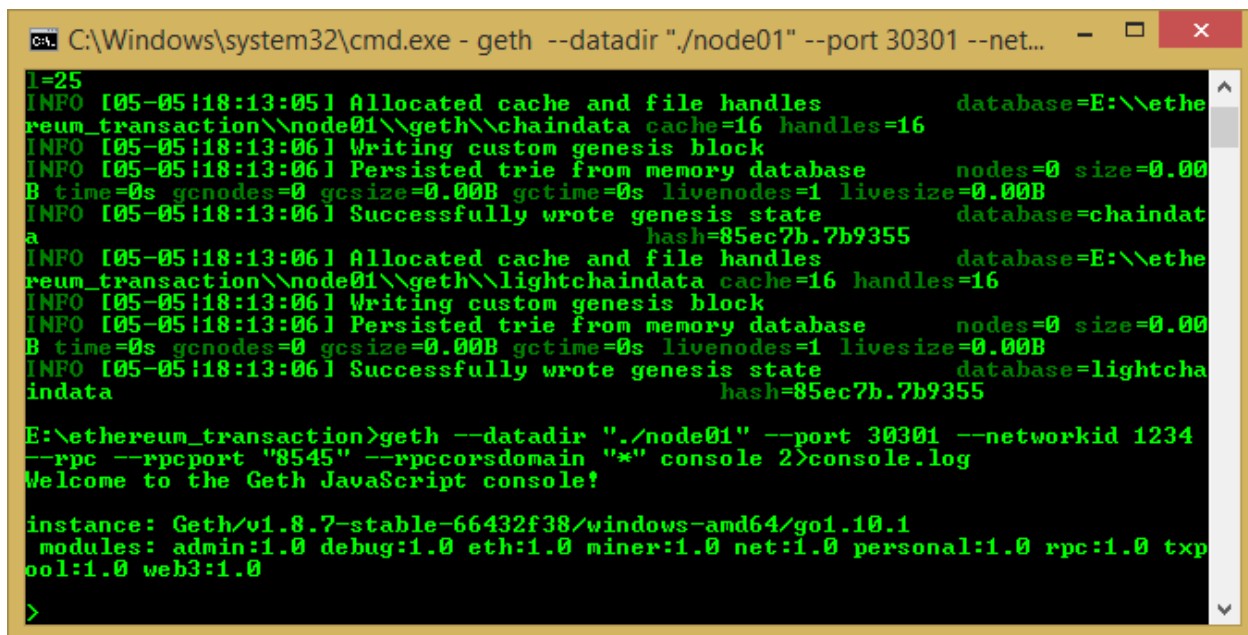
```
C:\Windows\system32\cmd.exe

E:\ethereum_transaction>geth --datadir "./node01" init "genesis.json"
INFO [05-05:18:13:05] Maximum peer count ETH=25 LES=0 total=25
INFO [05-05:18:13:05] Allocated cache and file handles database=E:\ethereum_transaction\node01\geth\chaindata cache=16 handles=16
INFO [05-05:18:13:06] Writing custom genesis block
INFO [05-05:18:13:06] Persisted trie from memory database nodes=0 size=0.00B time=0s gcnodes=0 gcsizes=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [05-05:18:13:06] Successfully wrote genesis state database=chaindata hash=85ec7b.7b9355
INFO [05-05:18:13:06] Allocated cache and file handles database=E:\ethereum_transaction\node01\geth\lightchaindata cache=16 handles=16
INFO [05-05:18:13:06] Writing custom genesis block
INFO [05-05:18:13:06] Persisted trie from memory database nodes=0 size=0.00B time=0s gcnodes=0 gcsizes=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [05-05:18:13:06] Successfully wrote genesis state database=lightchaindata hash=85ec7b.7b9355

E:\ethereum_transaction>
```

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.



```
C:\Windows\system32\cmd.exe - geth --datadir "./node01" --port 30301 --net...

E:\ethereum_transaction>geth --datadir "./node01" --port 30301 --networkid 1234 --rpc --rpcport "8545" --rpccorsdomain "*" console 2>console.log
Welcome to the Geth JavaScript console!

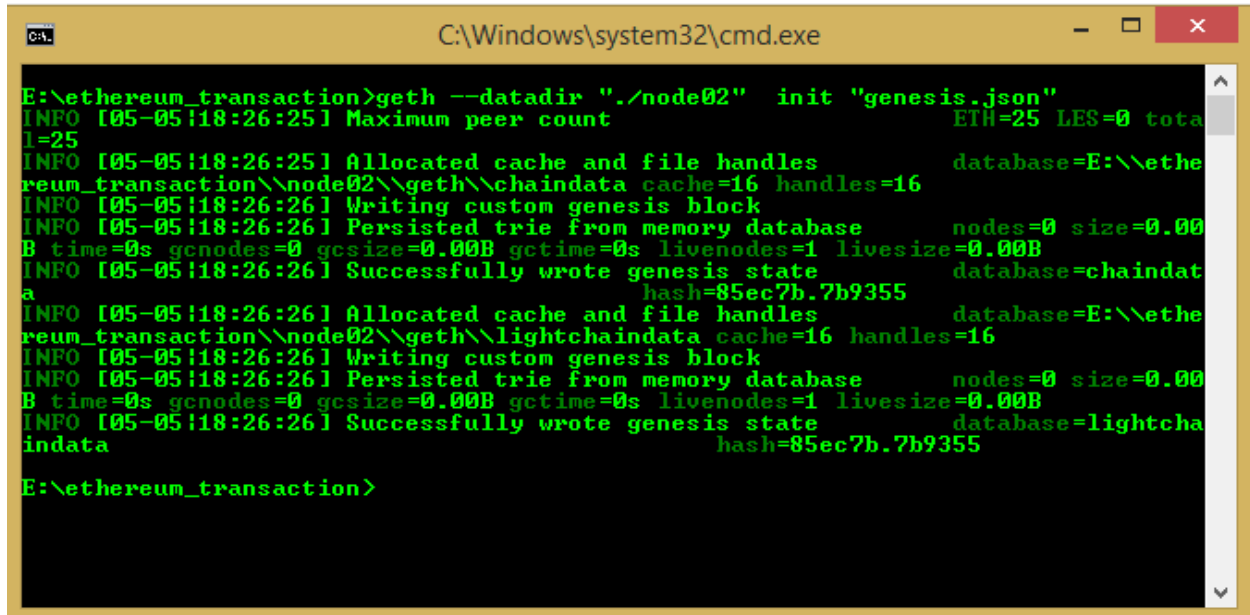
instance: Geth/v1.8.7-stable-66432f38/windows-amd64/go1.10.1
modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

>
```

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.



