**Raspichain**

Pasos a seguir para la creación de un blockchain de Raspberrys Pi:

1) Formatear tarjeta SD.

2) Copiar el contenido de la carpeta NOOBS en la tarjeta SD. Podemos encontrarlo en la siguiente dirección si aún no disponemos de los archivos:

<https://www.raspberrypi.org/downloads/noobs/>

3) Insertar la tarjeta en la Raspi, conectarla a la red y a un monitor, ratón y teclado, para que la labor sea más sencilla, e instalar Raspbian. Esperamos unos minutos a que acabe.

4) Configuramos las opciones que se nos dan al iniciar. El usuario y contraseña son:

Usuario: ***pi***

Contraseña: ***1234***

5) Activamos SSH accediendo, con el comando *sudo su*, a la configuración de la raspi escribiendo lo siguiente:

*$ raspi-config*

Y desde el menú que aparecerá, pulsamos las siguientes opciones:

*Interfacing options - P2 SSH - Sí*.

Esto nos permitirá poder controlar a las raspis a través de Putty.

6) Cambiamos en *Preferencias - Configuración* el nombre del host a *raspi1*, por ej.

7) Instalamos geth escribiendo secuencialmente las siguientes sentencias en la consola:

*$ sudo apt-get update*

*$ sudo apt-get dist-upgrade*

*$ sudo apt-get install git golang libgmp3-dev*

*$ mkdir src*

*$ cd src*

*$ git clone -b release/1.7 https://github.com/ethereum/go-ethereum.git*

*$ cd go-ethereum*

*$ make*

*$ sudo cp build/bin/geth /usr/local/bin/*

Windows: instalamos geth usando el link de descarga que encontramos en la siguiente página:

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

8) Creamos nueva cuenta en la red blockchain, a la que llamaremos *raspichain* (podemos elegir el nombre que queramos):

*$ geth --datadir .raspichain account new*

Windows (el path puede ser el que queramos, "*C:\ETH\raspichain*" por ejemplo):

*geth --datadir "C:\ETH\raspichain" account new*

9) Creamos un archivo génesis, que llamaremos *raspichain.json*, parecido a este y lo guardamos en "*/home/pi*" (el mismo directorio donde se encuentra la carpeta *.raspichain*), o en "*C:\ETH\configs*", para Windows:

*{*

*"config": {*

*"chainId": 555,*

*"homesteadBlock": 0,*

*"eip155Block": 0,*

*"eip158Block": 0*

*},*

*"difficulty": "200000",*

*"gasLimit": "2100000",*

*"alloc": {*

*"9fca95a77de4c3dc1f2ac048913afecff23f8cde": { "balance": "50000000000000000000" },*

*"5401a25cc2797e75b874c8ebcf4fccc24aa32870": { "balance": "80000000000000000000" },*

*"904894eba3f8dd7fcb7634830f75b807cfbd7759": { "balance": "30000000000000000000" }*

*}*

*}*

En *alloc* especificamos las cuentas de los nodos que queramos que tengan algo de saldo y dicha cantidad.

10) Inicializamos nuestra blockchain:

*$ geth --datadir .raspichain init raspichain.json*

Windows:

*geth --datadir "C:\ETH\raspichain" init "C:\ETH\configs\raspichain.json"*

11) Inicializamos el nodo (para cada nodo usaremos un parámetro *identity* diferente; y si alguno de los nodos está en un pc con Windows, cambiaremos su parámetro *datadir*):

*$ geth --identity raspi1 --rpc --rpcport 8080 --rpccorsdomain "\*" --datadir .raspichain --port 30303 --nodiscover --rpcapi "db,eth,net,web3" --networkid 555 console*

*$ geth --identity raspi2 --rpc --rpcport 8080 --rpccorsdomain "\*" --datadir .raspichain --port 30303 --nodiscover --rpcapi "db,eth,net,web3" --networkid 555 console*

*$ geth --identity raspi3 --rpc --rpcport 8080 --rpccorsdomain "\*" --datadir .raspichain --port 30303 --nodiscover --rpcapi "db,eth,net,web3" --networkid 555 console*

*$ geth --identity raspi4 --rpc --rpcport 8080 --rpccorsdomain "\*" --datadir .raspichain --port 30303 --nodiscover --rpcapi "db,eth,net,web3" --networkid 555 console*

