



THE THINGS NETWORK

LOS ANGELES



GETH



Geth (Go Ethereum) is a command line tool implemented in Go that runs a full Ethereum node.

We'll be using **geth** to:

- Create a new wallet
- Initialize our private chain
- Mine smart contracts & transactions

Docs: <https://godoc.org/github.com/ethereum/go-ethereum>

Source: <https://github.com/ethersphere/go-ethereum>

SWARM



Swarm is a distributed storage platform and content distribution service built as a native layer to Ethereum.

We'll be using **swarm** to:

- Upload uplink packets and save swarms content hash in our smart contract.
- Retrieve uplink packets

Docs: <https://swarm-guide.readthedocs.io/en/latest/>

Source: <https://github.com/ethersphere/go-ethereum/tree/master/swarm>

INSTALL GETH & SWARM



1. Create a directory for the source code

```
> mkdir -p $GOPATH/src/github.com/ethereum
```

2. Clone repository

```
> cd $GOPATH/src/github.com/ethereum
```

```
> git clone https://github.com/ethereum/go-ethereum
```

NOTE: Checkout to latest stable version - v1.7.3

INSTALL GETH & SWARM



3. Install geth and swarm

```
> cd go-ethereum
> go install ./cmd/geth/
> go install ./cmd/swarm/
```

4. Check if everything installed properly.

To go install command creates the project binaries in the workspace's bin directory (**located at \$GOPATH/bin**).

```
> cd $GOPATH/bin
> ./geth version && ./swarm version
```

DEMO APP



The demo app uses [Truffle](#), a development framework for Ethereum that allows us to easily compile, link and deploy smart contracts.

In addition, the demo app contains:

- Genesis file
- Smart contract for TTN service
- Node server for connection between TTN -> Web3
- Front end app

DEMO APP



```
> npm install -g truffle
```

```
> git clone https://github.com/async-la/ttn-eth
```

```
> cd ttn-eth
```

```
> npm install
```

CREATE DATA DIRECTORY



Designate a directory for the blockchains database and account keystore. I recommend adding this to your `~/.bashrc`, `~/.bash_profile`, or whatever your shell uses.

```
> echo "export DATADIR=/path/to/myDataDir" >> ~/.bashrc
```

```
> source ~/.bashrc
```


CREATE AN ACCOUNT



```
> geth --datadir $DATADIR account new
```

After you've entered your passphrase, you'll receive the public address of your new account. **You must remember this passphrase to unlock your account in the future.**

```
ttn-eth — christopherdro@cdro — ../OSS/ttn-eth  
[-> ttn-eth git:(master) ✗ geth --datadir $DATADIR account new  
Your new account is locked with a password. Please give a password. Do not forget this password.  
[Passphrase: ]  
[Repeat passphrase: ]  
Address: {eb43f44fa90b80bbb9c3edf9f6d9858e3d398b3d}  
-> ttn-eth git:(master) ✗
```

GENESIS BLOCK



Every blockchain starts with the genesis block. The settings of that initial block and the rest of the blockchain are defined in a single JSON file.

```
1  {
2    "config": {
3      "chainId": 9150,
4      "homesteadBlock": 0,
5      "eip155Block": 0,
6      "eip158Block": 0,
7      "byzantiumBlock": 0
8    },
9    "difficulty": "400",
10   "gasLimit": "2000000",
11   "alloc": {
12     "b361f6b28a2e67e1b3bd8ded0c4f0f19cac0c972": {
13       "balance": "100000000000000000000"
14     }
15   }
16 }
```

