Proof of Authority Development Chain Documentation

Please refer to the README for environment installation setup instructions.

Network Configuration of Genesis Block:

1. Create an account for node1 using the following command:
   ./geth --datadir node1 account new
2. Enter a password of your choice for node1.

```
(base) drewdisbrowmarnell@drews-mbp blockchain-hw % ./geth --datadir node1 account new
INFO [05-25|15:54:03.197] Maximum peer count                       ETH=50 LES=0 total=50
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:

Your new key was generated

Public address of the key:   0x0E2BB56f68b2AcA5E4245F27690032CDBdC8b48A
[redacted]

- You can share your public address with anyone. Others need it to interact with you.
- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!
```

3. Copy down password and key information for node1.
4. Repeat steps 1-3 for node2.
5. Run the puppeth program and follow the prompts as below to create a new blockchain:

```
(base) drewdisbrowmarnell@drews-mbp blockchain-hw % ./puppeth
+-----------------------------------------------------------+
| Welcome to puppeth, your Ethereum private network manager |
|                                                           |
| This tool lets you create a new Ethereum network down to  |
| the genesis block, bootnodes, miners and ethstats servers |
| without the hassle that it would normally entail.         |
|                                                           |
| Puppeth uses SSH to dial in to remote servers, and builds |
| its network components out of Docker containers using the |
| docker-compose toolset.                                   |
+-----------------------------------------------------------+

Please specify a network name to administer (no spaces, hyphens or capital letters please)
> testnet

Sweet, you can set this via --network=testnet next time!
```

```
What would you like to do? (default = stats)
 1. Show network stats
 2. Configure new genesis
 3. Track new remote server
 4. Deploy network components
> 2

What would you like to do? (default = create)
 1. Create new genesis from scratch
 2. Import already existing genesis
> 1

Which consensus engine to use? (default = clique)
 1. Ethash - proof-of-work
 2. Clique - proof-of-authority
> 2

How many seconds should blocks take? (default = 15)
> 15

Which accounts are allowed to seal? (mandatory at least one)
> 0x0E2BB56f68b2AcA5E4245F27690032CDBdC8b48A
> 0xcA401Ed2BfeB36E3BC03744944Db922627c3e05e
> 0x

Which accounts should be pre-funded? (advisable at least one)
> 0x0E2BB56f68b2AcA5E4245F27690032CDBdC8b48A
> 0xcA401Ed2BfeB36E3BC03744944Db922627c3e05e
> 0x

Should the precompile-addresses (0x1 .. 0xff) be pre-funded with 1 wei? (advisable yes)
> no

Specify your chain/network ID if you want an explicit one (default = random)
> 999
INFO [05-25|15:59:49.896] Configured new genesis block
```

- Here we are configuring a new genesis block for our blockchain named testnet
- clique sets up the blockchain as a proof-of-authority consensus engine
- blocktime defines the time is takes to mine a block (in this case 15 seconds)
- The two accounts entered *must* be the account addresses copied down in step 3 for node1 and node2
- Type no so addresses are not pre-funded with wei, this keeps the genesis block cleaner
- We choose a number to specific the chain/network ID, an additional unique chain identifier, in this case 999

6. When the original puppeth prompt returns after configuring new genesis block, enter the following commands to export genesis configurations (testnet.json).
    - **If blank hit enter

```
What would you like to do? (default = stats)
 1. Show network stats
 2. Manage existing genesis
 3. Track new remote server
 4. Deploy network components
> 2

 1. Modify existing configurations
 2. Export genesis configurations
 3. Remove genesis configuration
> 2

Which folder to save the genesis specs into? (default = current)
  Will create testnet.json, testnet-aleth.json, testnet-harmony.json, testnet-parity.json
>
INFO [05-25|16:00:39.411] Saved native genesis chain spec          path=testnet.json
ERROR[05-25|16:00:39.411] Failed to create Aleth chain spec        err="unsupported consensus eng
ine"
ERROR[05-25|16:00:39.411] Failed to create Parity chain spec       err="unsupported consensus eng
ine"
INFO [05-25|16:00:39.411] Saved genesis chain spec                 client=harmony path=testnet-ha
rmony.json
```

