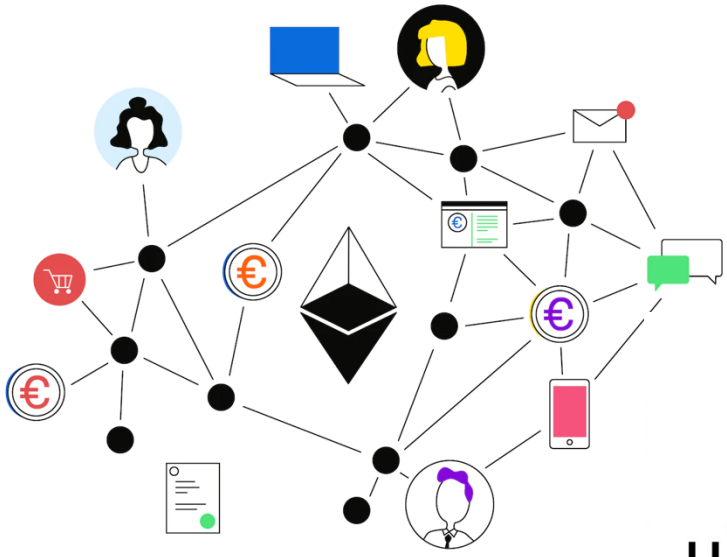
Yacine CHERIFI - MSI - A3

TRAVAUX PRATIQUE :

Création et configuration d'une Blockchain Privée Ethereum

(Client)



2022/2023

https://github.com/ycherifi/TP_Blockchain.git

I. Installation d'Ethereum sur Ubuntu :

Après avoir installé la machine virtuelle (ubuntu), nous allons commencer par installer ethereum.

```
ycherifi@Ubuntu3:~$ su root
Password:
root@Ubuntu3:/home/ycherifi# apt-get install software-proprties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package software-proprties-common
root@Ubuntu3:/home/ycherifi#
```

