Truffe Development Tool

Truffle - It is a smart contract development tool used for compiling, testing and deploying in testnets & Mainnets.

1.) truffle init

- contracts [solidity code files]
- migrations [deployments scripts, how to initialize & deploy]
- test [unit test with mocha and chai]
- build

(this folder will be generated when smart contract is deployed & it creates the config file (json) which is used to deploy in mainnets /testnets.)

2.) Truffle deployment - Create a seperate file for your respective contract in migrations folder (numbering is important to truffle which deploys in order).

truffle-migrate - runs all migrations files.

3.) truffle develop - sets up a local ethereum nodes, few account address, & some private keys for development.

- interact with contracts via console using instance.

```
let instance = await HelloWorld.deployed()
instance.Hello()
```

__Note__ :- Do truffle develop before using truffle console.

4.) We can pass the constructor arguments in the migrations file.

Ex:- deployer.deploy(TestContract, "Hello")

5.) After contract changes simply migrate doesn't work, 'it doesn't deploy new change' instead migrate --reset command helps to rebuild the contract with new changes & deploy.

6.) if a 'payable' function needs to be called & Let's say we should send some ether. web3 provides utility method to pass the value.

Ex:- instance.setMessage("abcd", {value: web3.utils.toWei("2", ether)})

7.) If you want to try with different account

(—, {value: web3..., from: accounts[2]})

8.) How to use 2 Contracts in a single migration file

```
deployer.deploy(HelloWorld, "hi").then
((async) => {
    let instance = HelloWorld.deployed();
    let message = instance.message();
}); return deployer.deploy(Contract2, message);
```

9.) How to deploy contracts on testnet/mainnet?

1.) Get moralis.io test node link (→ ropsten n/w)

2.) create mnemonic using npx

$\boxed{\text{npx mnemonics}}$

3.) create secrets.json & add mnemonic

4.) Uncomment ropsten n/w config
add node link & secret.

5.) truffle console --network ropsten
truffle migrate