7. Press control C to exit out of the puppeth program.
8. Initialize node1 with the following command:
   ./geth –datadir node1 init testnet.json

```
(base) drewdisbrowmarnell@Drews-MacBook-Pro blockchain-hw % ./geth --datadir node1 init testnet.j
son
INFO [05-25|16:05:51.419] Maximum peer count                       ETH=50 LES=0 total=50
INFO [05-25|16:05:51.441] Allocated cache and file handles         database=/Users/drewdisbrowmar
nell/Code/Fintech/blockchain-hw/node1/geth/chaindata cache=16.00MiB handles=16
INFO [05-25|16:05:51.518] Writing custom genesis block
INFO [05-25|16:05:51.520] Persisted trie from memory database      nodes=3 size=457.00B time=218.
915µs gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [05-25|16:05:51.522] Successfully wrote genesis state         database=chaindata hash=63c9f6
…944036
INFO [05-25|16:05:51.522] Allocated cache and file handles         database=/Users/drewdisbrowmar
nell/Code/Fintech/blockchain-hw/node1/geth/lightchaindata cache=16.00MiB handles=16
INFO [05-25|16:05:51.595] Writing custom genesis block
INFO [05-25|16:05:51.596] Persisted trie from memory database      nodes=3 size=457.00B time=836.
182µs gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [05-25|16:05:51.597] Successfully wrote genesis state         database=lightchaindata hash=6
3c9f6…944036
```

- init flag initializes node1

9. Repeat step 9 for node2.

To Start the Testnet Network:

1. Open the terminal and move to the directory where the blockchain resides.
2. Enter the following command:
   ./geth --datadir node1 --unlock 0x0E2BB56f68b2AcA5E4245F27690032CDBdC8b48A --mine --rpc --allow-insecure-unlock
   - --datadir flag to call the data directory for node1
   - --unlock flag to unlock node1
   - 0x0E2… is the public address to node1 of the blockchain
   - --mine flag tells the node to mine new blocks

- --rpc flag enables us to talk to node1, which will allow us to use MyCrypto to transact on our chain
- --allow-insecure-unlock flag allows insecure account unlocking when account-related RPCs are exposed by http

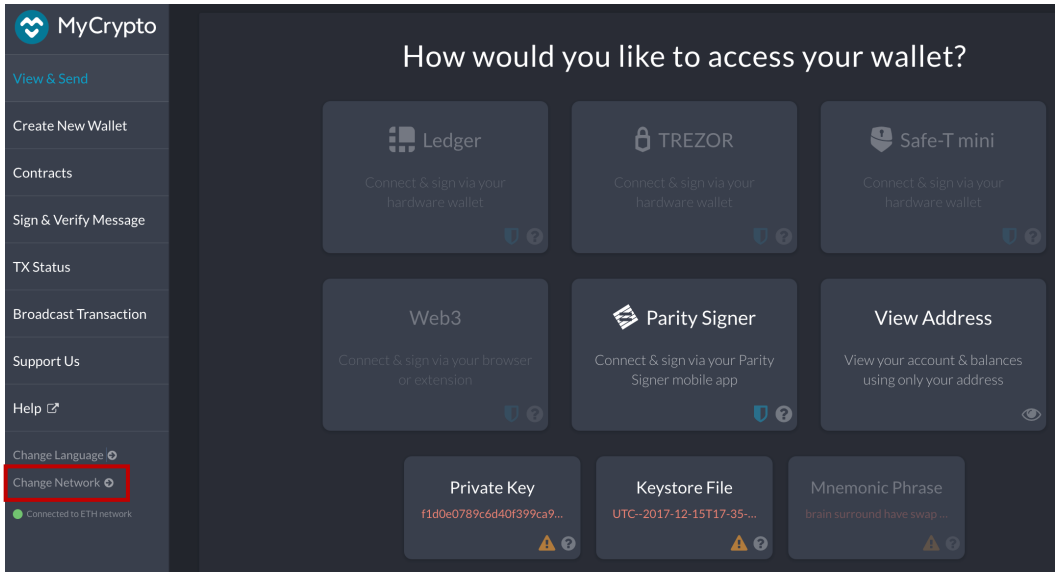
3. Find and copy down the enode address:
   Hint: look for Started P2P networking