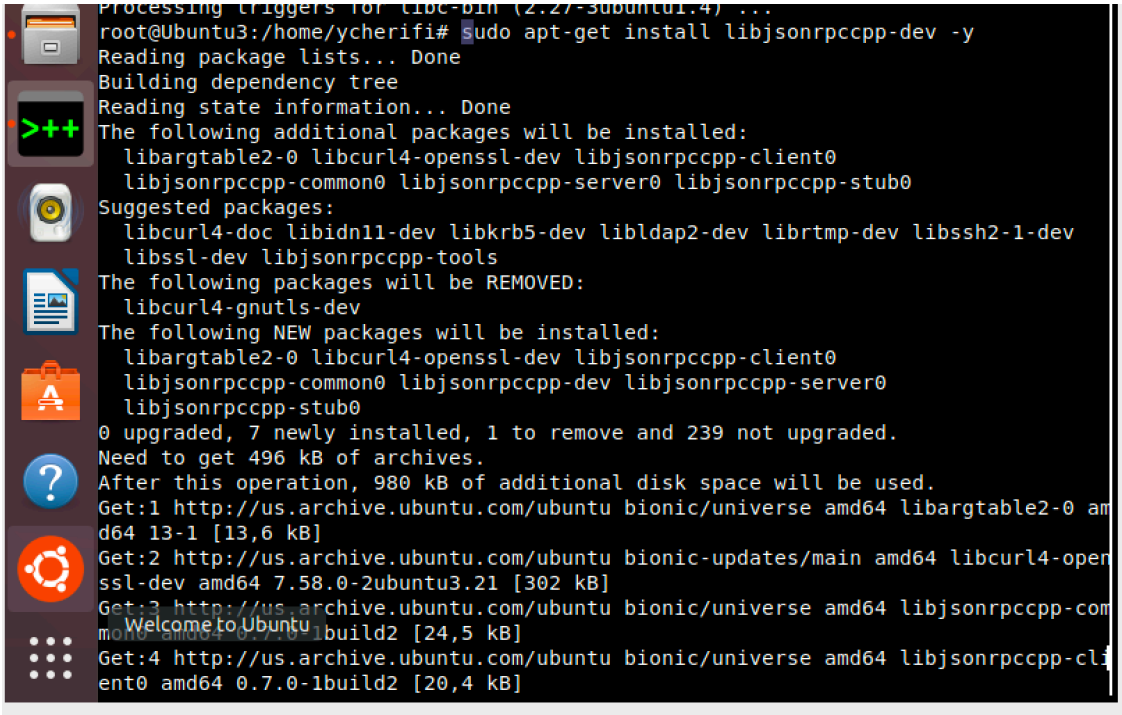
Rhythmbox

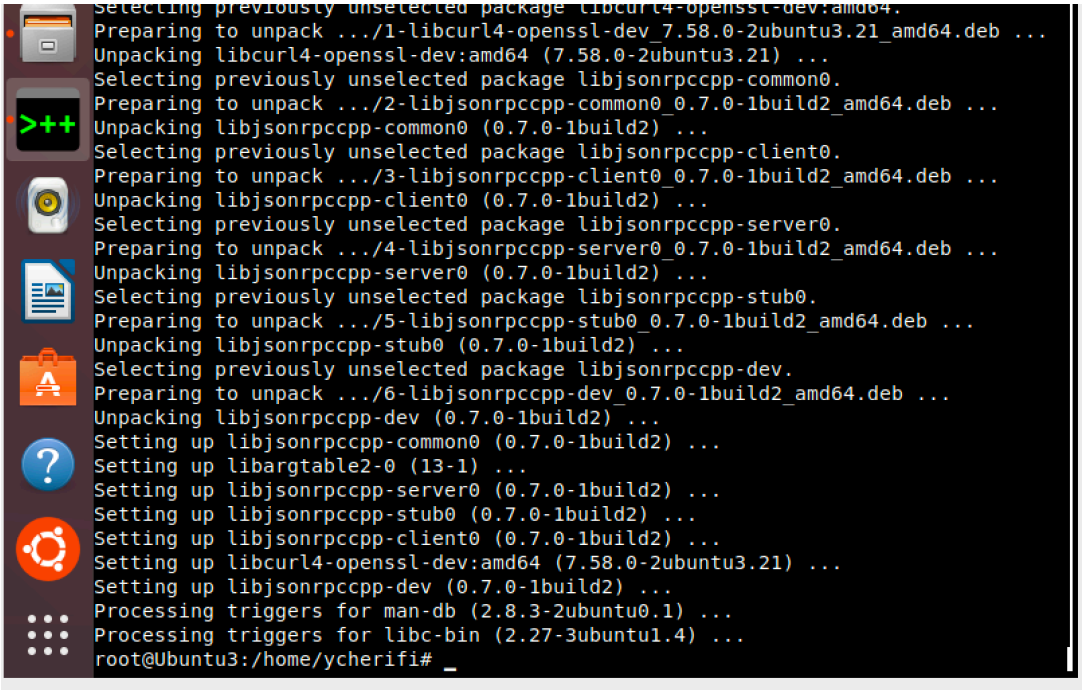
```
root@Ubuntu3:/home/ycherifi# sudo add-apt-repository -y ppa:ethereum/ethereum
Get:1 http://ppa.launchpad.net/ethereum/ethereum/ubuntu bionic InRelease [15,4 kB
]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88,7 kB]
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:4 http://ppa.launchpad.net/ethereum/ethereum/ubuntu bionic/main amd64 Package
s [2 772 B]
Get:5 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88,7 kB]
Get:6 http://ppa.launchpad.net/ethereum/ethereum/ubuntu bionic/main i386 Packages
 [2 828 B]
Get:7 http://ppa.launchpad.net/ethereum/ethereum/ubuntu bionic/main Translation-e
n [828 B]
Get:8 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [83,3 kB]
Fetched 283 kB in 1s (195 kB/s)
Reading package lists... Done
root@Ubuntu3:/home/ycherifi#
root@Ubuntu3:/home/ycherifi# _
```

