

Développeur blockchain

Introduction

Geth et les clients Eth



cyril@alyra.fr / benjamin.brucher@alyra.fr

Promo Buterin

Topo du live

- Les clients eth
- Séparation en deux groupes
 - **Lancer sa blockchain locale**
 - **Lancer une blockchain distante**
- se retrouver et mettre en commun

Présentation des formateurs nous accompagnant!

Les clients BC

- Communiquer avec la blockchain.
- Etablis une communication P2P avec les autres clients
- Permet de signer une transaction, la diffuser, miner, deployer ou interagir avec un smart contract (sur eth)
- Eth est open source et permet donc d'utiliser le client qu'on veut, de créer avec le langage qu'on veut... tout en respectant les devoirs techniques

<https://clientdiversity.org/>

Geth

Geth est un client ethereum développé en Go
Go ethereum, geth

Le client le plus utilisé sur la blockchain (62% des
nœuds)

Robuste, même si la mise à jour london avec une
ancienne version de geth a créé un fork (Tout
comme le passage à PoS avec EthPoW, non lié à
geth)

Fonctionnement

Permet d'opérer comme client de plusieurs manières différentes:

[Full, archive ou light node](#)

Full node:

Récupération de tous les blocks (header et transactions) de la blockchain. Récupère aussi l'état à l'instant T, et des 128 derniers blocs (variable selon les réseaux). Est capable de faire passer toutes les transactions de l'histoire pour avoir des states intermédiaires (1.1 tera sur eth - 01/2023)

Archive node:

Récupère tout: tous les états de la blockchain à chaque instant. (13 tera sur Eth - 01/2023)

Fonctionnement

Light node:

Récupère uniquement les headers de tous les blocks: participent à la sécurisation du réseau. Ils permettent aussi d'interroger des full node pour certaines taches: des call vers des balances, verifier des evenements, si une transaction a été effectuée

RPC:

[Doc ethereum.org](https://docs.ethereum.org)
[interroger un RPC](#)

Fonctionnement

Il peut être client ethereum du main net
Ou permettre de créer sa propre blockchain basée
sur eth, avec un EVM, un consensus...
Que ce soit en local, ou sur serveur
Les gens pourront donc s'y connecter,
-> blockchain privée

Séparation

On fait trois salles différentes.

Rejoignent la salle 1 ceux qui n'ont pas pu faire le cours sur geth

Rejoignent salle 2 et 3 ceux qui l'ont fait et veulent aller plus loin

But du tp pour ceux qui vont en 2/3:

Suivre une marche à suivre pour créer une blockchain distante auquel vous vous connectez, sur des serveurs fournis

La marche à suivre est dispo dans Slides : 5 - Geth SSH et Sshpass.txt

Installons Geth

[Doc officielle pour install](#)

```
sudo apt-get install software-properties-common
```

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

```
sudo apt-get update
```

```
sudo apt-get install ethereum
```

Première manip

On se place dans un répertoire, et on crée un compte ethereum

```
mkdir gethTest
```

```
cd gethTest
```

```
geth account new --datadir .
```

On va récupérer la première adresse créée

On va créer notre block génésis

Grâce à l'éditeur Puppeth



Image 01

```
What would you like to do? (default = stats)
 1. Show network stats
 2. Configure new genesis
 3. Track new remote server
 4. Deploy network components
> 2

What would you like to do? (default = create)
 1. Create new genesis from scratch
 2. Import already existing genesis
> 1

Which consensus engine to use? (default = clique)
 1. Ethash - proof-of-work
 2. Clique - proof-of-authority
> 2

How many seconds should blocks take? (default = 15)
> 10
```

Puppeth

OSSECOUR

Puppeth n'existe plus.

C'est dommage, c'était pratique pour générer notre fichier genesis

De fait, nous devons le faire nous meme. Des exemples existent ici:

<https://geth.ethereum.org/docs/fundamentals/private-network>

Initialisation

Initialisation du noeud avec fichier genesis:

```
geth --datadir . init NOM.json
```

Création du fichier avec le mot de passe du sealer :

```
echo 'opop' > pwd.txt
```

Execution

On lance la bc locale de cette manière:

```
geth --datadir . --syncmode 'full' --networkid "9999" --port "30303" --  
http --http.addr '127.0.0.1' --http.port "8545" --http.api  
'personal,eth,net,web3,txpool,miner,admin,clique' --nodiscover --  
mine --miner.gaslimit '9000000000000' --allow-insecure-unlock --  
unlock "VOTREADDRESSE" --password pwd.txt
```

Interaction

Lancement de la console

```
geth attach Adresse
```

Des premières manipulations

```
eth.accounts
```

```
eth.getBalance('ADDRESSE1')
```

```
eth.sendTransaction({from : 'ADDRESSE1', to : 'ADDRESSE2', value:  
'1000000000000000000000000'})
```

```
eth.getTransaction('ADDRESSESTRANSAC')
```

```
eth.getBlock('76')
```

16/05/2023

Demain on commence solidity!

Merci de votre attention!