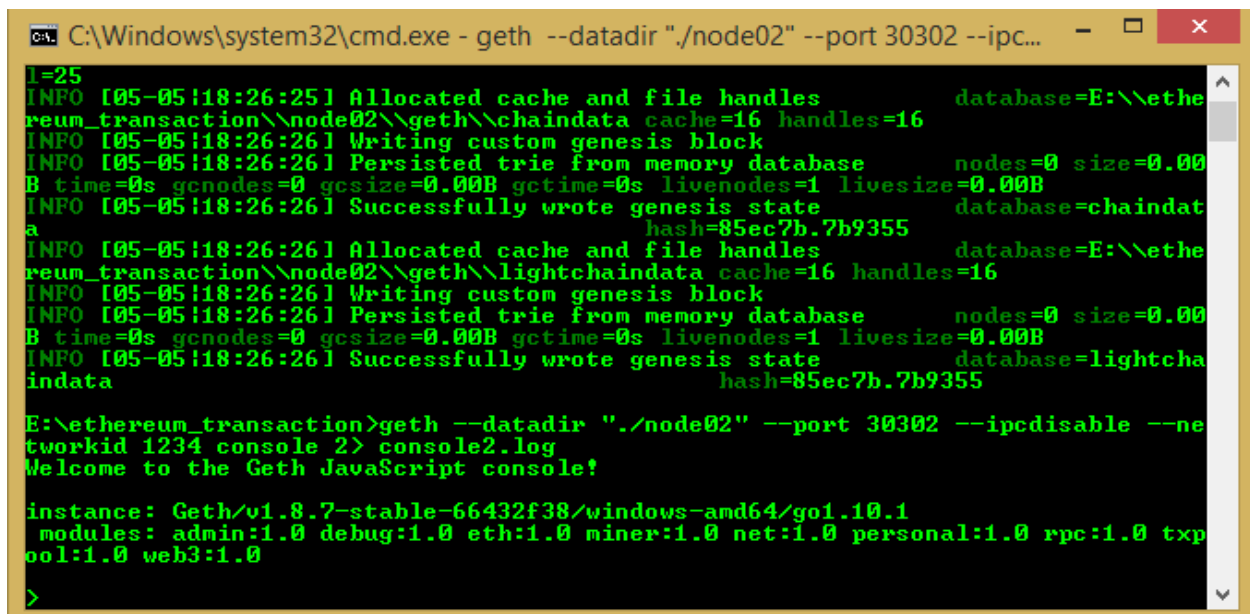
```
C:\Windows\system32\cmd.exe

E:\ethereum_transaction>geth --datadir "./node02" init "genesis.json"
INFO [05-05:18:26:25] Maximum peer count ETH=25 LES=0 total=25
INFO [05-05:18:26:25] Allocated cache and file handles database=E:\ethereum_transaction\node02\geth\chaindata cache=16 handles=16
INFO [05-05:18:26:26] Writing custom genesis block
INFO [05-05:18:26:26] Persisted trie from memory database nodes=0 size=0.00B time=0s gcnodes=0 gcspace=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [05-05:18:26:26] Successfully wrote genesis state database=chaindata hash=85ec7b.7b9355
INFO [05-05:18:26:26] Allocated cache and file handles database=E:\ethereum_transaction\node02\geth\lightchaindata cache=16 handles=16
INFO [05-05:18:26:26] Writing custom genesis block
INFO [05-05:18:26:26] Persisted trie from memory database nodes=0 size=0.00B time=0s gcnodes=0 gcspace=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [05-05:18:26:26] Successfully wrote genesis state database=lightchaindata hash=85ec7b.7b9355

E:\ethereum_transaction>
```

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.



```
C:\Windows\system32\cmd.exe - geth --datadir "./node02" --port 30302 --ipc...

E:\ethereum_transaction>geth --datadir "./node02" --port 30302 --ipcdisable --networkid 1234 console 2> console2.log
Welcome to the Geth JavaScript console!

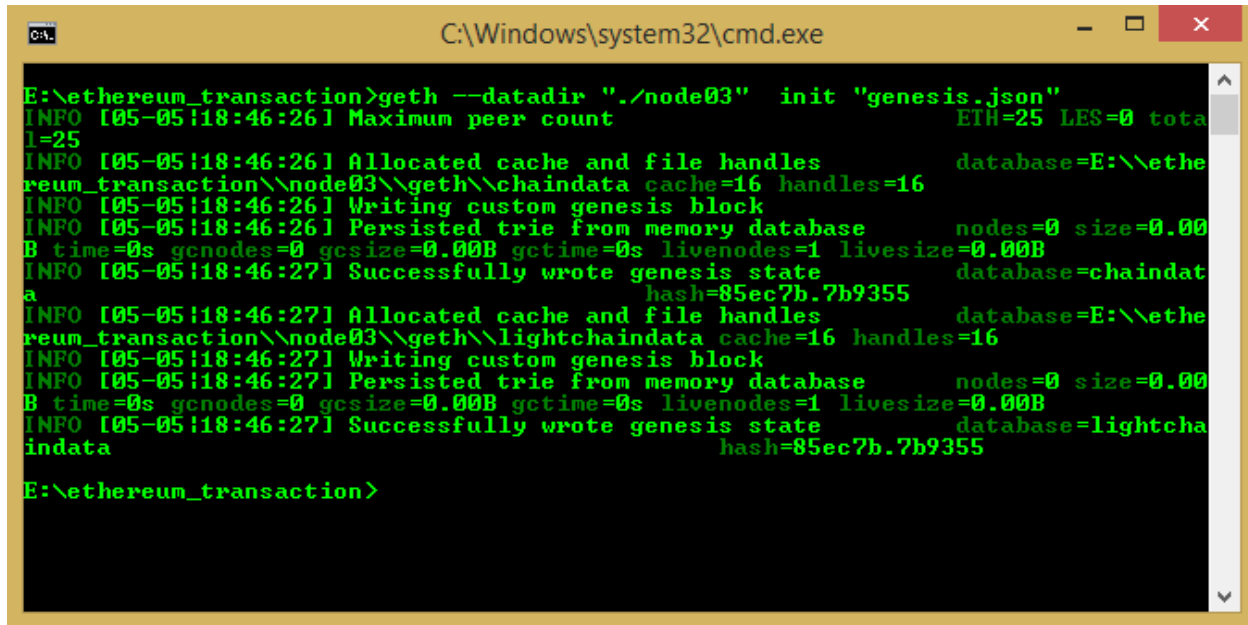
instance: Geth/v1.8.7-stable-66432f38/windows-amd64/go1.10.1
modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

>
```

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.