GENESIS BLOCK



```
> geth --datadir $DATADIR init genesis.json
```

```
ttn-eth — christopherdro@cdro — ../OSS/ttn-eth
[→ ttn-eth git:(master) geth --datadir $DATADIR init genesis.json ]
INFO [01-17|14:54:56] Allocated cache and file handles      database=/tmp/BZZ/geth/chaindata cache=16 handles=16
INFO [01-17|14:54:56] Successfully wrote genesis state                database=chaindata                hash=0169c8...b5fd43
INFO [01-17|14:54:56] Allocated cache and file handles      database=/tmp/BZZ/geth/lightchaindata cache=16 handles=16
INFO [01-17|14:54:56] Successfully wrote genesis state                database=lightchaindata           hash=0169c8...b5fd43
→ ttn-eth git:(master) x
```

STARTING GETH



```
> geth --datadir $DATADIR --rpc console --rpccorsdomain '*'
```

```
ttn-eth — geth --datadir $DATADIR --rpc console --rpccorsdomain '*' — geth
ttn-eth git:(master) x geth --datadir $DATADIR --rpc console --rpccorsdomain '*'
INFO [01-17|16:58:34] Starting peer-to-peer node instance=Geth/v1.7.3-stable/darwin-amd64/go1.8.3
INFO [01-17|16:58:34] Allocated cache and file handles database=/Users/christopherdro/Desktop/OSS/ttn-eth/dataDir/geth/chaindata
cache=128 handles=1024
INFO [01-17|16:58:34] Writing default main-net genesis block
INFO [01-17|16:58:34] Initialised chain configuration config="{ChainID: 1 Homestead: 1150000 DAO: 1920000 DAOsupport: true EIP15
0: 2463000 EIP155: 2675000 EIP158: 2675000 Byzantium: 4370000 Engine: ethash}"
INFO [01-17|16:58:34] Disk storage enabled for ethash caches dir=/Users/christopherdro/Desktop/OSS/ttn-eth/dataDir/geth/ethash count=3
INFO [01-17|16:58:34] Disk storage enabled for ethash DAGs dir=/Users/christopherdro/.ethash count=2
INFO [01-17|16:58:34] Initialising Ethereum protocol versions=[63 62] network=1
INFO [01-17|16:58:34] Loaded most recent local header number=0 hash=d4e567...cb8fa3 td=17179869184
INFO [01-17|16:58:34] Loaded most recent local full block number=0 hash=d4e567...cb8fa3 td=17179869184
INFO [01-17|16:58:34] Loaded most recent local fast block number=0 hash=d4e567...cb8fa3 td=17179869184
INFO [01-17|16:58:34] Regenerated local transaction journal transactions=0 accounts=0
INFO [01-17|16:58:34] Starting P2P networking
INFO [01-17|16:58:36] UDP listener up self=enode://f7921eefbb96d5697ba17abf28e7ad47bc43653d7eb86cdfae139dde6038c
f195324a4f2bf67e628db5e03c3b080e0b55ea99d8716bc375a794e3fa9d5e3fbf[:]::30303
INFO [01-17|16:58:36] RLPx listener up self=enode://f7921eefbb96d5697ba17abf28e7ad47bc43653d7eb86cdfae139dde6038c
f195324a4f2bf67e628db5e03c3b080e0b55ea99d8716bc375a794e3fa9d5e3fbf[:]::30303
INFO [01-17|16:58:36] IPC endpoint opened: /Users/christopherdro/Desktop/OSS/ttn-eth/dataDir/geth.ipc
INFO [01-17|16:58:36] HTTP endpoint opened: http://127.0.0.1:8545
Welcome to the Geth JavaScript console!

instance: Geth/v1.7.3-stable/darwin-amd64/go1.8.3
coinbase: 0x69d50623ef032c0af3d0a146c89337592b89a932
at block: 0 (Wed, 31 Dec 1969 16:00:00 PST)
datadir: /Users/christopherdro/Desktop/OSS/ttn-eth/dataDir
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

> |
```