LibreOffice Writer

```
root@Ubuntu3: /home/ycherifi
Preparing to unpack .../5-puppeth_1.10.26+build28212+bionic_amd64.deb ..
.
Unpacking puppeth (1.10.26+build28212+bionic) ...
Selecting previously unselected package rlpdump.
Preparing to unpack .../6-rlpdump_1.10.26+build28212+bionic_amd64.deb ..
.
Unpacking rlpdump (1.10.26+build28212+bionic) ...
Selecting previously unselected package ethereum.
Preparing to unpack .../7-ethereum_1.10.26+build28212+bionic_amd64.deb .
..
Unpacking ethereum (1.10.26+build28212+bionic) ...
Setting up bootnode (1.10.26+build28212+bionic) ...
Setting up puppeth (1.10.26+build28212+bionic) ...
Setting up clef (1.10.26+build28212+bionic) ...
Setting up evm (1.10.26+build28212+bionic) ...
Setting up abigen (1.10.26+build28212+bionic) ...
Setting up rlpdump (1.10.26+build28212+bionic) ...
Setting up geth (1.10.26+build28212+bionic) ...
Setting up ethereum (1.10.26+build28212+bionic) ...
root@Ubuntu3:/home/ycherifi#
root@Ubuntu3:/home/ycherifi#
root@Ubuntu3:/home/ycherifi# _
```

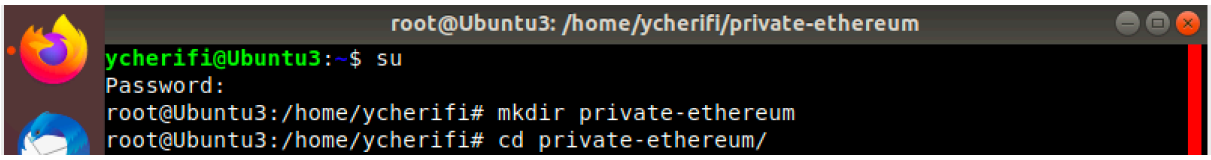
On installe le package libjsonrpccpp-dev sudo apt-get install libjsonrpccpp-dev -y





II. Création de comptes pour le réseau privé Ethereum

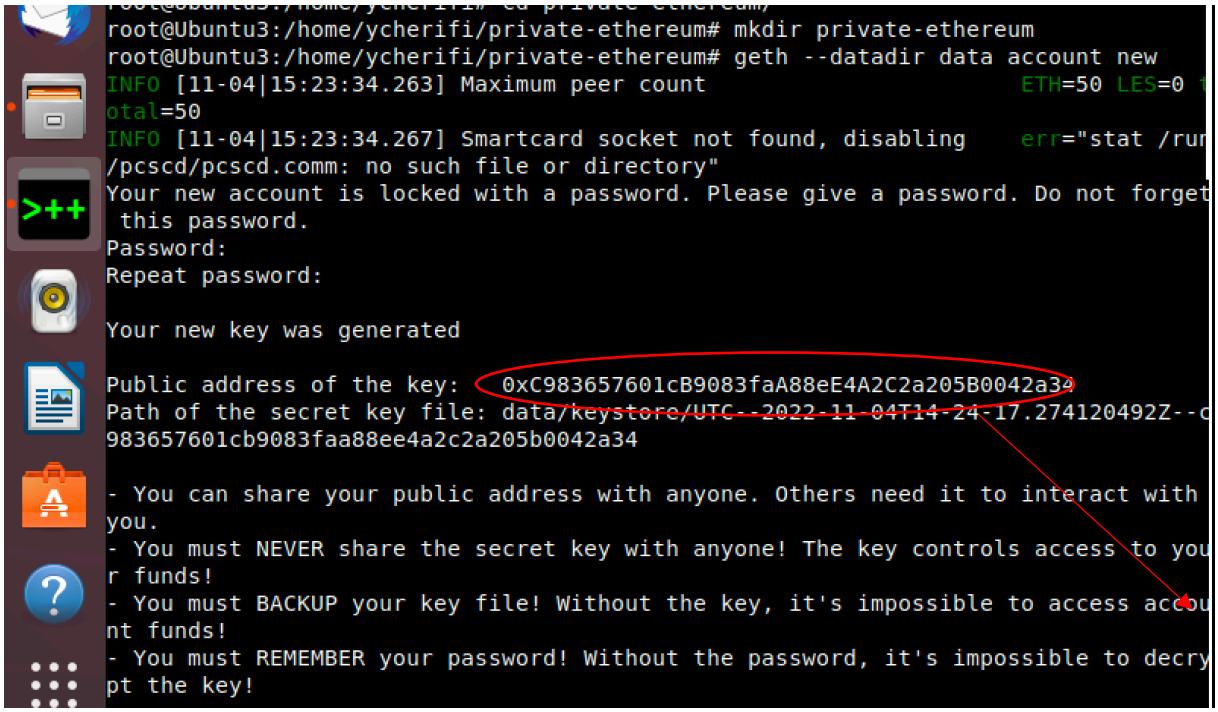
Dans un premier temps on crée un répertoire qui se nomme private-ethereum avec la commande mkdir, ensuite on se place dans ce répertoire grâce à la commande cd private-ethereum/