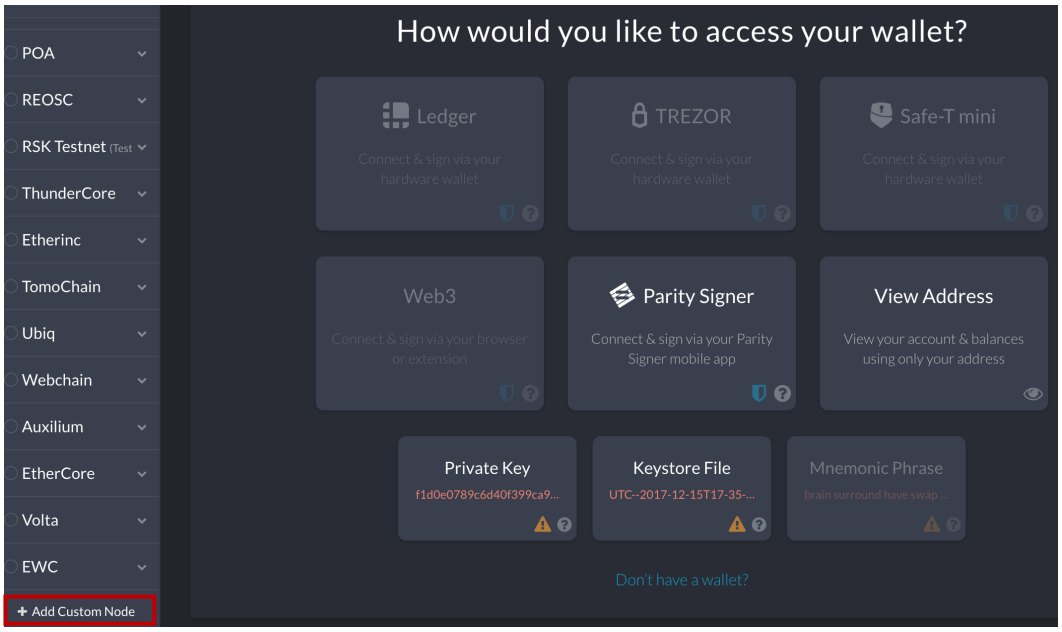
```
INFO [05-25|16:57:27.375] Started P2P networking                    self=enode://8dc7a7dc73b2fea10
350c3c6c8563e4feafbe5441dfe9a19e5f3a234c8cedd39dcf21f8591fc3bde697d1a872fbe9cbd870cbe9cc0c20f9d27
2c5314c853b612@24.12.190.227:30303
```

4. Enter the password.
5. Open a new terminal window and enter the following command:
   ./geth --datadir node2 --unlock 0xcA401Ed2BfeB36E3BC03744944Db922627c3e05e --mine --port 30304 --bootnodes enode://8dc7a7dc73b2fea10350c3c6c8563e4feafbe5441dfe9a19e5f3a234c8cedd39dcf21f8591fc3bde697d1a872fbe9cbd870cbe9cc0c20f9d272c5314c853b612@127.0.0.1:30303 --allow-insecure-unlock
   - Many of these flags are the same as the node1 command
   - Here we are using the address for node2 0xcA…
   - --port 30304 is specifying a port since we used 30303 for node1
   - --bootnodes flag allows you to pass the network info needed to find other nodes in the blockchain, this allows us to connect our nodes
   - enode://8dc… is the enodeid of node1 we copied down earlier
   - If using Microsoft Window, you must add the flag –ipcdisable due to the way Windows spawns new IPC/Unix sockets
6. Enter the password.
7. Congrats! Your network should begin mining!

