
THE THINGS
NETWORK
LOS ANGELES

# GETH

THE THINGS NETWORK
LOS ANGELES

**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

THE THINGS NETWORK
LOS ANGELES

**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

THE THINGS NETWORK
LOS ANGELES

**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

THE THINGS NETWORK
LOS ANGELES

**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 THINGS NETWORK
LOS ANGELES

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

THE THINGS NETWORK
LOS ANGELES

```
> npm install -g truffle

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

> cd ttn-eth

> npm install
```

# CREATE DATA DIRECTORY

THE THINGS NETWORK
LOS ANGELES

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

THE THINGS NETWORK
LOS ANGELES

```
> 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.**

# GENESIS BLOCK

THE THINGS NETWORK
LOS ANGELES

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.

```
{
   "config": {
      "chainId": 9150,
      "homesteadBlock": 0,
      "eip155Block": 0,
      "eip158Block": 0,
      "byzantiumBlock": 0
   },
   "difficulty": "400",
   "gasLimit": "2000000",
   "alloc": {
      "b361f6b28a2e67e1b3bd8ded0c4f0f19cac0c972": {
          "balance": "1000000000000000000000000"
      }
   }
}
```

# GENESIS BLOCK

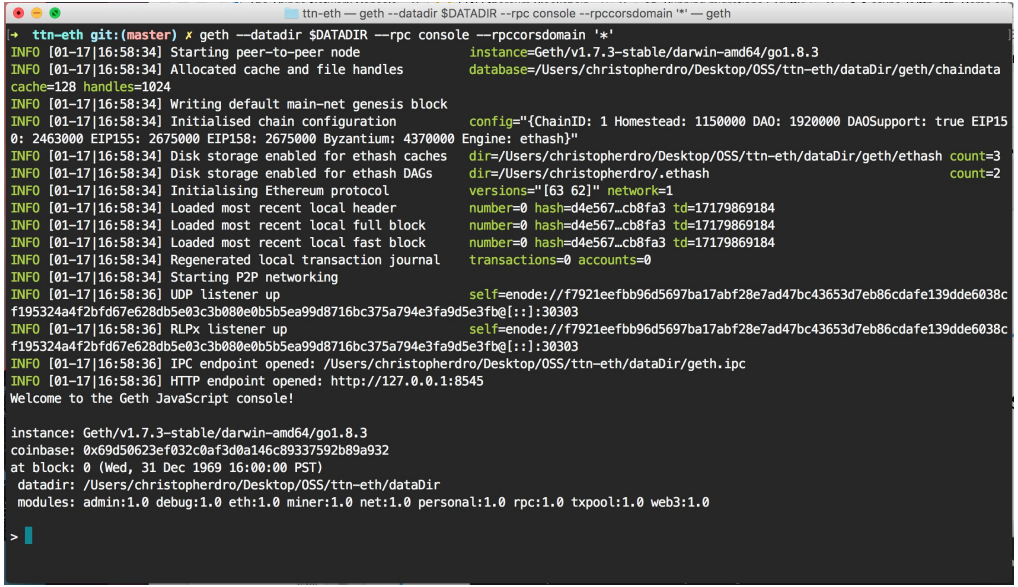
THE THINGS NETWORK
LOS ANGELES

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

# STARTING GETH

THE THINGS NETWORK
LOS ANGELES

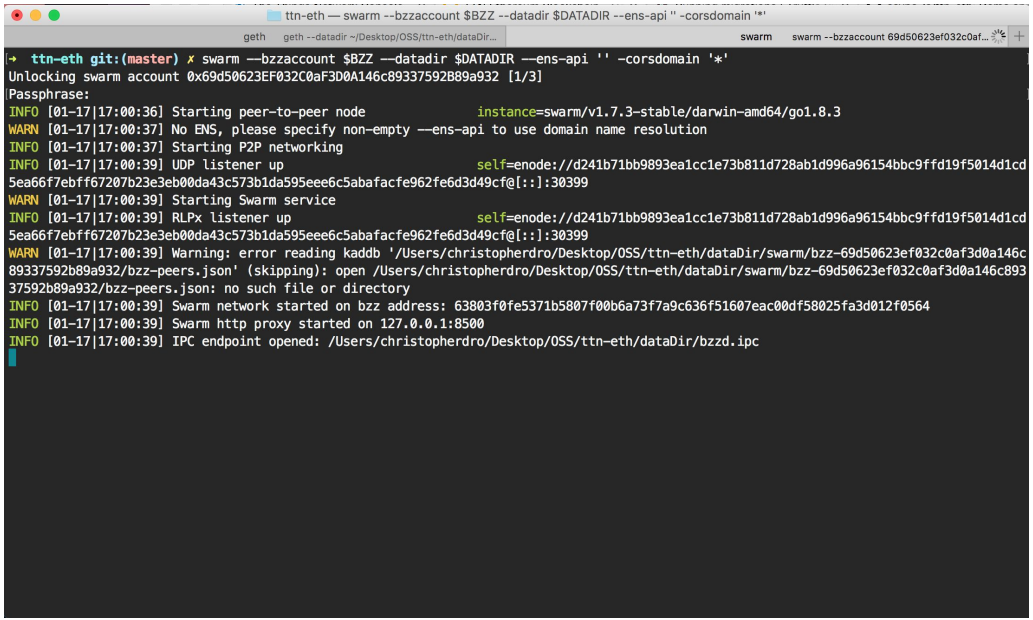
```
> geth --datadir $DATADIR --rpc console --rpccorsdomain '*'
```



# STARTING SWARM

THE THINGS NETWORK
LOS ANGELES

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



# 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

THE THINGS NETWORK
LOS ANGELES

| | |
|---|---|
| getDevice | address deviceId |
| getDeviceAtIndex | uint256 index |
| getDeviceCount | |
| isDevicePresent | address deviceId |
| getDeviceData | address device_id, uin |
| getDeviceTimestamps | address device_id |
| setDeviceData | address deviceId, strir |
| kill | |
| registerDevice | address deviceId, strir |

# 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

THE THINGS NETWORK
LOS ANGELES

- 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

@TTN_LA

# THANKS!