- Création d'un nouveau compte :

On crée un nouveau compte geth --datadir data account new. Et on lui attribue un mot de passe qu'il faut mémoriser.

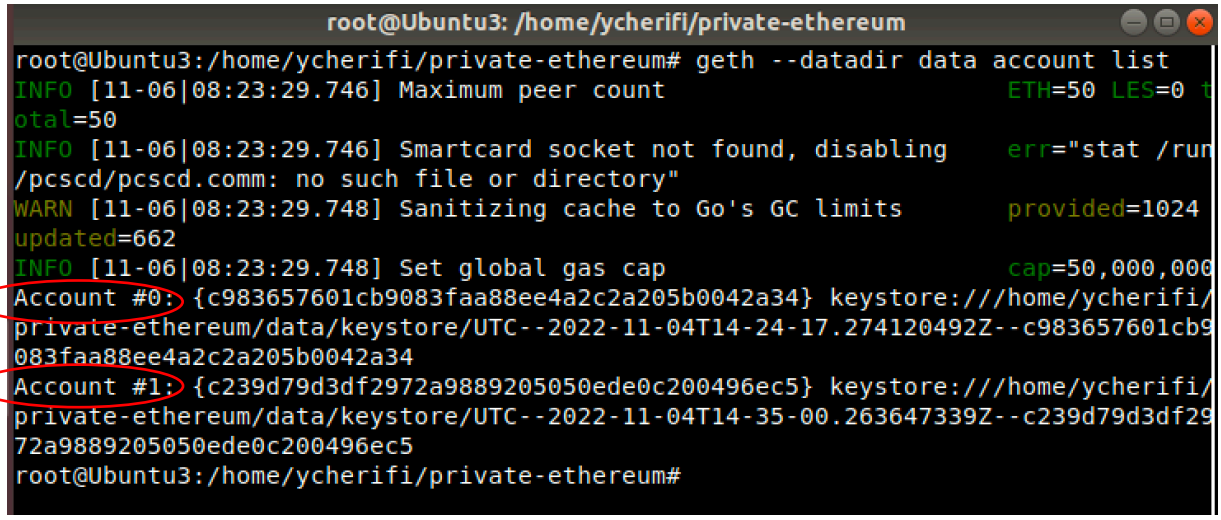
Voici le compte 1 :



L'adresse publique de la clé générée

On peut également créer un second compte avec la commande précédemment.