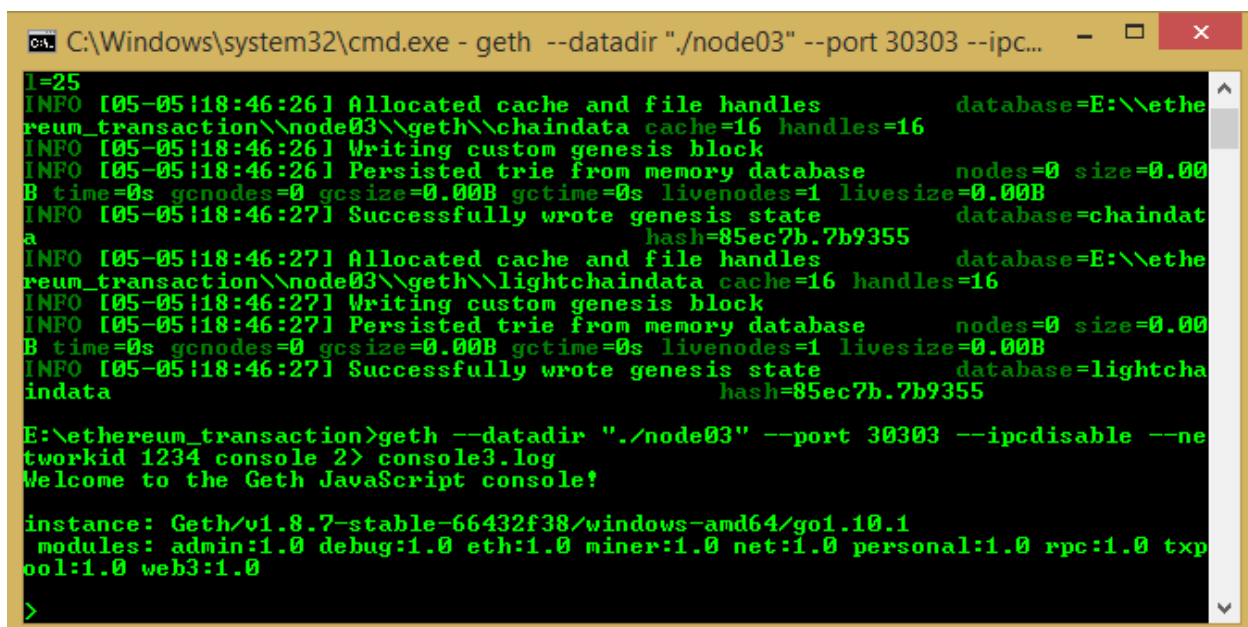
```
C:\Windows\system32\cmd.exe

E:\ethereum_transaction>geth --datadir "/node03" init "genesis.json"
INFO [05-05:18:46:26] Maximum peer count ETH=25 LES=0 total=25
INFO [05-05:18:46:26] Allocated cache and file handles database=E:\ethereum_transaction\node03\geth\chaindata cache=16 handles=16
INFO [05-05:18:46:26] Writing custom genesis block
INFO [05-05:18:46:26] Persisted trie from memory database nodes=0 size=0.00B time=0s gcnodes=0 gcspace=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [05-05:18:46:27] Successfully wrote genesis state database=chaindata hash=85ec7b.7b9355
INFO [05-05:18:46:27] Allocated cache and file handles database=E:\ethereum_transaction\node03\geth\lightchaindata cache=16 handles=16
INFO [05-05:18:46:27] Writing custom genesis block
INFO [05-05:18:46:27] Persisted trie from memory database nodes=0 size=0.00B time=0s gcnodes=0 gcspace=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [05-05:18:46:27] Successfully wrote genesis state database=lightchaindata hash=85ec7b.7b9355

E:\ethereum_transaction>
```

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.



```
C:\Windows\system32\cmd.exe - geth --datadir "/node03" --port 30303 --ipc...

1=25
INFO [05-05:18:46:26] Allocated cache and file handles database=E:\ethereum_transaction\node03\geth\chaindata cache=16 handles=16
INFO [05-05:18:46:26] Writing custom genesis block
INFO [05-05:18:46:26] Persisted trie from memory database nodes=0 size=0.00B time=0s gcnodes=0 gcspace=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [05-05:18:46:27] Successfully wrote genesis state database=chaindata hash=85ec7b.7b9355
INFO [05-05:18:46:27] Allocated cache and file handles database=E:\ethereum_transaction\node03\geth\lightchaindata cache=16 handles=16
INFO [05-05:18:46:27] Writing custom genesis block
INFO [05-05:18:46:27] Persisted trie from memory database nodes=0 size=0.00B time=0s gcnodes=0 gcspace=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [05-05:18:46:27] Successfully wrote genesis state database=lightchaindata hash=85ec7b.7b9355

E:\ethereum_transaction>geth --datadir "/node03" --port 30303 --ipcdisable --networkid 1234 console 2> console3.log
Welcome to the Geth JavaScript console!

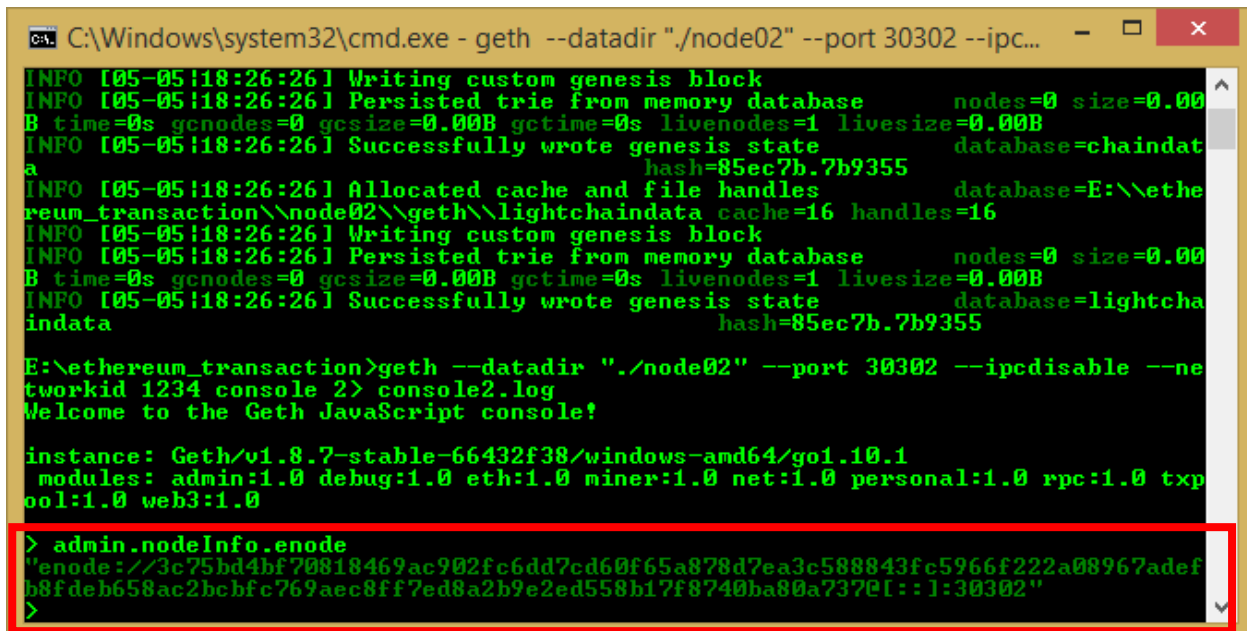
instance: Geth/v1.8.7-stable-66432f38/windows-amd64/go1.10.1
modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

>
```

