

Write and deploy my first dApp part.1

Deploy a local blockchain based on the Ethereum Virtual Machine

Prerequisites

Node.js

Installation

Ganache-cli (new testrpc) simulates a full node behaviour.

```
npm install -g ganache-cli
```

Web3 is a collection of libraries to interact with a local or remote Ethereum node.

```
npm install web3
```

First run

By starting ganache-cli without any option, 10 accounts are created with 100ETH each.

ganache-cli listens on the local port 8585

```
> ~$ ganache-cli
Ganache CLI v6.9.1 (ganache-core: 2.10.2)

Available Accounts
=====
(0) 0x8E351a2193C38FF81CBc546a4EA13db6631C69c0 (100 ETH)
(1) 0x40E5fE6d6B7c3f9B02D948A31fcEA58F03058B4 (100 ETH)
(2) 0x49c293eE596310a2F5bAb7aE3C78526b6996F477 (100 ETH)
(3) 0x5cdC18Db722B82F8A74eD2D3e92A397B1b8eF625 (100 ETH)
(4) 0xa31196eB17a005F89362d4Ca9F0e8269a1Fad04A (100 ETH)
(5) 0xA1192bc5F6bAe154F36d2fFa2124413611353C17 (100 ETH)
(6) 0xF51520e4A8CE3161A7aa688ECA19131939473C5e (100 ETH)
(7) 0xfC3c7c986d13C16ce17745f77170d8E77611339b (100 ETH)
(8) 0x1E0894cd57070Cc3Da0103312Be9C1644C84d402 (100 ETH)
(9) 0xD9135287781c28a3A0314a0FA0e1aCEBf3AfA7ED (100 ETH)

Private Keys
=====
(0) 0x33bb74c0f97bd59336b989cb85c7fa9471cebd06ce278cf44771d35de2ea9ca
(1) 0xb140c6c0d3d0ee400c79c92418b2f15222ee65a401a2f8c746ae67a6f6f2bb84
(2) 0x788ddc428006eccc2159e7b604528908af74709586da0cefbfefebe2e72ed1523
(3) 0xb1552f3796e713e432386224907a3b9deae4195a31b52fa439a236ede165d2ab
(4) 0xd2754d372bef01d1bfaa83adc6e7d88dac4cc2b13163fa6d1ebf246b3cbe88ea
(5) 0x6f111ee64ffe8aad-c3630b93e06a9b43b37df3762baf60529ae33af4ab50981c
(6) 0xaf93ec0deb0ad0766c77728ff14f64da9194dac68ea3d1b69da915e6820a2b7
(7) 0x0f7c2b84f578e9881507f974f7f8b2abf598e03a86b8463e7b0b9450859db737
(8) 0x794125a76f534d6ac13780a15dbee01b015e134fd3ae9698e2d629bf751b4971
(9) 0xc9730e7225113e493dc9f9774cbfb513d16e6917276e75af72722ff8d7b395d4

HD Wallet
=====
Mnemonic:      alone elder much plug sound medal depart clap cost truth equal security
Base HD Path:  m/44'/60'/0'/0/{account_index}

Gas Price
=====
20000000000

Gas Price
=====
20000000000

Gas Limit
=====
6721975

Call Gas Limit
=====
9007199254740991

Listening on 127.0.0.1:8545
```


Interactions with ganache-cli

Keep the terminal with ganache-cli running open.

Open a second terminal and run node. Initialise web3 with the following

```
var Web3=require('web3');  
var web3= new Web3('http://localhost:8545');
```

And display the accounts

```
web3.eth.getAccounts(console.log);
```

```
> ~$ node  
Welcome to Node.js v12.12.0.  
Type ".help" for more information.  
> var Web3=require('web3');  
undefined  
> var web3= new Web3('http://localhost:8545');  
undefined  
> web3.eth.getAccounts(console.log);  
Promise { <pending> }  
> null [  
  '0x43c97D3063d4ecAf6139EbAE1ac6c2F332e835a5',  
  '0xF8cC988b81308F6F2d21b18B10772bc5fCE30EEf'  
]
```

While displaying the accounts with getAccounts, in the other terminal you should see the following

```
Call Gas Limit  
=====  
9007199254740991  
  
Listening on 127.0.0.1:8545  
eth_accounts
```

Get the balance of an account

```
web3.eth.getBalance("0x00ae1858ea41f5667cda17c7915c2f289c4ee819").  
then(console.log);
```

```
> web3.eth.getBalance("0xF8cC988b81308F6F2d21b18B10772bc5fCE30EEf").then(console.log);  
Promise { <pending> }  
> 9000000000000000000000
```

In the other terminal you should see the following

```
Call Gas Limit  
=====  
9007199254740991  
  
Listening on 127.0.0.1:8545  
eth_accounts  
eth_getBalance
```

We create a transaction from one account to the other of 1000000000000000000 WEI with

```
web3.eth.sendTransaction({from :  
'0xF8cC988b81308F6F2d21b18B10772bc5fCE30EEf',  
to : '0x43c97d3063d4ecaf6139ebae1ac6c2f332e835a5',  
value: '1000000000000000000'})
```

```
> web3.eth.sendTransaction({from : '0xF8cC988b81308F6F2d21b18B10772bc5fCE30EEf',  
to : '0x43c97d3063d4ecaf6139ebae1ac6c2f332e835a5', value: '1000000000000000000'});  
Promise {  
  <pending>,  
  _events: Events <[Object: null prototype] {}> {},  
  emit: [Function: emit],  
  on: [Function: on],  
  once: [Function: once],  
  off: [Function: removeListener],  
  listeners: [Function: listeners],  
  addListener: [Function: on],  
  removeListener: [Function: removeListener],  
  removeAllListeners: [Function: removeAllListeners]  
}
```

And we check at which block we are with

```
web3.eth.getBlockNumber().then(console.log);
```

```
> web3.eth.getBlockNumber().then(console.log);  
Promise { <pending> }  
> 1
```

Meanwhile, in the other terminal

```
eth_sendTransaction  
  
Transaction: 0x05a1ae391a7c6400b2696beeb7e477e  
Gas usage: 21000  
Block Number: 1  
Block Time: Mon Jan 01 1970 00:00:01 GMT+0000  
  
eth_getTransactionReceipt  
eth_blockNumber
```

We can check the new amount in our account

```
> web3.eth.getBalance("0xF8cC988b81308F6F2d21b18B10772bc5fCE30EEf").then(console.log);  
Promise { <pending> }  
> 8989999580000000000000
```

It is easier for the different tests to put all the commands in a file and run it with
`node file.js`

Try different tests, for example:

- empty an account with a while loop;
- is there is a delay to respect between two transactions authored by a single account;
- with a several accounts, can we slow down ganache by performing many transactions simultaneously.

Truffle

Download the truffle environment to deploy smart-contracts written in Solidity.
Ganache-cli must be running besides.

```
npm install -g truffle
```

Create an empty directory and run

```
truffle init
```

Truffle creates a project, with configuration files and folders for the different contracts written in Solidity.

```
> truffle-smartcontract$ ls
build          migrations    truffle-config.js
contracts     test         truffle.js
> truffle-smartcontract$ ls contracts
Migrations.sol Token.sol
```

Make sure to use the last Solidity version in your .sol files, with the header
`pragma solidity ^0.5.8;`

The commands

```
truffle compile
compiles your code
```

```
truffle migrate
deploys the contracts to the local blockchain.
```