

# OpenLedger Bitshares VM Testnet

Pull the docker image from <https://hub.docker.com/r/aetsoft/evm-demo/>

```
docker pull aetsoft/evm-demo:1.0
```

Connect to the node.

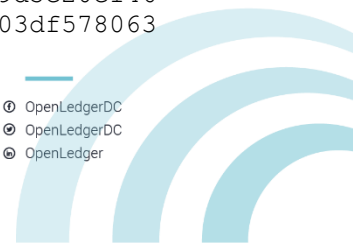
```
docker run --rm -it aetsoft/evm-demo:1.0
```

Inspect node's status, you should see the non-zero `head_block_number`, which means that you have successfully connected to the network. `b3.get_dynamic_global_properties()` might be rather slow to respond, typically this is the case for a freshly started node which is busy synchronizing blocks with the greater blockchain. There are more than 1500000 blocks at the time of writing, so syncing cannot be instant.

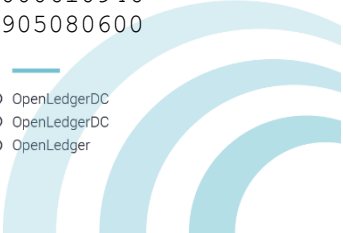
```
$ b3.get_dynamic_global_properties()
{ id: '2.1.0',
  head_block_number: 154354,
  head_block_id: '00025af25a3582e6f4528e47babe51f473fbf053',
  time: '2018-07-30T08:15:54',
  current_witness: '1.6.25',
  next_maintenance_time: '2018-07-30T08:16:00',
  last_budget_time: '2018-07-30T08:15:54',
  witness_budget: 34,
  accounts_registered_this_interval: 0,
  recently_missed_count: 4,
  current_aslot: 5606658,
  recent_slots_filled: '340282366604025803924690803153702486014',
  dynamic_flags: 1,
  last_irreversible_block_num: 154338 }
```

Let's create EOS contract.(EOS contract source code can be found at <https://etherscan.io/address/0x86fa049857e0209aa7d9e616f7eb3b3b78ecfdb0#code>) In the following example a contract with address `55141015444411108922282a02aa22804481149a` is created.

```
$ contract_code =
"606060405260126006556000600790600019169055341561001c57fe5b604051602080611c5d
833981016040528080519060200190919050505b5b60005b80600160003373fffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff1673fffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff16815260
200190815260200160002081905550806000819055505b5033600460006101000a81548173fff
ffffffffffffffffffffffffffffffffffffffffffffffff021916908373fffffffffffffffffffff
ffffffffffffffff1602179055503373fffffffffffffffffffffffffffffffffffffffffffff167f
ce241d7ca1f669fee44b6f00b8eba2df3bb514eed0f6f668f8f89096e81ed94604051809050604051
80910390a25b80600581600019169055505b505b611b2e8061012f6000396000f300606060405
2361561011b576000357c01000000000000000000000000000000000000000000000000000000
00900463ffffffff16806306fdde031461011d57806307da68f51461014b578063095ea7b3146
1015d57806313af4035146101b457806318160ddd146101ea57806323b872dd14610210578063
313ce567146102865780633452f51d146102ac5780635ac801fe1461031557806369d3e20e146
1033957806370a082311461036b57806375f12b21146103b55780637a9e5e4b146103df578063
```