Vérification de la liste des comptes crées avec la commande : geth --datadir data account list

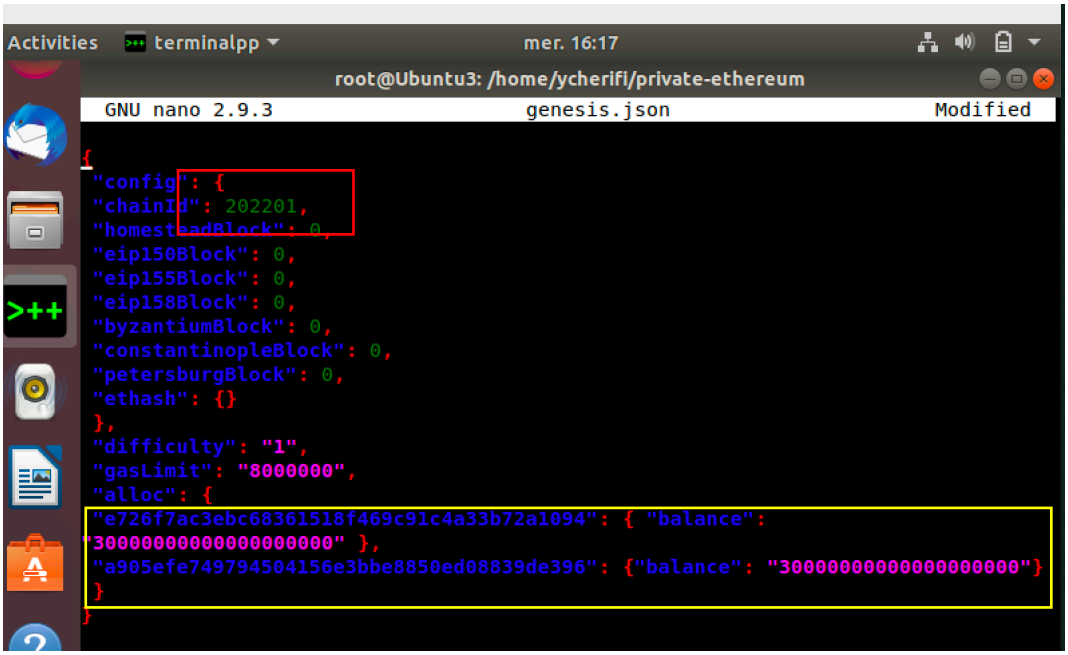


Nous pouvons voir les deux comptes que nous avons créés : account0 et 1.

III. Création du Genesis.

On crée le fichier **genesis.json** avec la commande nano. Et le modifie

On insère notre **chainID** : **202201** qui sert de protection ainsi que les 2 adresses des comptes crées par le serveur.



IV. Configuration du Bootnode.

Afin qu'un nœud puisse rejoindre le réseau et trouver d'autres nœuds nous allons créer le bootnode grâce à ces différentes commandes.

```
root@Ubuntu3:/home/ycherifi/private-ethereum# bootnode
```

```
root@Ubuntu3:/home/ycherifi/private-ethereum# bootnode --genkey=boot.key
```

```
root@Ubuntu3:/home/ycherifi/private-ethereum# bootnode --genkey=boot.key
```

```
root@Ubuntu3:/home/ycherifi/private-ethereum# bootnode --nodekey=boot.key
enode://1b14fbc4fbc2afc2157a96dfec06b38b45ddd315e52a6f11e4f8721e3155f97abd8cf5500
4bab5499a9fc933f06659bf6802bb22d5cea1f5926d94ebefa5ae1e@127.0.0.1:0?discport=3030
1
Note: you're using cmd/bootnode, a developer tool.
We recommend using a regular node as bootstrap node for production deployments.
INFO [11-17|10:07:31.897] New local node record                    seq=1,668,676,
051,894 id=ba85e32ed8b75a52 ip=<nil> udp=0 tcp=0
```