Figure 10. Open Geth console at node 3

Linking 3 nodes

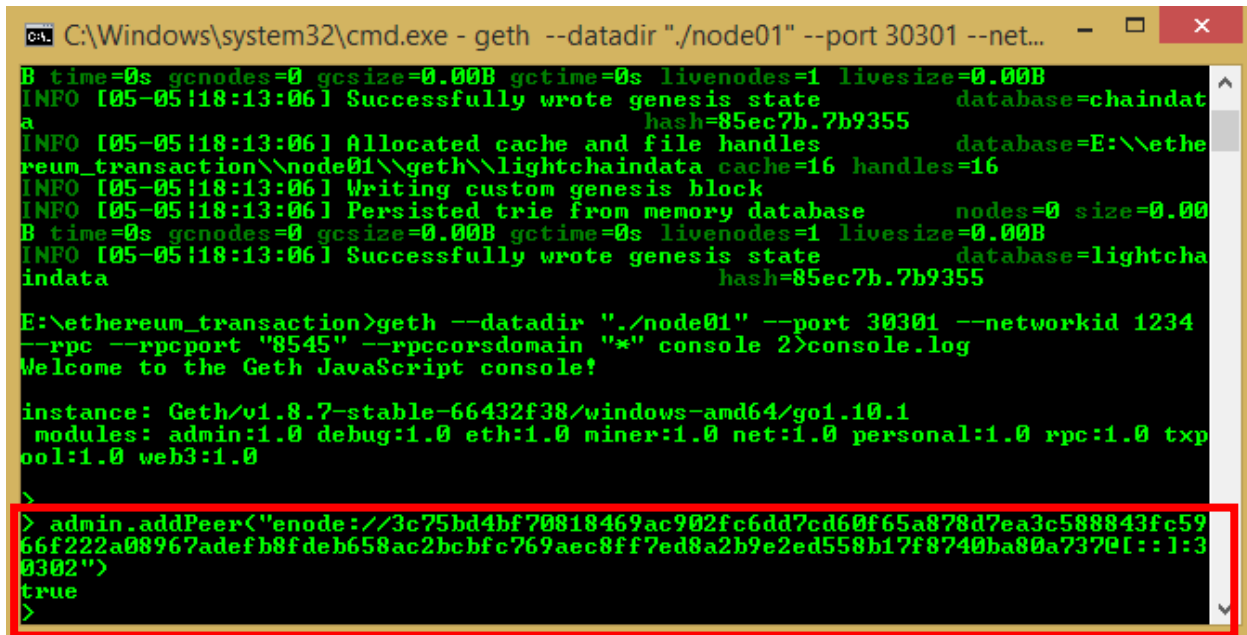
At **node 2**, we type “admin.nodeInfo.enode” like Figure 11.



```
C:\Windows\system32\cmd.exe - geth --datadir "./node02" --port 30302 --ipc...  
INFO [05-05:18:26:26] Writing custom genesis block  
INFO [05-05:18:26:26] Persisted trie from memory database      nodes=0 size=0.00B  
B time=0s gcnodes=0 gcsiz=0.00B gctime=0s livenodes=1 livesize=0.00B  
INFO [05-05:18:26:26] Successfully wrote genesis state      database=chaindata  
a hash=85ec7b.7b9355  
INFO [05-05:18:26:26] Allocated cache and file handles      database=E:\ether  
reum_transaction\node02\geth\lightchaindata cache=16 handles=16  
INFO [05-05:18:26:26] Writing custom genesis block  
INFO [05-05:18:26:26] Persisted trie from memory database      nodes=0 size=0.00B  
B time=0s gcnodes=0 gcsiz=0.00B gctime=0s livenodes=1 livesize=0.00B  
INFO [05-05:18:26:26] Successfully wrote genesis state      database=lightcha  
indata hash=85ec7b.7b9355  
  
E:\ethereum_transaction>geth --datadir "./node02" --port 30302 --ipcdisable --ne  
tworkid 1234 console 2> console2.log  
Welcome to the Geth JavaScript console!  
  
instance: Geth/v1.8.7-stable-66432f38/windows-amd64/go1.10.1  
modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txp  
ool:1.0 web3:1.0  
  
> admin.nodeInfo.enode  
"enode://3c75bd4bf70818469ac902fc6dd7cd60f65a878d7ea3c588843fc5966f222a08967ade  
fb8fdeb658ac2bcbfc769aec8ff7ed8a2b9e2ed558b17f8740ba80a7370e1::1:30302"  
>
```

Figure 11. Get enode address at node 2

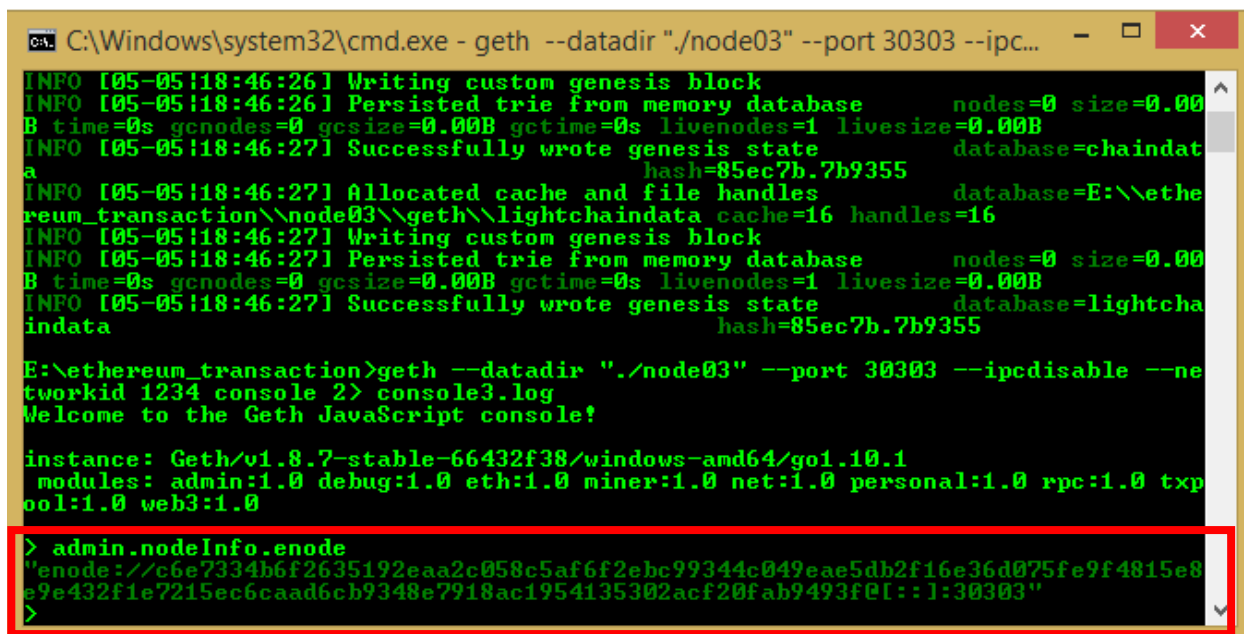
At **node 1**, we type “admin.addPeer(enode address **node 2**)” like Figure 12.



```
C:\Windows\system32\cmd.exe - geth --datadir "./node01" --port 30301 --net...  
B time=0s gcnodes=0 gcsiz=0.00B gctime=0s livenodes=1 livesize=0.00B  
INFO [05-05:18:13:06] Successfully wrote genesis state      database=chaindata  
a hash=85ec7b.7b9355  
INFO [05-05:18:13:06] Allocated cache and file handles      database=E:\ether  
reum_transaction\node01\geth\lightchaindata cache=16 handles=16  
INFO [05-05:18:13:06] Writing custom genesis block  
INFO [05-05:18:13:06] Persisted trie from memory database      nodes=0 size=0.00B  
B time=0s gcnodes=0 gcsiz=0.00B gctime=0s livenodes=1 livesize=0.00B  
INFO [05-05:18:13:06] Successfully wrote genesis state      database=lightcha  
indata hash=85ec7b.7b9355  
  
E:\ethereum_transaction>geth --datadir "./node01" --port 30301 --networkid 1234  
--rpc --rpcport "8545" --rpccorsdomain "*" console 2>console.log  
Welcome to the Geth JavaScript console!  
  
instance: Geth/v1.8.7-stable-66432f38/windows-amd64/go1.10.1  
modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txp  
ool:1.0 web3:1.0  
  
>  
> admin.addPeer("enode://3c75bd4bf70818469ac902fc6dd7cd60f65a878d7ea3c588843fc59  
66f222a08967adeb8fdeb658ac2bcbfc769aec8ff7ed8a2b9e2ed558b17f8740ba80a7370e1::1:3  
0302")  
true  
>
```

