Manual : Ethereum

OS : Ubuntu 16.04 LTS

Requirement : Truffle IDE, Testrpc

Installing npm and nodeJS without sudo :

```
echo 'export
PATH=$HOME/local/bin:$PATH' >>
~/.bashrc
. ~/.bashrc
mkdir ~/local
mkdir ~/node-latest-install
cd ~/node-latest-install
curl
http://nodejs.org/dist/node-latest.tar.g
z | tar xz --strip-components=1
./configure --prefix=~/local
make install # ok, fine, this step
probably took more than 10 mins
curl https://www.npmjs.org/install.sh |
sh
```

Source: https://gist.github.com/isaacs/579814 #

Atom – editing environment

req: git

```
sudo apt-get update
sudo apt-get install git
```

```
sudo add-apt-repository ppa:webupd8team/atom
sudo apt-get update
sudo apt-get install atom
```

Source: https://codeforgeek.com/2014/09/install-atom-editor-ubuntu-14-04/

Ethereum Test RPC for developing environment :

Req: npm and node.js

```
npm install -g ethereumjs-testrpc
```

Source: https://github.com/ethereumjs/testrpc

Truffle framework :

req : npm and node.js v5+

npm install -g truffle

Source: http://truffle.readthedocs.io/en/latest/getting_started/installation/

Install Solidity

npm install solc

source : http://solidity.readthedocs.io/en/latest/installing-solidity.html

Install Ethereum :

sudo apt-get install software-properties-common
sudo add-apt-repository -y ppa:ethereum/ethereum

sudo add-apt-repository -y ppa:ethereum/ethereum-dev

sudo apt-get update

sudo apt-get install ethereum

source : https://github.com/ethereum/go-ethereum/wiki/Installation-Instructions-for-Ubuntu

Commands to sync with the modern testnet network :

geth --testnet --rpc --unlock "0" --rpcaddr=127.0.0.1 --rpccorsdomain="" --rpcport=8545*
This will connect to the modern testnet : replica of ethereum blockcahin for testing and development purpose, at port 8545.

To attach with a JavaScript console : *geth --testnet attach http://127.0.0.1:8545*

To create a new account in the test network: *geth --testnet account new*

Enter the passphrase, for the account. Allow the blockchain to sync to the latest block number. It can be found from : testnet, it is the modern testnet interface, here you can view your account, your ether balance and all the transaction details, contracts creation and block details.

To work with Truffle:

In the application directory : *cd myproject*

use Command: *truffle init*, to initialize the truffle project.

Once completed, you'll now have a project structure with the following items:

app/ - directory where your application files go by default. This includes recommended folders for Javascript files and stylesheets, but you have free reign over how this folder is used, if at all.

- ❏ contracts/ - directory where Truffle expects to find solidity contracts.
- ❏ migrations/ - directory to place scriptable deployment files.
- ❏ test/ - location of test files for testing your application and contracts.
- ❏ truffle.js - your main Truffle configuration file.

After the contract and frontend is created :

Use *truffle compile* to compile the source. Truffle will compile only the contracts that have been changed since the last compile, to reduce any unnecessarily compilation. If you'd like to override this behavior, run the above command with the*--compile-all* option.

To build your frontend, simply run:

$ truffle build

build artifacts are saved within the ./build directory, along side compiled deployed contract artifacts in ./build/contracts

To deploy the application in the blockchain : *$ truffle deploy*

To connect to the truffle console : *$ truffle console*

To serve the built app from http://localhost:8080 , rebuilding and redeploying changes as needed :
$ truffle serve