Genaro Network

Genaro Network

DApp Develop Guide

V1.0

Genaro Network

1. How to join Genaro testnet

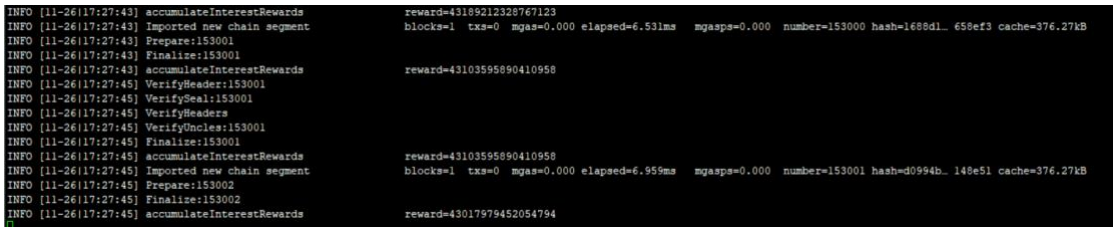
Install docker:
Centos: yum install docker
Ubuntu: apt-get install docker

Pull testnet node doker image:
docker pull genaronetwork/go-genaro-test

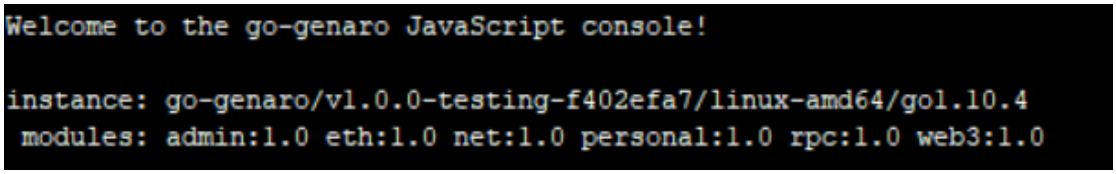
Start a Genaro node for testnet:
docker run -d --name "testnode" -p 30303:30303 -p 8545:8545 genaronetwork/go-genaro-test

See logs:
docker logs -f testnode

Wait for synchronization:



Client attach node:
docker run -it --rm genaronetwork/go-genaro-test go-genaro attach http://[ip]:8545



2. Write smart contract

Download genaro-solidity compiler:
wget https://github.com/GenaroNetwork/genaro-solidity/releases/download/1.0.0/solc-linux.zip
unzip solc-linux.zip
chmod 0775 solc

create a contract:
touch test.sol

cat test.sol:

```
pragma solidity ^0.4.13;
contract MyContract {
	mapping(address => uint) public scores;

	function SetScore(uint score) public {
		scores[msg.sender] = score;
	}
}
```

Building source code:
./solc test.sol --bin --abi --optimize -o test-output/

```
>ls test-output/
MyContract.abi    MyContract.bin
```

>cat test-output/MyContract.abi

[{"constant":true,"inputs":[{"name":"","type":"address"}],"name":"scores","outputs":[{"name":"","type":"uint256"}],"payable":false,"stateMutability":"view","type":"function"},{"constant":false,"inputs":[{"name":"score","type":"uint256"}],"name":"SetScore","outputs":[],"payable":false,"stateMutability":"nonpayable","type":"function"}]

>cat test-output/MyContract.bin

60806040523480156100105760008fd5b5060f18061001f6000396000f30060806040526004361060485763ffffffff7c0100000000000000000000000000000000000000000000000000000000600003504166376dd110f8114604d57806396e9c85314608a575b600080fd5b34801560587600080fd5b50607873ffffffffffffffffffffffffffffffffffffffff6004351660a1565b60408051918252519081900360200190f35b34801560957600080fd5b50609f60043560b3565b005b6000602081905290815260409020548156b33600090815260208190526040902055600a165627a7a7230582063796536ca4052d134adb0f2ae3f5a27e8073fb09cece937cba5ba91944c3d4e0029

If your solc can not run,you can pull Genaro solc image:

```
docker pull genaronetwork/genaro-solc
docker run --rm -v `pwd`:/root:z genaronetwork/genaro-solc solc /root/test.sol --bin --abi --optimize -o /root
```

You can get more help from the website(https://solidity.readthedocs.io/en/v0.4.24/).