Figure 12. Add node 2 at node 1

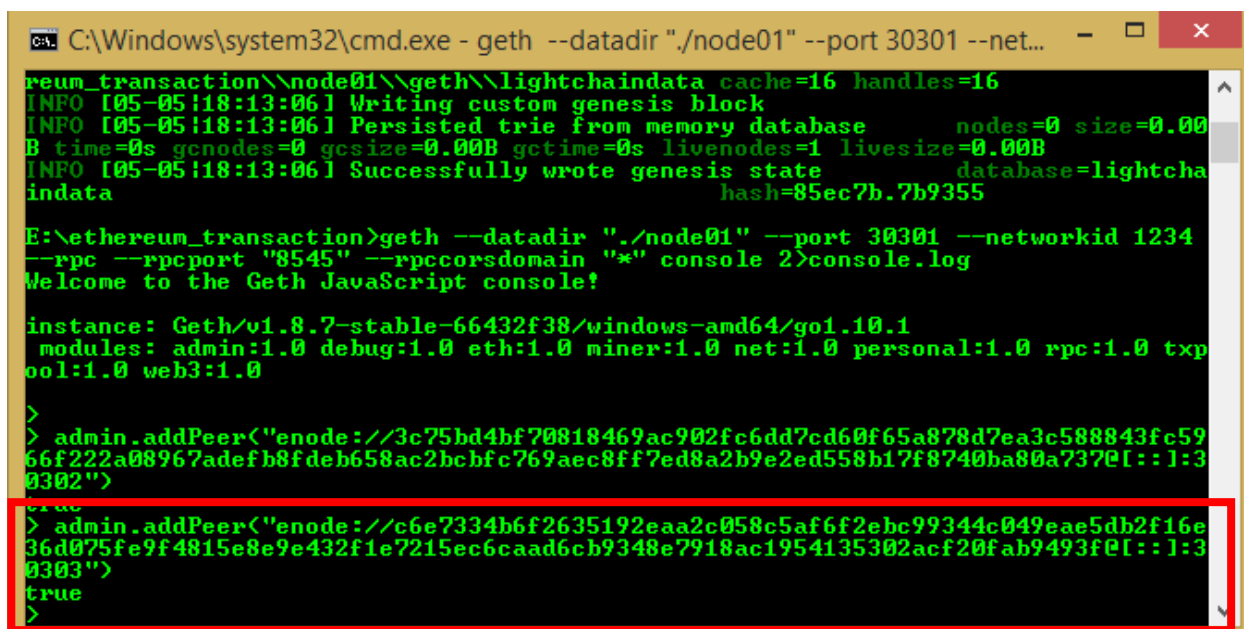
At **node 3**, we type “admin.nodeInfo.enode” like Figure 13.



```
C:\Windows\system32\cmd.exe - geth --datadir "./node03" --port 30303 --ipc...  
INFO [05-05:18:46:26] Writing custom genesis block  
INFO [05-05:18:46:26] Persisted trie from memory database      nodes=0 size=0.00B  
B time=0s gcnodes=0 gcsz=0.00B gctime=0s livenodes=1 livesize=0.00B  
INFO [05-05:18:46:27] Successfully wrote genesis state      database=chaindata  
INFO [05-05:18:46:27] Allocated cache and file handles      database=E:\ether  
reum_transaction\node03\geth\lightchaindata cache=16 handles=16  
INFO [05-05:18:46:27] Writing custom genesis block  
INFO [05-05:18:46:27] Persisted trie from memory database      nodes=0 size=0.00B  
B time=0s gcnodes=0 gcsz=0.00B gctime=0s livenodes=1 livesize=0.00B  
INFO [05-05:18:46:27] Successfully wrote genesis state      database=lightcha  
indata  
hash=85ec7b.7b9355  
E:\ethereum_transaction>geth --datadir "./node03" --port 30303 --ipcdisable --ne  
tworkid 1234 console 2> console3.log  
Welcome to the Geth JavaScript console!  
  
instance: Geth/v1.8.7-stable-66432f38/windows-amd64/go1.10.1  
modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txp  
ool:1.0 web3:1.0  
  
> admin.nodeInfo.enode  
"enode://c6e7334b6f2635192eaa2c058c5af6f2ebc99344c049eae5db2f16e36d075fe9f4815e8  
e9e432f1e7215ec6caad6cb9348e7918ac1954135302acf20fab9493f0e1::1:30303"  
>
```

Figure 13. Get enode address at node 2

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



```
C:\Windows\system32\cmd.exe - geth --datadir "./node01" --port 30301 --net...  
reum_transaction\node01\geth\lightchaindata cache=16 handles=16  
INFO [05-05:18:13:06] Writing custom genesis block  
INFO [05-05:18:13:06] Persisted trie from memory database      nodes=0 size=0.00B  
B time=0s gcnodes=0 gcsz=0.00B gctime=0s livenodes=1 livesize=0.00B  
INFO [05-05:18:13:06] Successfully wrote genesis state      database=lightcha  
indata  
hash=85ec7b.7b9355  
E:\ethereum_transaction>geth --datadir "./node01" --port 30301 --networkid 1234  
--rpc --rpcport "8545" --rpccorsdomain "*" console 2>console.log  
Welcome to the Geth JavaScript console!  
  
instance: Geth/v1.8.7-stable-66432f38/windows-amd64/go1.10.1  
modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txp  
ool:1.0 web3:1.0  
  
>  
> admin.addPeer("enode://3c75bd4bf70818469ac902fc6dd7cd60f65a878d7ea3c588843fc59  
66f222a08967adefb8fdeb658ac2bcbfc769aec8ff7ed8a2b9e2ed558b17f8740ba80a7370e1::1:3  
0302")  
>  
> admin.addPeer("enode://c6e7334b6f2635192eaa2c058c5af6f2ebc99344c049eae5db2f16e  
36d075fe9f4815e8e9e432f1e7215ec6caad6cb9348e7918ac1954135302acf20fab9493f0e1::1:3  
0303")  
true  
>
```