Windows:

*geth --identity pcmurphy --rpc --rpcport 8080 --rpccorsdomain "\*" --datadir "C:\ETH\raspichain" --port 30303 --nodiscover --rpcapi "db,eth,net,web3" --networkid 555 console*

12) Una vez inicializada, podemos ver la dirección de la cuenta y su saldo desde la consola:

*> eth.accounts*

*> balance = web3.fromWei(eth.getBalance(eth.accounts[0]), "ether");*

13) Creamos el resto de nodos de la misma forma, con **EXACTAMENTE EL MISMO ARCHIVO GÉNESIS QUE EL PRIMERO**, y los inicializamos cambiando la identidad (o el *datadir* si es necesario).

14) En la consola escribimos lo siguiente para ver la URL de inodo de cada raspi:

*> admin.nodeInfo.enode*

15) Con esa información escribimos en un archivo (al que llamaremos *static-nodes.json* y que guardaremos en la carpeta *.gisaichain*, en todas las raspis, o en "*C:\ETH\raspichain*", en el PC) la información sobre los inodos, añadiendo además la dirección ip de cada uno de la siguiente forma:

*[*

*"enode://2c773c0c163567c9fc70042c74cf5be8825f16b30acbb8a4095abeef12726ff98c806fba41879cef3133fc10b8d766e8eae312eb38ae8ae6fdcc2a57ae2b5239@169.254.18.239:30303",*

*"enode://e8087674a396156d1968e9a4ce1513da4db6deb422730c9751bd5db44ca6db933b11efab48a3ee8b141e916150ffc18c18d0af5c0ca6e61928f47f4a256adf99@169.254.23.143:30303",*

*"enode://213a1c0aba999be139e45960fc453f569b335a31e1e07c4381455836699f4f9553a05145f8bcc1bb8fa1d83a63e82d3dd3721afff68fdadc05bc89cb2890f9e6@169.254.124.255:30303",*

*"enode://c84ac33b7eb062c57759281c3dca2be3d26b5a36cdf57b710d582802e9c9fc5c9f3c37380bd79e5984ed86b2c635030b52052efc31f379827917a50a9000e74c@169.254.115.106:30303",*

*"enode://f7eceff5b191882a1ae283d8870034562d1903b3dd23d193921b4d357cdeffc890060b92f7ce31e440a3efc954e9375d82e8255ca518e4d1294b245cf7498eed@169.254.173.233:30303"*

*]*

16) Cerramos las consolas de las raspis con *Ctrl + D*, inicializamos de nuevo la cadena y escribimos lo siguiente para comprobar que los pares (*peers*) se ven entre ellos:

*> admin.peers*

17) Comprobamos el saldo de cada nodo:

*> web3.fromWei(eth.getBalance(eth.coinbase), "ether")*

18) Para traspasar fondos a, por ej, raspi4, escribimos lo siguiente en la consola del nodo desde el cual queremos traspasar los fondos:

*> var pi4 = "0x34ec9c281c173a6a6853681253f7bcfd1b294677";*

*> personal.unlockAccount(eth.coinbase, "")*

*> eth.sendTransaction({from: eth.coinbase, to: pi4, value: web3.toWei(15, "ether")})*

Para actualizar el resultado, probablemente tengamos que salir de la consola e iniciar de nuevo la cadena.

19) Escribimos en la consola del nodo que va a minar la siguiente sentencia para que se produzca la transacción anterior (sincronización y minado):

*> miner.start()*

20) Pasados unos instantes podemos comprobar que se ha realizado la transacción mirando el saldo de los nodos. El nodo minador tendrá ether extra debido a la labor que está realizando.

Cuando queramos detener el minado, escribiremos:

*> miner.stop()*

**Añadir más nodos a la red**

Si queremos añadir otra raspberry o un pc a la red, seguiremos los mismos pasos anteriores como si fuera uno más, pero teniendo en cuenta tres aspectos esenciales:

1) Tenemos que usar el **MISMO archivo génesis** que en el resto de nodos, sin cambiar absolutamente nada del archivo (paso 9).

2) Al inicializar el nodo hemos de usar una nueva identidad para él (paso 11).

3) Hemos de actualizar el archivo *static-nodes.json* en todos los demás nodos, además de añadirlo en el nuevo (paso 15).