3. Deploy contract

Attach node:

docker run -it --rm genaronetwork/go-genaro-test go-genaro attach http://[ip]:8545

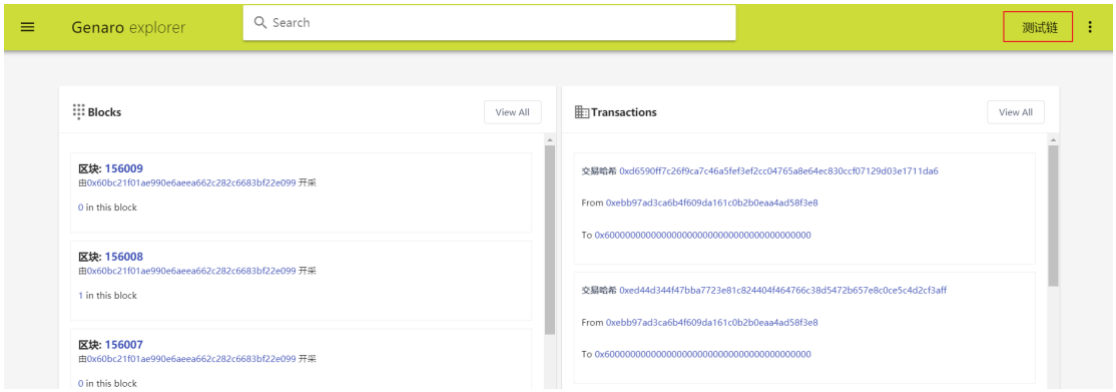
create a account:

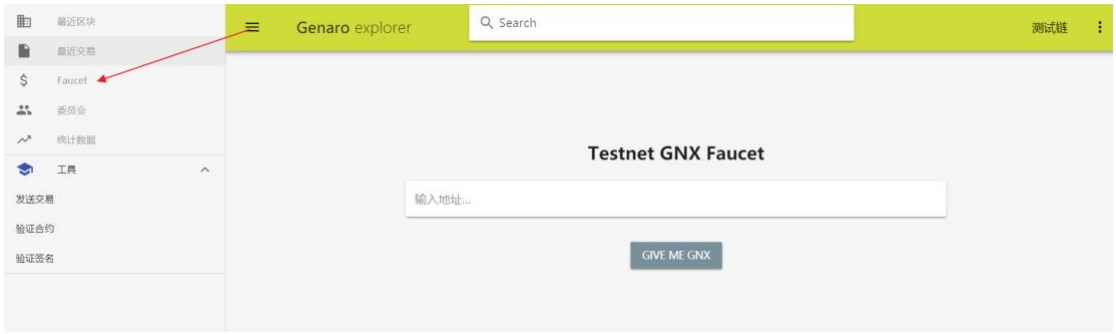
> personal.newAccount("your password")

0x…… (this is your account address)

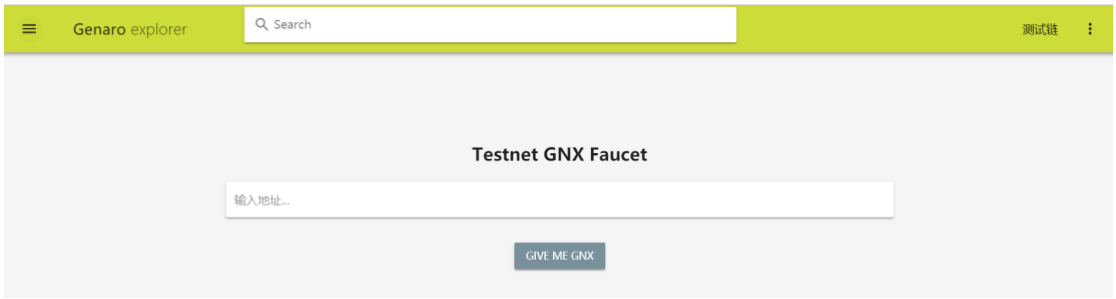
Get some balance for your account:

