



## Création et configuration d'une Blockchain Privée Ethereum

**Je suis client.**

### I. Installation d'Ethereum sur Ubuntu (Linux recommandé)

Pour créer un réseau privé Ethereum, nous aurons besoin de créer au moins 2 nœuds du réseau qui vont communiquer ensemble et stocker la blockchain.

- Nous devons tout d'abord installer Ethereum dans notre système.

On utilise les commandes suivantes pour installer **geth et ethereum**:

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

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

```
sudo 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-headers mesa-common-dev libmicrohttpd-dev build-  
essential -y
```

```
sudo apt-get install libjsonrpcpp-dev -y
```

```
File Edit View Search Terminal Help
See "man sudo_root" for details.
malia@malia-VirtualBox:~$ sudo apt-get software-properties-common
[sudo] password for malia:

Sorry, try again.
[sudo] password for malia:
Sorry, try again.
[sudo] password for malia:
sudo: 3 incorrect password attempts
malia@malia-VirtualBox:~$ sudo su
[sudo] password for malia:
root@malia-VirtualBox:/home/malia# sudo apt-get properties-common
E: Invalid operation properties-common
root@malia-VirtualBox:/home/malia# sudo apt-get install properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package properties-common
root@malia-VirtualBox:/home/malia# sudo apt-get install properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package properties-common
root@malia-VirtualBox:/home/malia# sudo apt-get install software-properties-com
mon
Reading package lists... Done
Building dependency tree
```

Ensuite on peut passer à la création du réseau blockchain privé.



## II. Création de comptes pour le réseau privé Ethereum

D'abord on crée un répertoire **private-ethereum**.

Ensuite on crée un premier compte wallet sur le nœud avec la commande suivante :

```
geth --datadir data account new
```

```
Setting up libjsm-pcscpp-dev (0.7.0-1ubuntu2) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for install-info (6.5.0.dfsg.1-2) ...
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...
root@malia-VirtualBox:/home/malia# mkdir private-ethereum
root@malia-VirtualBox:/home/malia# cd private-ethereum
root@malia-VirtualBox:/home/malia/private-ethereum# geth --datadir data account
new
INFO [11-04|11:43:28.003] Maximum peer count                ETH=50 LES=0
total=50
INFO [11-04|11:43:28.004] Smartcard socket not found, disabling err="stat /r
un/pcscd/pcscd.comm: no such file or directory"
Your new account is locked with a password. Please give a password. Do not forg
et this password.
Password:
Repeat password:
Your new key was generated
```

On vérifie la création du compte à partir de cette commande :

```
geth --datadir data account list
```

```
Account #0: {75ede2926bd9f3f1b118eae4baef285d3866c57b} keystore:///home/malia/p
rivate-ethereum/data/keystore/UTC--2022-11-04T10-43-33.538179295Z--75ede2926bd9
f3f1b118eae4baef285d3866c57b
root@malia-VirtualBox:/home/malia/private-ethereum#
```

## III. Création du Genesis File uniquement sur le serveur fourni dans un premier temps !

Je copie le **genesis.json** du serveur. Ce fichier contient les propriétés qui définissent la blockchain.

```
root@malia-VirtualBox:/home/malia/private-etherieum2# ls
data genesis.json
root@malia-VirtualBox:/home/malia/private-etherieum2#
```

Ensuite on instancie le répertoire de données. Le répertoire de données est le répertoire où sont stockées les données liées à la Blockchain. Pour l'instancier, on exécute la commande ci-après :

```
geth init --datadir data genesis.json
```



## La définition de chaque paramètre du genesis bloc :

### **chainId:**

66 pour le réseau principal et 96 pour le réseau de test. Pour être compatible avec les services tiers qui supportent déjà Ethereum, il est préférable de ne pas utiliser l'identifiant de réseau déjà utilisé par l'écologie d'Ethereum. L'identifiant de réseau de test-net doit être distinct de celui de main-net.

### **homesteadBlock:**

Lorsqu'il est défini à 0, signifie que vous utiliserez la version Homestead d'Ethereum. Ceci est attendu, et la configuration du mainnet genesis a également cette valeur à 0.

### **eip155Block:**

Pour aider à prévenir les attaques par rejeu.

### **eip158Block:**

Pour changer la façon dont les clients d'Ethereum traitent les comptes vides. Ce nouveau protocole a commencé à les traiter comme inexistant, économisant ainsi de l'espace sur la blockchain.

### **byzantiumBlock:**

Un codage natif de la blockchain d'Ethereum

**constantinopleBlock et petersburgBlock:** ils servent tout simplement à accroître la capacité du réseau et de réduire la consommation d'énergie du minage.

### **Ethash:**

Il sert pour le minage des blocs.

### **difficulty:**

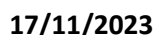
Une valeur scalaire correspondant au niveau de difficulté appliqué lors de la découverte du nonce de ce bloc.

### **gasLimit:**

Une valeur scalaire égale à la limite actuelle, à l'échelle de la chaîne, de la dépense en gaz par bloc.