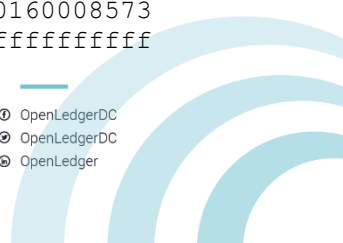


8402181f146104155780638da5cb5b1461047e57806390bc1693146104d057806395d89b41146  
10502578063a9059cbb14610530578063be9a655514610587578063bf7e214f14610599578063  
dd62ed3e146105eb575bfe5b341561012557fe5b61012d610654565b604051808260001916600  
01916815260200191505060405180910390f35b341561015357fe5b61015b61065a565b005b34  
1561016557fe5b61019a600480803573fffffffffffffffffffffffffffffffffffff16906  
02001909190803590602001909190505061075e565b6040518082151515158152602001915050  
60405180910390f35b34156101bc57fe5b6101e8600480803573fffffffffffffffffffff  
ffffffffffffffffffff1690602001909190505061083c565b005b34156101f257fe5b6101fa610920  
565b6040518082815260200191505060405180910390f35b341561021857fe5b61026c6004808  
03573fffffffffffffffffffff1690602001909190803573fffffffffffff  
ffffffffffff16906020019091905050610a11565b604051808215151515815  
260200191505060405180910390f35b341561031d57fe5b610337600480803560001916906020  
019091905050610a38565b005b341561034157fe5b61036960048080356fffffffffffff  
ffff16906020019091905050610a7e565b005b341561037357fe5b61039f600480  
803573ffffffffffffffffffff16906020019091905050610c44565b6  
040518082815260200191505060405180910390f35b34156103bd57fe5b6103c5610c8e565b60  
4051808215151515815260200191505060405180910390f35b34156103e757fe5b61041360048  
0803573fffffffffffff16906020019091905050610ca1565b  
005b341561041d57fe5b610464600480803573fffffffffffff  
f169060200190919080356fffffffffffff1690602001909190505061  
0d85565b604051808215151515815260200191505060405180910390f35b341561048657fe5b6  
1048e610dad565b604051808273fffffffffffff1673ffffff  
ffffffffffff16815260200191505060405180910390f35b3415610  
4d857fe5b61050060048080356fffffffffffff169060200190919050  
50610dd3565b005b341561050a57fe5b610512610f99565b60405180826000191660001916815  
260200191505060405180910390f35b341561053857fe5b61056d600480803573ffffffffff  
ffff16906020019091908035906020019091905050610f9f565b6  
04051808215151515815260200191505060405180910390f35b341561058f57fe5b6105976110  
7d565b005b34156105a157fe5b6105a9611181565b604051808273fffffffffffff  
ffff1673ffffff  
ffff1681526020019150  
5060405180910390f35b34156105f357fe5b61063e600480803573fffffffffffff  
ffff1690602001909190803573fffffffffffff  
ff169060200190919050506111a7565b6040518082815260200191505060405180910390f35b6  
0075481565b61069061068b336000357fffffffff000000000000000000000000000000000000000  
00000000000000000000001661122f565b611491565b60006000600435915060243590508060001  
91682600019163373fffffffffffff166000357ffffffffff00  
00167bffffffffff  
ffff1916346000366040518084815260200180602001  
82810382528484828181526020019250808284378201915050945050505060405180910390a  
46001600460146101000a81548160ff0219169083151502179055505b5b50505b565b60006107  
79600460149054906101000a900460ff1615611491565b6000600060043591506024359050806  
000191682600019163373fffffffffffff166000357ffffffffff  
ff00167bffffffffff  
ffff191634600036604051808481526020018060  
200182810382528484828181526020019250808284378201915050945050505060405180910  
390a461082f85856114a2565b92505b5b50505b92915050565b61087261086d336000357fffff  
ffff001661122f565b61149  
1565b80600460006101000a81548173fffffffffffff021916  
908373fffffffffffff1602179055506004600090549061010  
00a900473fffffffffffff1673ffffffffff  
ffff167fce241d7ca1f669fee44b6fc00b8eba2df3bb514eed0f6f668f8f89096  
e81ed9460405180905060405180910390a25b5b50565b600060005490505b90565b6000610946  
600460149054906101000a900460ff1615611491565b600060006004359150602435905080600