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.

```
C:\Windows\system32\cmd.exe - geth --datadir \"./node01\" --port 30301 --net...
>
> admin.addPeer(<\"enode://3c75bd4bf70818469ac902fc6dd7cd60f65a878d7ea3c588843fc5966f222a08967adefb8fdeb658ac2bcbfc769aec8ff7ed8a2b9e2ed558b17f8740ba80a7370[:1:30302]\">
true
> admin.addPeer(<\"enode://c6e7334b6f2635192eaa2c058c5af6f2ebc99344c049eae5db2f16e36d075fe9f4815e8e9e432f1e7215ec6caad6cb9348e7918ac1954135302acf20fab9493f[:1:30303]\">
true
> net.peerCount
2
> admin.peers
[<
  {
    caps: [\"eth/63\"],
    id: <\"3c75bd4bf70818469ac902fc6dd7cd60f65a878d7ea3c588843fc5966f222a08967adefb8fdeb658ac2bcbfc769aec8ff7ed8a2b9e2ed558b17f8740ba80a737\">,
    name: <\"Geth/v1.8.7-stable-b6432f38/windows-amd64/go1.10.1\">,
    network: {
      inbound: false,
      localAddress: <\"127.0.0.1:50322\">,
      remoteAddress: <\"127.0.0.1:30302\">,
      static: true,
      trusted: false
    },
    protocols: {
      eth: {
        difficulty: 4000,
        head: <\"0x85ec7bc4c32d326f216d6aa8857df1efdaa4f62084e4acae882fb99fd7b9355\">,
        version: 63
      }
    }
  },
  {
    caps: [\"eth/63\"],
    id: <\"c6e7334b6f2635192eaa2c058c5af6f2ebc99344c049eae5db2f16e36d075fe9f4815e8e9e432f1e7215ec6caad6cb9348e7918ac1954135302acf20fab9493f\">,
    name: <\"Geth/v1.8.7-stable-b6432f38/windows-amd64/go1.10.1\">,
    network: {
      inbound: false,
      localAddress: <\"127.0.0.1:52921\">,
      remoteAddress: <\"127.0.0.1:30303\">,
      static: true,
      trusted: false
    },
    protocols: {
      eth: {
        difficulty: 4000,
        head: <\"0x85ec7bc4c32d326f216d6aa8857df1efdaa4f62084e4acae882fb99fd7b9355\">,
        version: 63
      }
    }
  }
]
>
```

Node 1 has 2 peers

Enode address node 2

Enode address node 3

Figure 15. Check peer-to-peer network

Create account

```
C:\Windows\system32\cmd.exe - geth --datadir "./node01" --port 30301 --net...  
caps: ["eth/63"],  
id: "c6e7334b6f2635192eaa2c058c5af6f2ebc99344c049eae5db2f16e36d075fe9f4815e8  
e9e432f1e7215ec6caad6cb9348e7918ac1954135302acf20fab9493f",  
name: "Geth/v1.8.7-stable-66432f38/windows-amd64/go1.10.1",  
network: {  
  inbound: false,  
  localAddress: "127.0.0.1:52921",  
  remoteAddress: "127.0.0.1:30303",  
  static: true,  
  trusted: false  
},  
protocols: {  
  eth: {  
    difficulty: 4000,  
    head: "0x85ec7bc4c32d326f216d6aa8857df1efd4a4f62084e4acae882fb99fd7b935  
5",  
    version: 63  
  }  
}  
}>  
> personal.newAccount(<)>  
Passphrase:  
Repeat passphrase:  
"0x28447dcd0c6b5f2cd49d114f4c4caefc50921b75"  
>
```

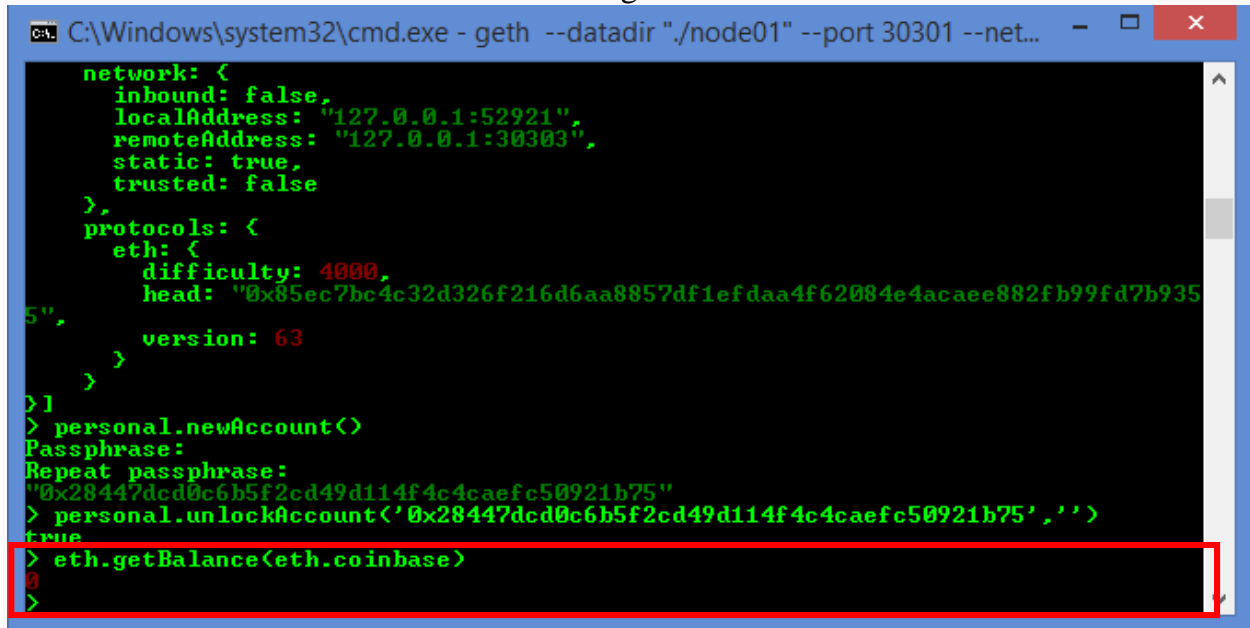
Then we have to unlock account to transmit transaction or mine by typing “*personal.unlockAccount(‘account’,’)*” as shown in Figure 17, just like login your account into facebook to post status or chat.

```
C:\Windows\system32\cmd.exe - geth --datadir "./node01" --port 30301 --net...
e9e432f1e7215ec6caad6cb9348e7918ac1954135302acf20fab9493f"
name: "Geth/v1.8.7-stable-66432f38/windows-amd64/go1.10.1",
network: {
  inbound: false,
  localAddress: "127.0.0.1:52921",
  remoteAddress: "127.0.0.1:30303",
  static: true,
  trusted: false
},
protocols: {
  eth: {
    difficulty: 4000,
    head: "0x85ec7bc4c32d326f216d6aa8857df1efdaa4f62084e4acaee882fb99fd7b935
5",
    version: 63
  }
}
}
]
> personal.newAccount(<)
Passphrase:
Repeat passphrase:
"0x28447dcd0c6b5f2cd49d114f4c4caefc50921b75"
> personal.unlockAccount('0x28447dcd0c6b5f2cd49d114f4c4caefc50921b75', '')
true
>
```