STARTING SWARM



```
> swarm --bzzaccount $BZZ --datadir $DATADIR --ens-api '' --corsdomain '*'
```

```
ttn-eth git:(master) x swarm --bzzaccount $BZZ --datadir $DATADIR --ens-api '' --corsdomain '*'  
Unlocking swarm account 0x69d50623EF032C0aF3D0A146c89337592B89a932 [1/3]  
Passphrase:  
INFO [01-17|17:00:36] Starting peer-to-peer node instance=swarm/v1.7.3-stable/darwin-amd64/go1.8.3  
WARN [01-17|17:00:37] No ENS, please specify non-empty --ens-api to use domain name resolution  
INFO [01-17|17:00:37] Starting P2P networking  
INFO [01-17|17:00:39] UDP listener up self=enode://d241b71bb9893ea1cc1e73b811d728ab1d996a96154bbc9ffd19f5014d1cd5ea66f7ebff67207b23e3eb00da43c573b1da595eee6c5abafacfe962fe6d3d49cf@[:]:30399  
WARN [01-17|17:00:39] Starting Swarm service  
INFO [01-17|17:00:39] RLPx listener up self=enode://d241b71bb9893ea1cc1e73b811d728ab1d996a96154bbc9ffd19f5014d1cd5ea66f7ebff67207b23e3eb00da43c573b1da595eee6c5abafacfe962fe6d3d49cf@[:]:30399  
WARN [01-17|17:00:39] Warning: error reading kaddb '/Users/christopherdro/Desktop/OSS/ttn-eth/dataDir/swarm/bzz-69d50623ef032c0af3d0a146c89337592b89a932/bzz-peers.json' (skipping): open /Users/christopherdro/Desktop/OSS/ttn-eth/dataDir/swarm/bzz-69d50623ef032c0af3d0a146c89337592b89a932/bzz-peers.json: no such file or directory  
INFO [01-17|17:00:39] Swarm network started on bzz address: 63803f0fe5371b5807f00b6a73f7a9c636f51607eac00df58025fa3d012f0564  
INFO [01-17|17:00:39] Swarm http proxy started on 127.0.0.1:8500  
INFO [01-17|17:00:39] IPC endpoint opened: /Users/christopherdro/Desktop/OSS/ttn-eth/dataDir/bzzd.ipc
```

SMART CONTRACT



Smart contracts are account holding objects on the Ethereum blockchain and code functions and can interact with other contracts, make decisions, store data, and send ether to others.

Smart contracts run on the Ethereum Virtual Machine(EVM) and have no access to network, filesystem or other processes.

SMART CONTRACT



getDevice	address deviceId
getDeviceAtIndex	uint256 index
getDeviceCount	
isDevicePresent	address deviceId
getDeviceData	address device_id, uint256
getDeviceTimestamps	address device_id
setDeviceData	address deviceId, string
kill	
registerDevice	address deviceId, string

COMPILING SMART CONTRACT



Contracts are located in your project's ``contracts/`` directory. **Contract need to be compiled on initial deployment and after any changes.**

Smart contracts are written in Solidity and will have a file extension of `.sol`

```
> truffle compile
```


MIGRATING SMART CONTRACT



Migrations help us deploy contracts to the Ethereum network and are responsible for staging our deployment tasks. Truffle provides a special Migration contract that keeps a history of previously run migrations.

```
> truffle migrate
```

BACKEND



``server.js`` uses TTN's Node.JS SDK and web3 to store our devices and uplink packets in our smart contract.

Breakdown:

- Set provider for web3
- Fetch accounts (required to pay for transactions)
- Create an instance to our deployed contract
- Initialize TTN application and data clients
- Register uplink event for data client

> `node server.js`

FRONTEND



Sample front end application demonstrating how to retrieve devices stored in our smart contract and their payload data stored on swarm.

```
> npm start
```

TROUBLESHOOTING



- Check that the genesis block is pre-filled with the correct wallet address.
- Remember to unlock your wallet before migrating contracts and sending transactions. (**default is 5 min**)
- Make sure you've started mining - ``miner.start()``



[MEETUP.COM/TTN-LA](https://www.meetup.com/TTN-LA)



[@TTN_LA](https://twitter.com/TTN_LA)

THANKS!