### **alloc:**

Permet de définir une liste de portefeuilles préremplis.



**Sur cette partie le serveur obtient une URL enode.**

## VI. Démarrage de vos nœuds sur les ordinateurs de vos camarades

Pour lancer le nœud on veillera à lui associer un identifiant réseau **202201**, un **numéro de port** différent de celui du premier nœud et un fichier **genesis.log** qui contient déjà la chaîne.

On utilise la commande suivante :

**geth --networkid 202201 --datadir data --bootnodes <url\_enode> console**

en utilisant l' **url enode du serveur** obtenu à la partie V après minage.

```

To exit, press ctrl-d or type exit
> INFO [11-17|10:38:13.454] New local node record                      seq=1,668,
677,891,699 id=a8ad72e7a88139c9 ip=89.85.250.122 udp=9828 tcp=30303
> newWARN [11-17|10:38:13.790] Snapshot extension registration failed    peer=55d
93f39 err="peer connected on snap without compatible eth support"
> net.listeningINFO [11-17|10:38:21.777] Looking for peers
peercount=1 tried=112 static=0
INFO [11-17|10:38:21.787] Block synchronisation started
INFO [11-17|10:38:24.595] Imported new block headers                  count=192 el
apsed=932.985ms number=192 hash=b51b88..eaa1d0 age=49m38s
INFO [11-17|10:38:24.596] Downloader queue stats                      receiptTasks
=0 blockTasks=0 itemSize=580.008 throttle=8192
INFO [11-17|10:38:24.598] Imported new block receipts                 count=192 el
apsed=1.781ms number=192 hash=b51b88..eaa1d0 age=49m38s size=16.12KiB
INFO [11-17|10:38:24.659] Imported new block headers                  count=192 el
apsed=64.167ms number=384 hash=6ef16b..8af9e8 age=35m45s
INFO [11-17|10:38:24.664] Imported new block receipts                 count=192 el
apsed="934.704µs" number=384 hash=6ef16b..8af9e8 age=35m45s size=16.12KiB
INFO [11-17|10:38:24.810] Imported new block headers                  count=192 el
apsed=81.766ms number=576 hash=fad9f1..d78307 age=18m
INFO [11-17|10:38:24.841] Imported new block receipts                 count=192 el
apsed=28.576ms number=576 hash=fad9f1..d78307 age=18m size=16.12KiB
INFO [11-17|10:38:26.618] Imported new block headers                  count=192 el
apsed=1.589s number=768 hash=9f9f0d..be9dd1 age=1m56s
INFO [11-17|10:38:26.621] Imported new block receipts                 count=143 el
apsed="832.291µs" number=719 hash=f2b747..39094a age=6m30s size=12.01KiB
INFO [11-17|10:38:26.730] Imported new block headers                  count=15 el
apsed=112.137ms number=783 hash=9e6e31..abeb0e
INFO [11-17|10:38:29.829] Imported new block headers                  count=1 el

```

On voit depuis cette capture d'écran qu'on est bien synchronisé au serveur.

**NB : nous pouvons rajouter autant de nœud que nous pouvons.**

Après cette étape, on est dans la console geth.

```

> net.listening
true
> INFO [11-17|10:02:24.165] Looking for peers                      peercount=
1 tried=120 static=0
INFO [11-17|10:02:37.174] Looking for peers                      peercount=0
tried=119 static=0
INFO [11-17|10:02:47.301] Looking for peers                      peercount=1
tried=86 static=0
INFO [11-17|10:02:57.315] Looking for peers                      peercount=1
tried=141 static=0
INFO [11-17|10:03:07.669] Looking for peers                      peercount=1

```

[illegible]

Ensuite vérifier si vous avez bien créé un compte [0] avec la commande suivante via la console sur le client avant de commencer à votre tour le processus de mining

```
To exit, press ctrl-d or type exit
> eth.accounts
["0x14663bc92027f7e10d4544d4761c4277b15a1b56"]
> INFO [11-17|11:34:05.491] New local node record                               seq=1,668,
677,891,701 id=a8ad72e7a88139c9 ip=89.85.250.122 udp=9785 tcp=30303
```

Nous pouvons maintenant également participer au réseau en démarrant le processus de mining avec la commande suivante dans la console :

```
>
> miner.start(1)
INFO [11-17|11:44:12.743] Updated mining threads          threads=1
INFO [11-17|11:44:12.750] Transaction pool price threshold updated price=1,000,000,000
null
> INFO [11-17|11:44:12.753] Commit new sealing work          number=1
sealhash=07d7c2..107aae uncles=0 txs=0 gas=0 fees=0 elapsed=2.139ms
INFO [11-17|11:44:12.754] Commit new sealing work          number=1 s
ealhash=07d7c2..107aae uncles=0 txs=0 gas=0 fees=0 elapsed=2.735ms

>
> INFO [11-17|11:44:13.332] Looking for peers                peercount=
0 tried=144 static=0
INFO [11-17|11:44:21.158] Generating DAG in progress      epoch=0 perc
centage=0 elapsed=7.636s
```