Figure 17. Unlock account at node 1

Send transaction

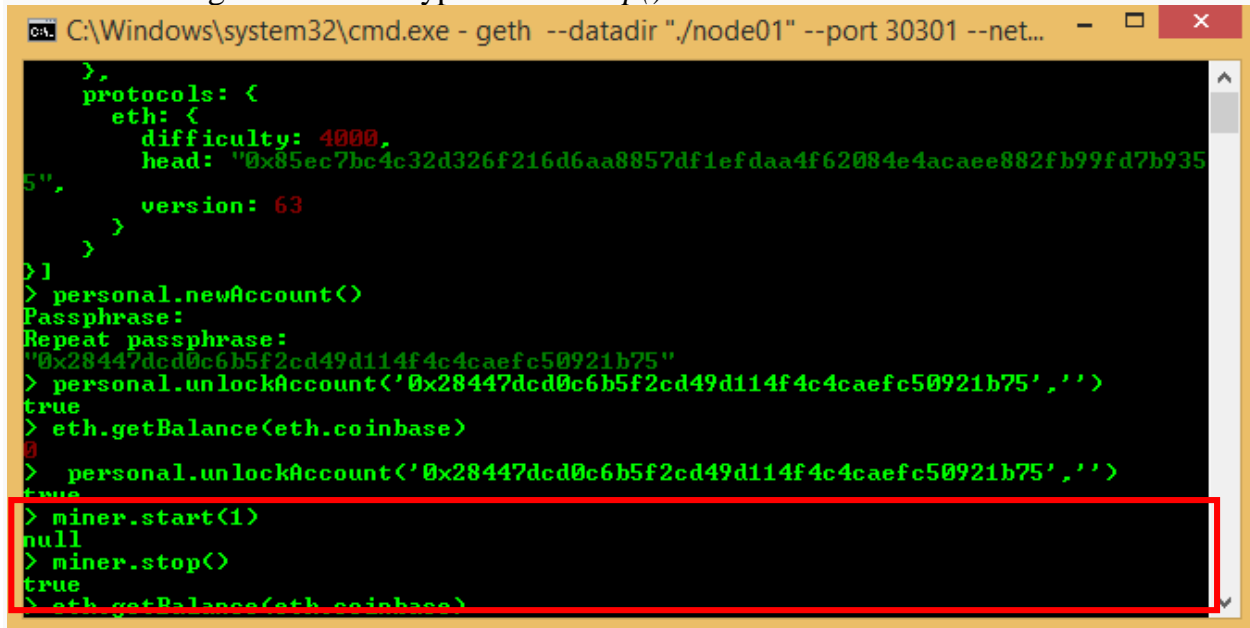
Accounts created in the previous steps do not have any ether, meaning that the balance is 0. How do we know that? At **node 1**, we type “*eth.getBalance(eth.coinbase)*” to show the balance of node 1 as shown in Figure 18.



```
C:\Windows\system32\cmd.exe - geth --datadir "./node01" --port 30301 --net...  
  
network: {  
  inbound: false,  
  localAddress: "127.0.0.1:52921",  
  remoteAddress: "127.0.0.1:30303",  
  static: true,  
  trusted: false  
},  
protocols: {  
  eth: {  
    difficulty: 4000,  
    head: "0x85ec7bc4c32d326f216d6aa8857df1efdaa4f62084e4acaee882fb99fd7b935",  
    version: 63  
  }  
}  
}<br><br>>1  
> personal.newAccount()  
Passphrase:  
Repeat passphrase:  
"0x28447dcd0c6b5f2cd49d114f4c4caefc50921b75"  
> personal.unlockAccount('0x28447dcd0c6b5f2cd49d114f4c4caefc50921b75', '')  
true  
> eth.getBalance(eth.coinbase)  
0  
>
```

Figure 18. Unlock account at node 1

If your account doesn't have any ethers, you can'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()*”.



```
C:\Windows\system32\cmd.exe - geth --datadir "./node01" --port 30301 --net...  
  
network: {  
  inbound: false,  
  localAddress: "127.0.0.1:52921",  
  remoteAddress: "127.0.0.1:30303",  
  static: true,  
  trusted: false  
},  
protocols: {  
  eth: {  
    difficulty: 4000,  
    head: "0x85ec7bc4c32d326f216d6aa8857df1efdaa4f62084e4acaee882fb99fd7b935",  
    version: 63  
  }  
}  
}<br><br>>1  
> personal.newAccount()  
Passphrase:  
Repeat passphrase:  
"0x28447dcd0c6b5f2cd49d114f4c4caefc50921b75"  
> personal.unlockAccount('0x28447dcd0c6b5f2cd49d114f4c4caefc50921b75', '')  
true  
> eth.getBalance(eth.coinbase)  
0  
> personal.unlockAccount('0x28447dcd0c6b5f2cd49d114f4c4caefc50921b75', '')  
true  
> miner.start(1)  
null  
> miner.stop()  
true  
> eth.getBalance(eth.coinbase)  
0  
>
```

Figure 19. Get ether at node 1

```
C:\Windows\system32\cmd.exe - geth --datadir "./node01" --port 30301 --net...  
status: "Locked",  
url: "keystore://E:\\ethereum_transaction\\node01\\keystore\\UTC--2018-05-05T13-29-12.094936600Z--28447dcd0c6b5f2cd49d114f4c4caefc50921b75"  
>}.  
deriveAccount: function(),  
ecRecover: function(),  
getListAccounts: function(callback),  
getListWallets: function(callback),  
importRawKey: function(),  
lockAccount: function(),  
newAccount: function github.com/ethereum/go-ethereum/console.(*bridge).NewAccount-fm(),  
openWallet: function github.com/ethereum/go-ethereum/console.(*bridge).OpenWallet-fm(),  
sendTransaction: function(),  
sign: function github.com/ethereum/go-ethereum/console.(*bridge).Sign-fm(),  
signTransaction: function(),  
unlockAccount: function github.com/ethereum/go-ethereum/console.(*bridge).UnlockAccount-fm()  
>  
> miner.stop()  
true  
> eth.getBalance(eth.coinbase)  
2150000000000000000000  
>
```

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

```
C:\Windows\system32\cmd.exe - geth --datadir "./node02" --port 30302 --ipc... - [X]
INFO [05-05:18:26:26] Persisted trie from memory database      nodes=0 size=0.00B
B time=0s gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [05-05:18:26:26] Successfully wrote genesis state    database=lightcha
indata                                hash=85ec7b.7b9355

E:\ethereum_transaction>geth --datadir "./node02" --port 30302 --ipcdisable --ne
tworkid 1234 console 2> console2.log
Welcome to the Geth JavaScript console!

instance: Geth/v1.8.7-stable-66432f38/windows-amd64/go1.10.1
modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txp
ool:1.0 web3:1.0

> admin.nodeInfo.enode
"enode://3c75bd4bf70818469ac902fc6dd7cd60f65a878d7ea3c588843fc5966f222a08967adef
b8fdeb658ac2bcbfc769aec8ff7ed8a2b9e2ed558b17f8740ba80a7370[:]:1:30302"
> personal.newAccount(<
Passphrase:
Repeat passphrase:
"0xdbdb02814532625579ac71b017e87a2b56a75fc8"
> personal.unlockAccount(<'0xdbdb02814532625579ac71b017e87a2b56a75fc8', '>
true
> eth.getBalance(eth.coinbase)
0
>
```

