**Ethereum Tutorial @ EPISTAR**

Ruby Tseng ( 曾敏涵 )

July 30, 2018

本教程為根據實際於Windows 10 64bit的環境下，建置以太坊私有鏈( Private Ethereum network )為目標所撰寫。

**1. Install**

**1.1 Geth 1.8.12 for Windows 64-bit**

<https://ethereum.github.io/go-ethereum/downloads/>

$ geth version

*Geth*

*Version: 1.8.12-stable*

*Git Commit: 37685930d953bcbe023f9bc65b135a8d8b8f1488*

*Architecture: amd64*

*Protocol Versions: [63 62]*

*Network Id: 1*

*Go Version: go1.10.3*

*Operating System: windows*

*GOPATH=C:\Users\22404*

*GOROOT=C:\tools\go*

**1.2 Solidity v0.4.24**

<https://github.com/ethereum/solidity/releases>

$ solc --version

*solc, the solidity compiler commandline interface*

*Version: 0.4.24+commit.6ae8fb59.Windows.msvc*

**2. Steps**

**2.1 Create Blockchain Account for Private Network**

$ geth account new --datadir .eth-test

*Address: { abb67ca6a4549dc15fa5ad95ca85bef49c277f33 }*

*Pass:730*

**2.2 Create Private Blockchain Network**

$ geth --datadir .eth-test init genesis.json

檔名:genesis.json

{

"config": {

"chainId": 88888,

"homesteadBlock": 0,

"eip155Block": 0,

"eip158Block": 0

},

"alloc" : {" abb67ca6a4549dc15fa5ad95ca85bef49c277f33 ": { "balance":"40000000000000000"}},

"coinbase" : "0x0000000000000000000000000000000000000000",

"difficulty" : "0x20000",

"extraData" : "",

"gasLimit" : "0x2fefd8",

"nonce" : "0x0000000000000042",

"mixhash" : "0x0000000000000000000000000000000000000000000000000000000000000000",

"parentHash" : "0x0000000000000000000000000000000000000000000000000000000000000000",

"timestamp" : "0x00"

}

**2.3 Compile Smart Contract**

**2.3.1 Smart Contract Source Code – the greeter**

檔名:HelloEPISTAR.sol

pragma solidity ^0.4.8;

contract mortal {

/\* Define variable owner of the type address\*/

address owner;

/\* this function is executed at initialization and sets the owner of the contract \*/

function mortal() { owner = msg.sender; }

/\* Function to recover the funds on the contract \*/

function kill() { if (msg.sender == owner) suicide(owner); }

}

contract greeter is mortal {

/\* define variable greeting of the type string \*/

string greeting;

/\* this runs when the contract is executed \*/

function greeter(string \_greeting) public {

greeting = \_greeting;

}

/\* main function \*/

function greet() constant returns (string) {

return greeting;

}

}

**2.3.2 Compile HelloEPISTAR**

$ solc --bin HelloEPISTAR.sol > HelloEPISTAR.bin

$ solc --abi HelloEPISTAR.sol > HelloEPISTAR.abi

$ solc --abi HelloEPISTAR.sol > HelloEPISTAR.interface

$ solc --combined-json abi,bin,interface HelloEPISTAR.sol > HelloEPISTAR.js

**edit HelloEPISTAR.js**

add **var greeterCompiled =** in the begining



**2.4 Launch Ethereum Node Process**

$ geth --mine --minerthreads=1 --datadir .eth-test --networkid 88888 --nodiscover --maxpeers 0 console 2>> .eth-test.log --dev.period 1

*Welcome to the Geth JavaScript console!*

*instance: Geth/v1.8.12-stable-37685930/windows-amd64/go1.10.3*

*coinbase: 0x3d07c09d41908f8b064f6e4480ad725e25ada63f*

*at block: 0 (Thu, 01 Jan 1970 08:00:00 CST)*

*datadir: C:\Users\22404\BC\.eth-private*

*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*

*>*

**2.5 Submit Smart Contract**

**2.5.1 Unlock Account to submit SM**

> web3.fromWei(eth.accounts[0],"ether")

> primary = eth.accounts[0]

> personal.unlockAccount(primary)

*Unlock account 0xabb67ca6a4549dc15fa5ad95ca85bef49c277f33*

*Passphrase:*

*true*

*>*

**2.5.2 Submit Compiled Smart Contract**

> loadScript("HelloEPISTAR.js");

> var \_greeting = "Hello EPISTAR!" ;

> var greeterContract = web3.eth.contract(JSON.parse(greeterCompiled.contracts["HelloEPISTAR.sol:greeter"].abi));

> var greeter = greeterContract.new(\_greeting, {from: eth.accounts[0], data:"0x" + greeterCompiled.contracts["HelloEPISTAR.sol:greeter"].bin, gas: 1000000},

function(e, contract){

if(!e) {

if(!contract.address) {

console.log("Contract transaction send: TransactionHash: " +

contract.transactionHash + " waiting to be mined...");

} else {

console.log("Contract mined! Address: " + contract.address);

}

}

})

*Contract transaction send: TransactionHash: 0x77ef75160561eee412d53434b2dd30652105b4e7ca431372cf857bc9a3efbaf6 waiting to be mined...*

*undefined*

*> Contract mined! Address: 0x1d5e79eeb75d88de18ff941d2741a06b2cff04f4*

**2.6 Execute Smart Contract**

> greeter.greet()

"Hello EPISTAR!"

**2.7 Miner Address**

查看當前節點下，是否有帳戶存在

>personal.listAccounts

*["0xabb67ca6a4549dc15fa5ad95ca85bef49c277f33"]*

執行設置miner地址

>miner.setEtherbase(eth.accounts[0])

*True*

查詢餘額

> web3.fromWei(eth.getBalance(eth.coinbase));

*5755.04*

查詢當前的區塊數

> eth.blockNumber

*8040.04*