Rapport de BlockChain

Configuration d’une Blockchain privé Ethereum

Azzeddine LAHMAR – A3MSI

Installation de Etherum sur Ubuntu et procédure de l'étape de création des comptes

Sur une machine Ubuntu (Ubuntu 18) nous allons installer Ethereum. Les commandessuivantes vont permettre d'installer les paquets nécessaires :

sudo apt-get install software-properties-common sudo

add-apt-repository -y ppa:ethereum/ethereumsudo

apt-get update

sudo apt-get install ethereum

sudo apt-get install git cmake libleveldb-dev libjsoncpp-dev libboost-all-dev libgmp-dev libreadline-dev libcurl4-gnutls-dev ocl-icd-libopencl1 opencl-headersmesa-common-dev libmicrohttpd-dev build-essential -y

sudo apt-get install libjsonrpccpp-dev -y

Après installation, nous pouvons créer des comptes.

geth --datadir data account new

```
INFO [11-16|17:29:03.660] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:

Your new key was generated

Public address of the key:   0x7B0838Ca571B203DB7aecE4917DCEA824EBb9F1E
Path of the secret key file: data/keystore/UTC--2022-11-16T16-29-11.699059510Z--7b0838ca571b203db7aece4917dcea824ebb9f1e

- You can share your public address with anyone. Others need it to interact with you.
- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!

root@TPBlockchain:/home/azzeddine/private-ethereum2# geth --datadir data account2 new
invalid command: "account2"
root@TPBlockchain:/home/azzeddine/private-ethereum2# geth --datadir data account new2
No help topic for 'new2'
root@TPBlockchain:/home/azzeddine/private-ethereum2# geth --datadir data2 account new
INFO [11-16|17:29:56.047] Maximum peer count                       ETH=50 LES=0 total=50
INFO [11-16|17:29:56.050] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:

Your new key was generated

Public address of the key:   0x7Bfc8363B7D56390288f6A80c5731cBea49B876f
Path of the secret key file: data2/keystore/UTC--2022-11-16T16-30-09.394508997Z--7bfc8363b7d56390288f6a80c5731cbea49b876f

- You can share your public address with anyone. Others need it to interact with you.
- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!

root@TPBlockchain:/home/azzeddine/private-ethereum2#
```

Configuration de Bootnode, côté client

Après avoir procédé à l'étape de création des comptes, la suite impliquait la création d'un fichier genesis.json contenant des données identiques à celui du serveur :

```
{
"config":  {
"chainId":
28112020,
"homesteadBlock": 0,
"eip150Block": 0,
"eip155Block": 0,
"eip158Block": 0,
"byzantiumBlock": 0,
"constantinopleBlock": 0,
"petersburg
Block": 0,
"ethash": {}
 },
 "difficulty": "1",
 "gasLimit": "8000000",
 "alloc": { "b3d9fac3f9d338af629aa7ff1880d7cce06fdb60": {
"balance":"30000000000000000000" }
 }
}
```

chainId : L'identifiant unique de la blockchain ;

homesteadBlock : Le block ou la mise à jour du Homestead devient actif ;

eip155Block : Le block ou le hard fork EIP150 devient actif ;

eip158Block : Le block ou le hard fork EIP155 devient actif ;

byzantiumBlock : Le block ou le Byzantium hard fork devient actif ;

ethash : L'algorithme de proof-of-work utilisé par Ethereum ;

difficulty : Représente la difficulté du block Genesis ;

gasLimit : Quantité maximale de gaz qui peut être utilisé lors d'une transaction sur le réseau ;

alloc : Allocation initiale d'éther aux comptes du réseau Ethereum.