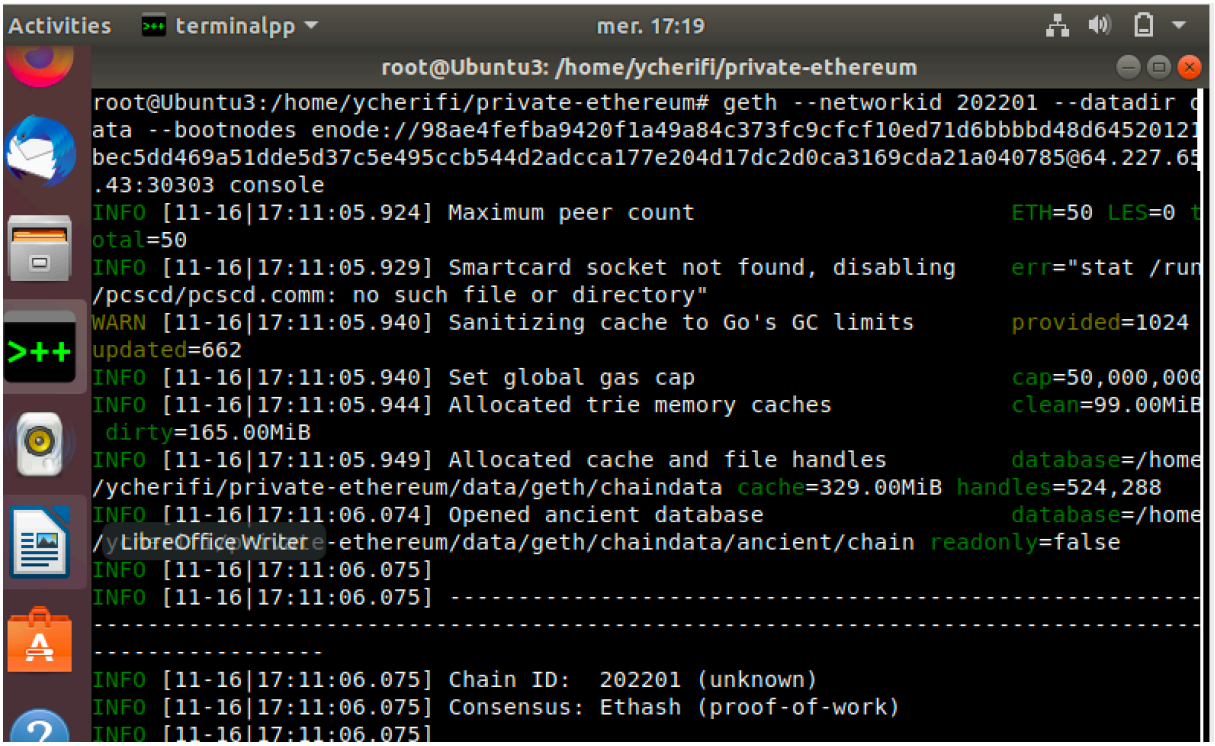
V. Démarrage des nœuds serveur avec les clients.

Une fois que bootnode sont créé, nous allons écrire cette commande en renseignant :