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

```

C:\Windows\system32\cmd.exe - geth --datadir "./node03" --port 30303 --ipc...
05T15-07-35.774741900Z--016b2bcf658aae1c817a7e88ca19075b0ee141bd"
}l,
deriveAccount: function(),
ecRecover: function(),
getListAccounts: function(callback),
getListWallets: function(callback),
importRawKey: function(),
lockAccount: function(),
newAccount: function github.com/ethereum/go-ethereum/console.(*bridge).NewAcco
unt-fm(),
openWallet: function github.com/ethereum/go-ethereum/console.(*bridge).OpenWal
let-fm(),
sendTransaction: function(),
sign: function github.com/ethereum/go-ethereum/console.(*bridge).Sign-fm(),
signTransaction: function(),
unlockAccount: function github.com/ethereum/go-ethereum/console.(*bridge).Unlo
ckAccount-fm()
>
> personal.unlockAccount('0x016b2bcf658aae1c817a7e88ca19075b0ee141bd','')
true
> miner.start(1)
null
>

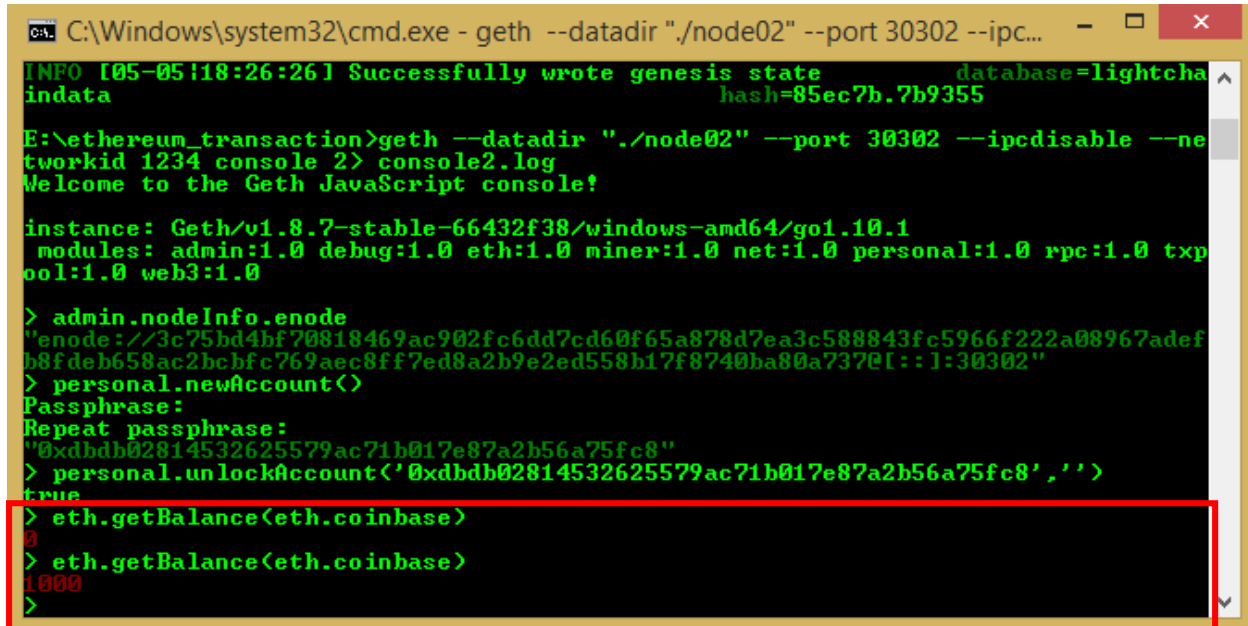
```

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.

```
C:\Windows\system32\cmd.exe - geth --datadir "./node01" --port 30301 --net...
> eth.getBalance(eth.coinbase)
2150000000000000000000000
> eth.sendTransaction(<from: eth.coinbase, to: "0xdbdb02814532625579ac71b017e87a
2b56a75fc8", value: 1000>)
"0x718a3f524a238293417b29856d594b818158380ed0ba55438e3c6f894ffa5874"
> eth.pendingTransactions
[[
  blockHash: null,
  blockNumber: null,
  from: "0x28447dcd0c6b5f2cd49d114f4c4caefc50921b75",
  gas: 90000,
  gasPrice: 18000000000,
  hash: "0x718a3f524a238293417b29856d594b818158380ed0ba55438e3c6f894ffa5874",
  input: "0x",
  nonce: 0,
  r: "0x2814d4b6ea1825999ef682919d37058606909e50d70ed2c5ce74a71da110f1eb",
  s: "0x262f6ad8c3645b5e3c2e424a9f6225de1eba43b98f6c5123f413c74a61f406bb",
  to: "0xdbdb02814532625579ac71b017e87a2b56a75fc8",
  transactionIndex: 0,
  v: "0xfa6",
  value: 1000
]]
> eth.pendingTransactions
[[
]]
```

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.

A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe - geth --datadir \"./node02\" --port 30302 --ipc...". The window displays the Geth JavaScript console interface. The output shows the successful writing of the genesis state, followed by the Geth version and module information. The user enters the command "admin.nodeInfo.enode" and receives a long hexadecimal string. Then, the user enters "personal.newAccount()" and is prompted for a passphrase. After entering the passphrase "0xdbdb02814532625579ac71b017e87a2b56a75fc8", the user enters "personal.unlockAccount('0xdbdb02814532625579ac71b017e87a2b56a75fc8', '')" and receives "true". Finally, the user enters "eth.getBalance(eth.coinbase)" and receives "0", then enters "eth.getBalance(eth.coinbase)" again and receives "1000". The last two commands and their outputs are highlighted with a red rectangle.

```
C:\Windows\system32\cmd.exe - geth --datadir \"./node02\" --port 30302 --ipc...
INFO [05-05!18:26:26] Successfully wrote genesis state      database=lightcha
indata                                          hash=85ec7b.7b9355

E:\ethereum_transaction>geth --datadir \"./node02\" --port 30302 --ipcdisable --ne
tworkid 1234 console 2> console2.log
Welcome to the Geth JavaScript console!

instance: Geth/v1.8.7-stable-66432f38/windows-amd64/go1.10.1
modules: admin:1.0 debug:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txp
ool:1.0 web3:1.0

> admin.nodeInfo.enode
'enode://3c75bd4bf70818469ac902fc6dd7cd60f65a878d7ea3c588843fc5966f222a08967adef
b8fdeb658ac2bcbfc769aec8ff7ed8a2b9e2ed558b17f8740ba80a7370[:]:30302'
> personal.newAccount()
Passphrase:
Repeat passphrase:
'0xdbdb02814532625579ac71b017e87a2b56a75fc8'
> personal.unlockAccount('0xdbdb02814532625579ac71b017e87a2b56a75fc8', '')
true
> eth.getBalance(eth.coinbase)
0
> eth.getBalance(eth.coinbase)
1000
>
```

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 Ethereum Network.

Reference

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