Après initialisation du fichier Genesis, nous pouvons procéder à l'instanciation de ce dernier grâce à la commande

geth init --datadir data genesis.json

```
INFO [11-16|16:49:48.918] Successfully wrote genesis state         database=chai
ndata hash=a406cd..5e5325
INFO [11-16|16:49:48.919] Allocated cache and file handles         database=/hom
e/azzeddine/private-ethereum2/data/geth/lightchaindata cache=16.00MiB handles=16
INFO [11-16|16:49:48.937] Opened ancient database                  database=/hom
e/azzeddine/private-ethereum2/data/geth/lightchaindata/ancient/chain readonly=fa
lse
INFO [11-16|16:49:48.937] Writing custom genesis block
INFO [11-16|16:49:48.938] Persisted trie from memory database      nodes=3 size=
409.00B time="46.002µs" gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.
00B
INFO [11-16|16:49:48.939] Successfully wrote genesis state         database=ligh
tchaindata hash=a406cd..5e5325
```

Configuration du Bootnode

Afin qu'un client puisse se connecter au serveur grâce aux configurations Bootnode apportées sur ce dernier, le client doit lui-même configuré un Bootnode, voici ci-dessous les étapes :

bootnode --genkey=boot.key
bootnode --nodekey=boot.key

```
root@TPBlockchain:/home/azzeddine/private-ethereum2# bootnode --genkey=boot.key
root@TPBlockchain:/home/azzeddine/private-ethereum2# bootnode --node

enode://331e7a87d5ebe19f85bd08282aa6e53ea009790d94973472538d4b48e40c68b8a30b3d57052f42a3b4b91d9f74f27cfe5150a0c363
Note: you're using cmd/bootnode, a developer tool.
We recommend using a regular node as bootstrap node for production deployments.
INFO [11-16|16:53:26.331] New local node record                    seq=1,668,614,006,329 id=e35f5646f8bcf93c ip=<n
```

Après exécution,on obtient le lien vers le bootnode. Dans notre cas,il est défini notre adresse locale, 127.0.0.1 sur le port 3030.
On peut vérifier que l'on possède bien la même adresse comme identifiant avec la commande suivante :

bootnode --nodekey=boot.key –writeaddress

```
root@TPBlockchain:/home/azzeddine/private-ethereum2# bootnode --nodekey=boot.key --writeaddress
331e7a87d5ebe19f85bd08282aa6e53ea009790d94973472538d4b48e40c68b8a30b3d57052f42a3b4b91d9f74f27cfe5150a0c363cd248689a221a2c2ba223b
```

Démarrage du noeud du serveur côté client

Grâce à la configurattion du nœud de démarrage avec Bootnode, le serveur est à présent accessible depuis des machines clientes.

En tant que client, les prérequis nécessaires pour la connexion sont remplies, à savoir :

- Création d'un compte
- Configuration d'un bootnode côté client
- Récupération et instanciation du fichier Genesis.json du serveur

geth --networkid 202201 --datadir data--bootnodes enode://98ae4fefba9420f1a49a84c373fc9cfcf10ed71d6bbbbd48d64520121bec5dd469a51dde5d37c5e495ccb544d2adcca177e204d17dc2d0ca3169cda21a040785@64.227.65.43:30303 console

Les données en rouges sont des infos spécifiques au serveur configuré par notre camarade

La donnée en bleu et le nom du 1er compte crée plus tôt côté client, à savoir « data ».

Une fois la commande saisie, une synchronisation a lieu :

```
root@TPBlockchain:/home/azzeddine/private-ethereum3# geth --networkid 202201 --datadir data --bootnodes enode://98ae4fefba9420f1a49a84c373fc9cfcf10ed71d6bbbbd48d645201
ca177e204d17dc2d0ca3169cda21a040785@64.227.65.43:30303  console
INFO [11-17|12:02:28.879] Maximum peer count                       ETH=50 LES=0 total=50
INFO [11-17|12:02:28.884] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
```

En saisissant la commande eth.accounts, nous obtenons l'adresse du compte « data » synchronisé avec le réseau :

```
To exit, press ctrl-d or type exit
> eth.accounts
["0xc1e65c54816bee63cd41c37b72c20c34b451ee61"]
```

Nous pouvons par ailleurs vérifier la synchronisation avec nos camarada en vérifiant la valeur Hash d'un certain block.
Pour le block 0, la commande à exécuter sur plusieurs machines pour vérification :

eth.getBlock(0)

Finalement, nous pourrons bien contribuer au réseau avec le processus de mining, réalisable via la commande suivante :

miner.start(1)