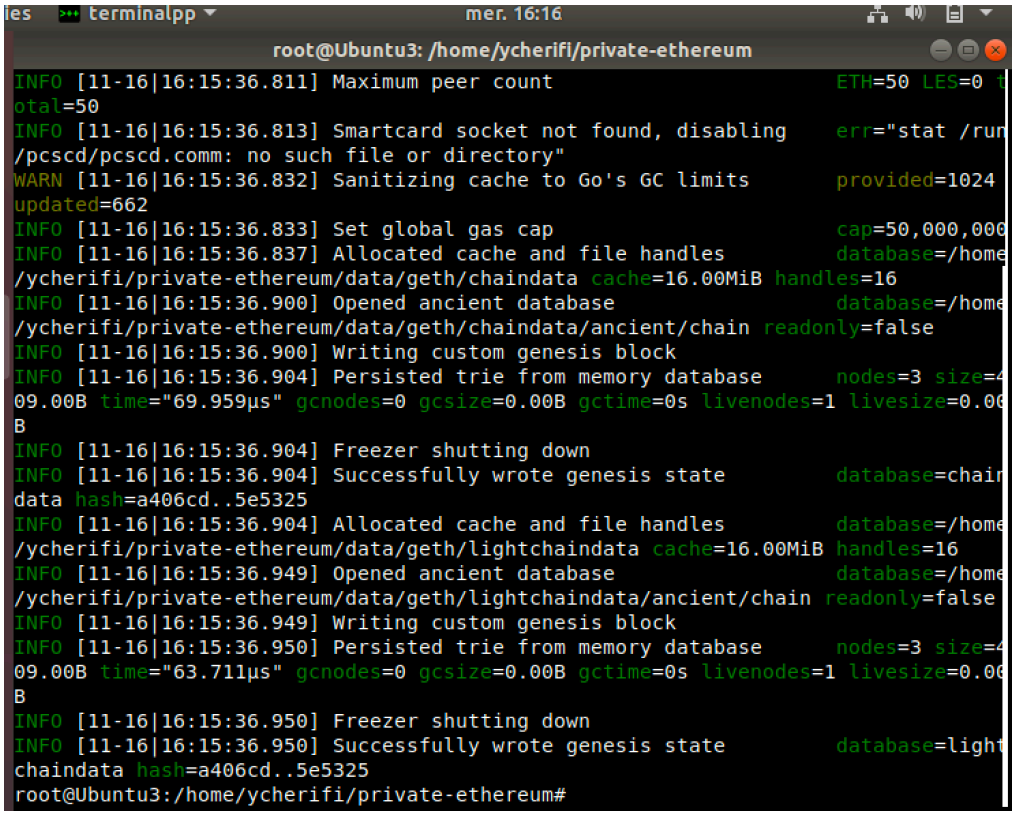
- chaineID (202201)
- Enode :
- enode://98ae4fefba9420f1a49a84c373fc9cfcf10ed71d6bbbbd48d64520121bec5dd469a51dde5d37c5e495ccb544d2adcca177e204d17dc2d0ca3169cda21a040785@64.227.65.43:30303
-
- Adresse IP du server & port : 64.227.65.43 port :30303

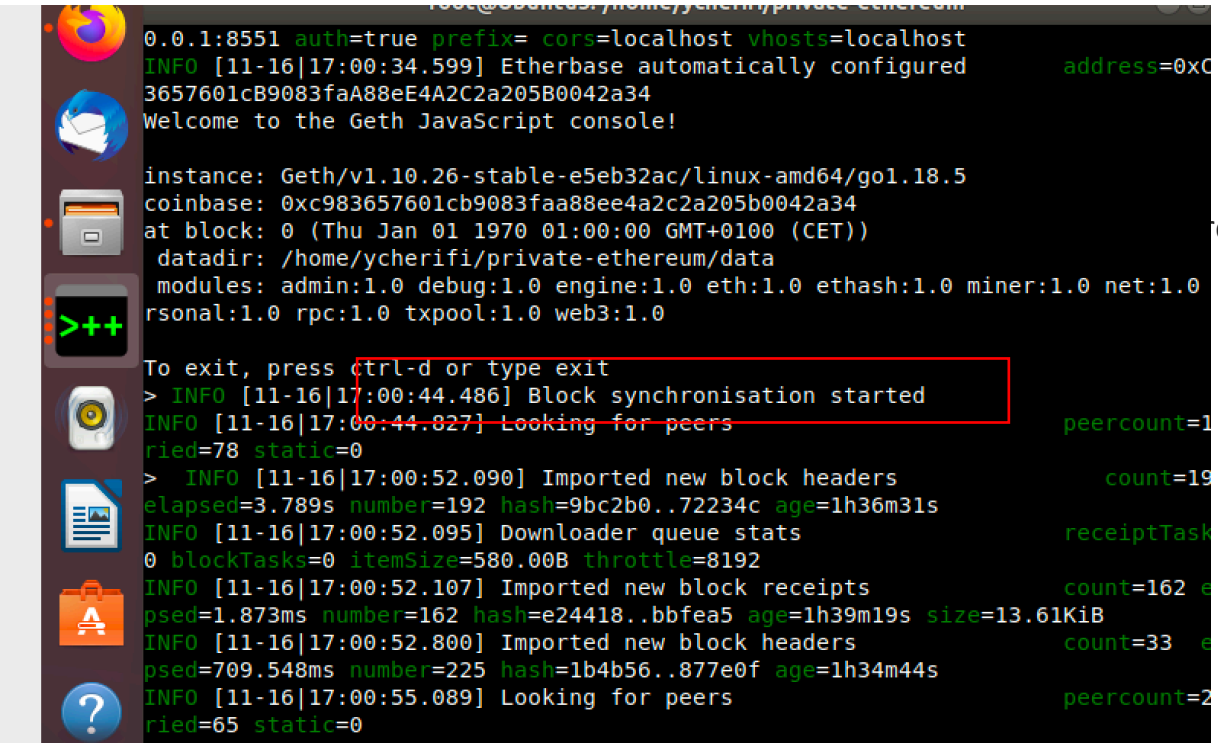
geth --networkid chaineID --datadir data --bootnodes enode fournie par le serveur@ adresse ip du serveur :port console