008373ff1673ffffffffffffffffffffffffffffffff  
ffffffffffffffff1681526020019081526020016000205490505b92915050565b60003073ffffff  
ff168373ffffffffffffffffffffffffffffffff  
fff16141561126e576001905061148b565b600460009054906101000a900473ffffffffffff  
ffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffff168373f  
ffffffffffffffffffffffffffffffff1614156112cd576001905061148b565b600073  
ffffffffffffffffffffffff16600360009054906101000a900473ffffff  
ffffffffffffffff1673ffffffffffffffffffffffff16141561132d576000905061148b565b600360009054906101000a900473ffffff  
ffffffff1673ffffffffffffffff1663b70096  
138430856000604051602001526040518463ffffffff167c010000000000000000000000  
00  
ffffffff1673ffffffffffffffff1681526020018373ffffff  
ffffffff1673ffffffffffffffff1673ffffffffffffffff168152602001827bffffff  
bffffffff19167  
bffffffff19168152602001935050  
5050602060405180830381600087803b151561146957fe5b6102c65a03f1151561147757fe5b5  
0505060405180519050905061148b565b5b5b5b92915050565b80151561149e5760006000fd5b  
5b50565b600081600260003373ffffffff1673ffffff  
ffffffff16815260200190815260200160002060008573ffffff  
ffffffff1673ffffffff168152602001908152602001600020819055508273ffffff  
ffffff163373ffffff167f8c5be1e5ebec7d5bd14f7  
1427d1e84f3dd0314c0f7b2291e5b200ac8c7c3b9258460405180828152602001915050604051  
80910390a3600190505b92915050565b600081600160008673ffffff  
ffffff1673ffffff16815260200190815260  
200160002054101515156115e257fe5b81600260008673ffffff  
ffffff1673ffffff168152602001908152602001600020541015151561166a57fe5b6116  
f0600260008673ffffff1673ffffff  
ffffff16815260200190815260200160002060003373ffffff  
ffffff1673ffffff16815260200  
19081526020016000205483611913565b600260008673ffffff  
ffffff1673ffffff1681526020019081526020016  
002060003373ffffff1673ffffff  
ffffff168152602001908152602001600020819055506117b9600160008673f  
ffffff1673ffffff  
ffffff1681526020019081526020016000205483611913565b600160008673ffffff  
ffffff1673ffffff  
ffffff16815260  
200190815260200160002081905550611845600160008573ffffff  
ffffff1673ffffff  
ffffff1681526020019081526020  
0160002054836118f9565b600160008573ffffff  
ffffff1673ffffff  
ffffff168152602001908152602001600020819055  
508273ffffff  
ffffff168473ffffff  
ffffff167fddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df5  
23b3ef846040518082815260200191505060405180910390a3600190505b9392505050565b600  
0828284019150811015151561190c57fe5b5b92915050565b6000828284039150811115151561  
192657fe5b5b92915050565b600081600160003373ffffff  
ffffff1673ffffff  
ffffff1681526020019081526020016000  
20541015151561197a57fe5b6119c3600160003373ffffff  
ffffff1673ffffff  
ffffff1681526020019081526020016000  
205483611913565b600160003373ffffff  
ffffff1673ffffff  
ffffff16815260200190815260200160002081905550611a  
4f600160008573ffffff  
ffffff1673ffffff  
ffffff16815260200190815260200160002054836118f9565b600160008573  
ffffff  
ffffff1673ffffff







```
'totalSupply() ',
'transferFrom(address,address,uint256) ',
'decimals() ',
'push(address,uint128) ',
'setName(bytes32) ',
'mint(uint128) ',
'balanceOf(address) ',
'stopped() ',
'setAuthority(address) ',
'pull(address,uint128) ',
'owner() ',
'burn(uint128) ',
'symbol() ',
'transfer(address,uint256) ',
'start() ',
'authority() ',
'allowance(address,address) ',
'undefined(bytes32) ',
'LogNote(bytes4,address,bytes32,bytes32,uint256,bytes) ',
'LogSetAuthority(address) ',
'LogSetOwner(address) ',
'Transfer(address,address,uint256) ',
'Approval(address,address,uint256) ' ]
```

Before we were talking about read access to a contract. To write anything (or otherwise produce and broadcast a transaction) you need to use a slightly different approach. Consider this example, in which we'll be minting new tokens

```
$ parseInt(eos.totalSupply(), 16)
0

$ mint_result = eos.exec.mint.gas(100000)(100000)
[ '0cde8ca52002ee84f05dc66d66094e50b0e636b6',
  { ref_block_num: 23560,
    ref_block_prefix: 7292465,
    expiration: '2018-07-30T08:26:06',
    operations: [ [Array] ],
    extensions: [],
    signatures:
      [
        '1f7b87d14ba58b00c4732134f84d0a8d5062d087ee2b33ab3c0e8d9cf2b252168f4d70066786
        6371205197ab90518ca8cb3667bd158096cde40df3b1009f8210f5' ] } ]

$ parseInt(eos.totalSupply(), 16)
100000
```