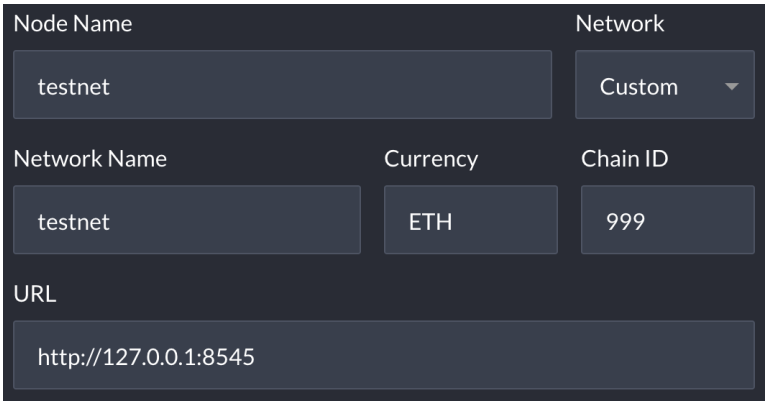
Sending a Transaction:

1. Open MyCrypto.
2. Select Change Network then scroll down to the bottom of the network list and select Add Custom Node.

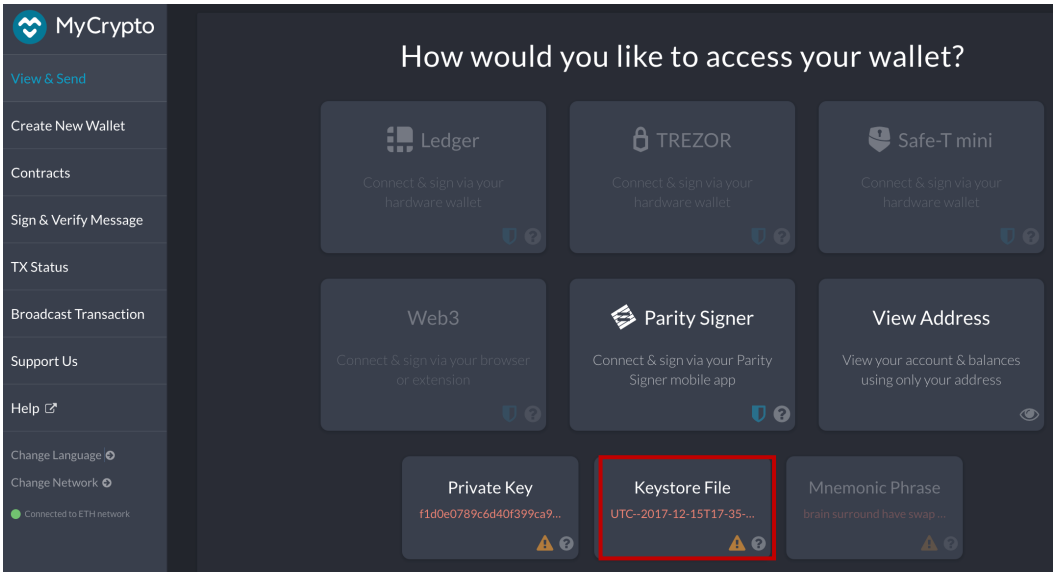




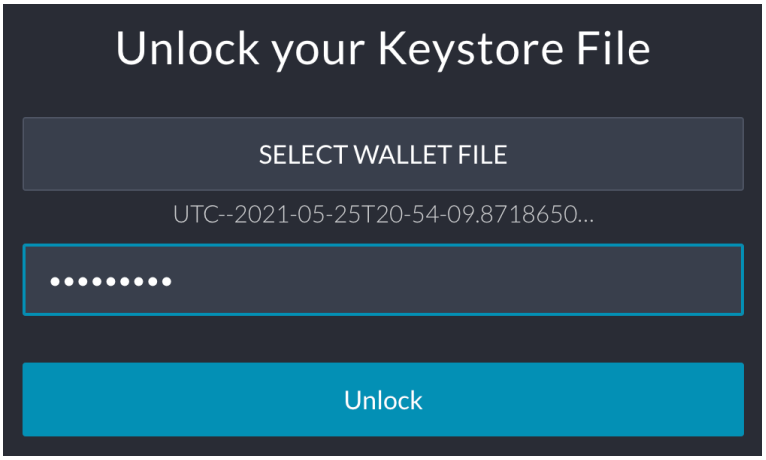
3. Select Custom for Network and fill in fields accordingly, being sure Chain ID matches the Chain ID selected when configuring the genesis block. Then Add & Use Network.