Go to: http://explorer.genaro.network/





Input your address:



Check your balance:

>eth.getBalance("your account")

5000000000000000000

Deploy contract with your account:

> code =

"0x6080604052348015610010576000080fd5b5060f18061001f6000396000f300608060405260
04361060485763ffffffff7c0100000000000000000000000000000000000000000000000000000000
0000600035041663376dd110f8114604d57806396e9c85314608a575b600080fd5b348015605
857600080fd5b50607873ffffffffffffffffffffffffffffffffffffffff6004351660a1565b604080519182525
19081900360200190f35b34801560957600080fd5b50609f60043560b3565b005b60006020
8190529081526040902054815650b3360009081526020819052604090205555600a165627a7a7
230582063796536ca4052d134adb0f2ae3f5a27e8073fb09cece937cba5ba91944c3d4e0029"

> abi =

[{"constant":true,"inputs":[{"name":"","type":"address"}],"name":"scores","outputs":[{"name":"",
"type":"uint256"}],"payable":false,"stateMutability":"view","type":"function"},{"constant":false,"i
nputs":[{"name":"score","type":"uint256"}],"name":"SetScore","outputs":[],"payable":false,"stat
eMutability":"nonpayable","type":"function"}]

> myContract = eth.contract(abi)

...

> personal.unlockAccount("your account address")

···(input your password)

> contract = myContract.new({from:"your account address",data:code,gas:1000000})

...

```
{
   abi: [{
         constant: false,
         inputs: [{...}],
         name: "SetScore",
         outputs: [],
         payable: false,
         stateMutability: "nonpayable",
         type: "function"
   }],
   address: undefined,
   transactionHash: "your contract transaction address"
}
```

Get your contract address:

>eth.getTransactionReceipt("your contract transaction address")

...

contractAddress: "your contract address",

...

4. Use contract

Attach node:

docker run -it --rm genaronetwork/go-genaro-test go-genaro attach http://[ip]:8545

> abi =

[{"constant":true,"inputs":[{"name":"","type":"address"}],"name":"scores","outputs":[{"name":"","type":"uint256"}],"payable":false,"stateMutability":"view","type":"function"},{"constant":false,"inputs":[{"name":"score","type":"uint256"}],"name":"SetScore","outputs":[],"payable":false,"stateMutability":"nonpayable","type":"function"}]

...

> myContract=eth.contract(abi)

...

> contract=myContract.at("your contract address")

...

Get your score:

> contract.scores("your account address")

...

Set your score:

> personal.unlockAccount("your account address")

···(input your password)

> contract.SetScore.sendTransaction(10,{from: "your account address",gas:1000000})

"your transaction address"

>eth.getTransactionReceipt("your transaction address")

...

status: "0x1",(0x0 is fail,0x01 is success)

...

> contract.scores(eth.accounts[0])

10

The contract worked correctly.

5. Develop android DApp client

Start your own server node:

docker run -d --name "server" -p 30310:30303 -p 8560:8545 --env SERVER="true" genaronetwork/go-genaro-test

download web3j:

wget https://github.com/web3j/web3j/releases/download/v3.4.0/web3j-3.4.0.zip

generate java file for contract:

./web3j solidity generate /opt/contract/MyContract.bin /opt/contract/MyContract.abi -o /opt/contract/ -p com.genaro