To see the status of a transaction do the following commands

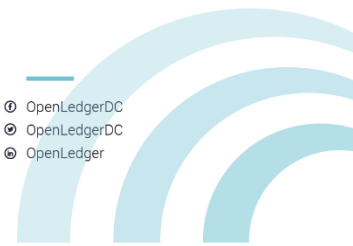
```
$ receipt = b3.get_transaction_receipt(mint_result[0])
{ id: '1.17.5',
  transaction_id: '0cde8ca52002ee84f05dc66d66094e50b0e636b6',
  operation_receipts:
    [ { success: true,
        gas_used: 66703,
```

```
gas_price: 1,  
logs: [Array],  
contract_id: '1.16.1',  
contract_address: '' } ] }
```

```
$ receipt.operation_receipts  
[ { success: true,  
  gas_used: 66703,  
  gas_price: 1,  
  logs: [ [Object] ],  
  contract_id: '1.16.1',  
  contract_address: '' } ]
```

Test account has previously minted tokens on its balance. In order to get EOS token balance of an account execute the following commands.

```
$ account = b3.get_account("test")  
{ id: '1.2.39',  
  membership_expiration_date: '1970-01-01T00:00:00',  
  registrar: '1.2.34',  
  referrer: '1.2.34',  
  lifetime_referrer: '1.2.34',  
  network_fee_percentage: 2000,  
  lifetime_referrer_fee_percentage: 3000,  
  referrer_rewards_percentage: 0,  
  name: 'test',  
  owner:  
    { weight_threshold: 1,  
      account_auths: [],  
      key_auths: [ [Array] ],  
      address_auths: [] },  
  active:  
    { weight_threshold: 1,  
      account_auths: [],  
      key_auths: [ [Array] ],  
      address_auths: [] },  
  options:  
    { memo_key: 'BTS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV',  
      voting_account: '1.2.5',  
      num_witness: 0,  
      num_committee: 0,  
      votes: [],  
      extensions: [] },  
  statistics: '2.6.39',  
  whitelisting_accounts: [],  
  blacklisting_accounts: [],  
  whitelisted_accounts: [],  
  blacklisted_accounts: [],  
  owner_special_authority: [ 0, {} ],  
  active_special_authority: [ 0, {} ],  
  top_n_control_flags: 0,  
  nonce: 0 }  
  
$ parseInt(eos.balanceOf(account.id), 16)  
100000
```





Since test account possesses tokens it can transfer them to any account (e.g nathan)

```
$ parseInt(eos.balanceOf(b3.get_account("nathan").id), 16)
0

$ transfer_result = eos.exec.transfer.gas(80000)(b3.get_account("nathan").id,
5)
[ 'b679b250f9bfce3d6930fbca0b17bf7b0a54a1ff',
  { ref_block_num: 23692,
    ref_block_prefix: 417432252,
    expiration: '2018-07-30T08:30:42',
    operations: [ [Array] ],
    extensions: [],
    signatures:
      [
'200e5540b7f6ea1826283161c94972d359e4c00e72b671a17619078099cd5e61bc6e5b9bc962
4f48c01d076d1852b80ee7082f666eae53871114f9f3b1e97616de' ] } ]

$ receipt = b3.get_transaction_receipt(transfer_result[0])
{ id: '1.17.7',
  transaction_id: 'b679b250f9bfce3d6930fbca0b17bf7b0a54a1ff',
  operation_receipts:
    [ { success: true,
      gas_used: 54371,
      gas_price: 1,
      logs: [Array],
      contract_id: '1.16.1',
      contract_address: '' } ] ]

$ receipt.operation_receipts
[ { success: true,
  gas_used: 54371,
  gas_price: 1,
  logs: [ [Object], [Object] ],
  contract_id: '1.16.1',
  contract_address: '' } ]

$ parseInt(eos.balanceOf(b3.get_account("nathan").id), 16)
5
```

To see the list of supported commands execute the following

```
$ console.log(b3.help())
```