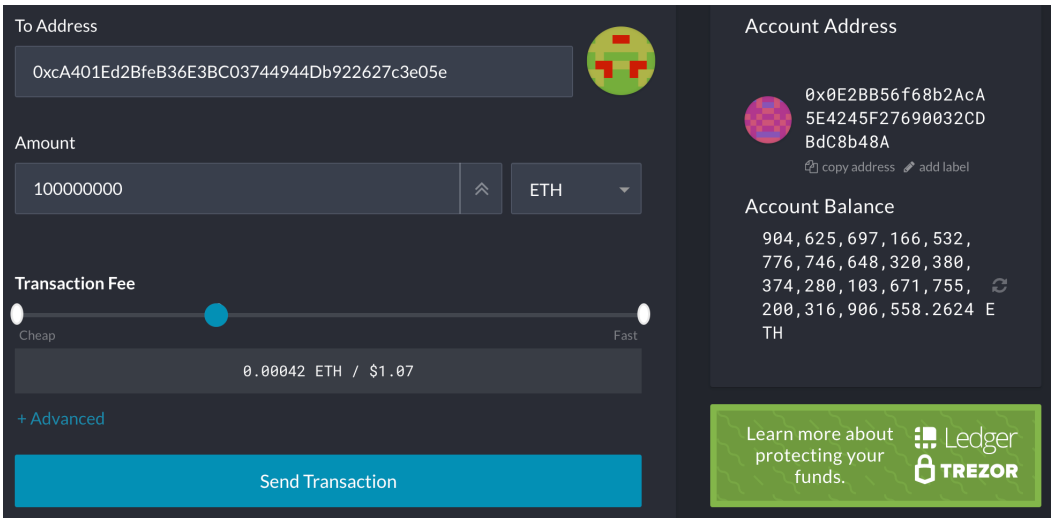
4. Navigate to home screen and select Keystore File.



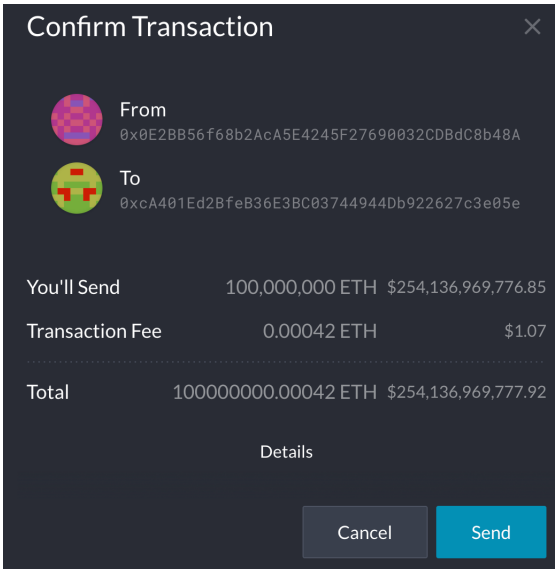
5. Select the keystore file for node1. This can be found in the directory for node1 inside the directory where the blockchain resides. Enter your password for node1 and hit Select.



6. You should see the address for node1 under Account Address along with the prefunded test ether we added during the initial configuration. Enter the address for node2 in the To Address field and a large number in the Amount field to test the network. Hit Send Transaction.



7. You should see a Confirm Transaction window with the transaction details, hit Send to send the test transaction.



8. Allow some time for the transaction to go through. Logout and select Tx Status from the home navigation bar to view your transaction. If the transaction status changes from Pending to Successful the transaction was completed and the network is valid.