Generate file: com/genaro/MyContract.java

Java code:

```
        String rpcAddr = "your rpc address";
        String contractAddress = "your contract address";

        Web3j web3 = Web3j.build(new HttpService(rpcAddr));

        try {
            Credentials ownerAddrCredentials =
WalletUtils.loadCredentials("your password", "your account key file");
            MyContract contract = MyContract.load(contractAddress, web3,
ownerAddrCredentials, web3.ethGasPrice().send().getGasPrice(),
BigInteger.valueOf(1000000));
            TransactionReceipt rec =
contract.SetScore(BigInteger.valueOf(20)).send();
            if (rec.isStatusOK()) {
                System.out.println(contract.scores("your account
address").send().intValue());
            }
        } catch (IOException e) {
            e.printStackTrace();
        } catch (CipherException e) {
            e.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        }
```

6. use Genaro Network storage

Java lib: https://github.com/GenaroNetwork/libgenaro-java

Java code example:

Initialize:

```
String V3JSON = "{ \"address\":
\"aaad65391d2d2eafda9b27326d1e81d52a6a3dc8\",
       \"crypto\": { \"cipher\": \"aes-128-ctr\",
```

```
       \"ciphertext\":
\"e968751f3d60827b6e62e3ff6c024ecc82f33a6c55428be33249c83edba444ca\",
       \"cipherparams\": { \"iv\": \"e80d9ec9ba6241a143c756ec78066ad9\" },
\"kdf\": \"scrypt\",
       \"kdfparams\": { \"dklen\": 32, \"n\": 262144, \"p\": 1, \"r\": 8,
\"salt\":

\"ea7cb2b004db67d3103b3790caced7a96b636762f280b243e794fb5bef8ef74b\" },
       \"mac\":
\"ceb3789e77be8f2a7ab4d205bf1b54e048ad3f5b080b96e07759de7442e050d2\" },
       \"id\": \"e28f31b4-1f43-428b-9b12-ab586638d4b1\", \"version\": 3 }";
String passwd = "xxxxxx";
Genaro api = new Genaro("http://192.168.50.206:8080");
GenaroWallet gw = new GenaroWallet(V3JSON, passwd);
api.logIn(gw);
```

Upload file:

```
String bucketId = "5bfcf4ea7991d267f4eb53b4";
String filePath = "xxxxxxxxx";
String fileName = "xxx";
new Uploader(api, false, filePath, fileName, bucketId, new Progress() {
    @Override
    public void onBegin() {
    }
    @Override
    public void onEnd(int status) {
    }
    @Override
    public void onProgress(float progress, String message) {
    }
}).start();
```

Download file:

```
String V3JSON = "{ \"address\":
\"aaad65391d2d2eafda9b27326d1e81d52a6a3dc8\",
       \"crypto\": { \"cipher\": \"aes-128-ctr\",
       \"ciphertext\":
\"e968751f3d60827b6e62e3ff6c024ecc82f33a6c55428be33249c83edba444ca\",
       \"cipherparams\": { \"iv\": \"e80d9ec9ba6241a143c756ec78066ad9\" },
\"kdf\": \"scrypt\",
```

```
       \"kdfparams\": { \"dklen\": 32, \"n\": 262144, \"p\": 1, \"r\": 8,
\"salt\":

\"ea7cb2b004db67d3103b3790caced7a96b636762f280b243e794fb5bef8ef74b\" },
       \"mac\":
\"ceb3789e77be8f2a7ab4d205bf1b54e048ad3f5b080b96e07759de7442e050d2\" },
       \"id\": \"e28f31b4-1f43-428b-9b12-ab586638d4b1\", \"version\": 3 }";

String passwd = "xxxxxx";

Genaro api = new Genaro("http://192.168.50.206:8080");
GenaroWallet gw = new GenaroWallet(V3JSON, passwd);
api.logIn(gw);

String bucketId = "5bfcf4ea7991d267f4eb53b4";
String fileId = "5c0103fd5a158a5612e67461";
String filePath = "xxxxxxxxx";
new Downloader(api, bucketId, fileId, filePath, new Progress() {
    @Override
    public void onBegin() {
    }

    @Override
    public void onEnd(int status) {
    }

    @Override
    public void onProgress(float progress, String message) {
    }
}).start();
```