> net.listening



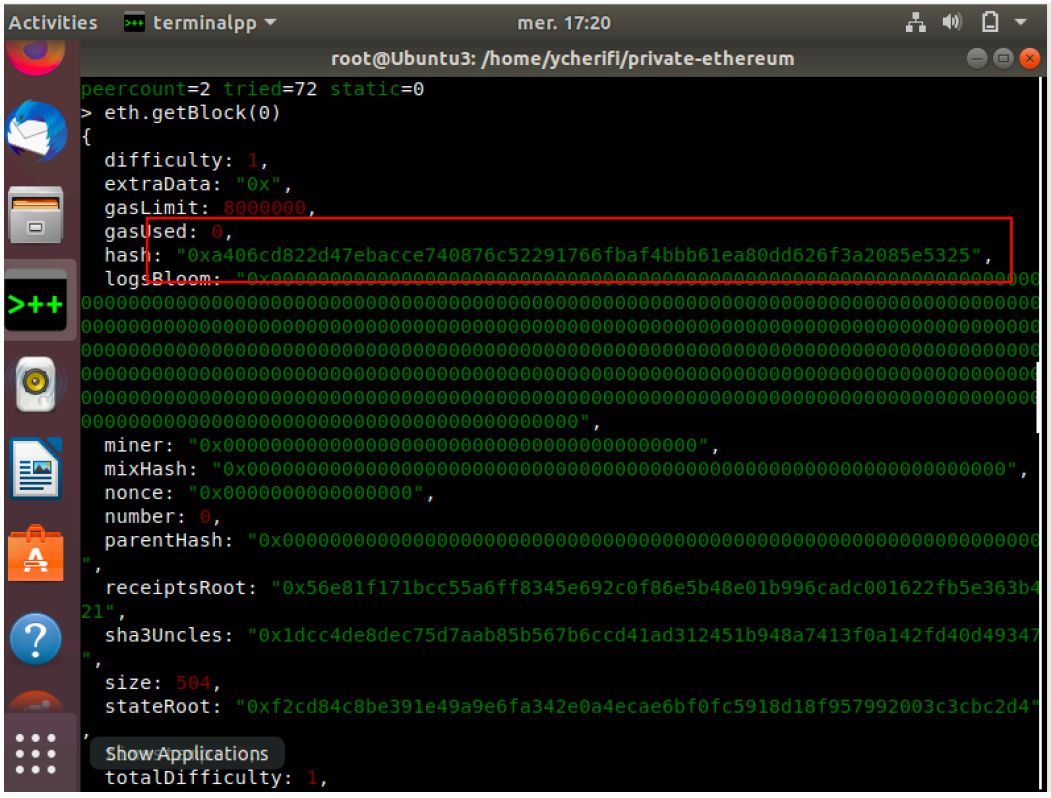
Nous pouvons voir que la synchronisation a bien commencé et le téléchargement des premiers blocks a commencé.



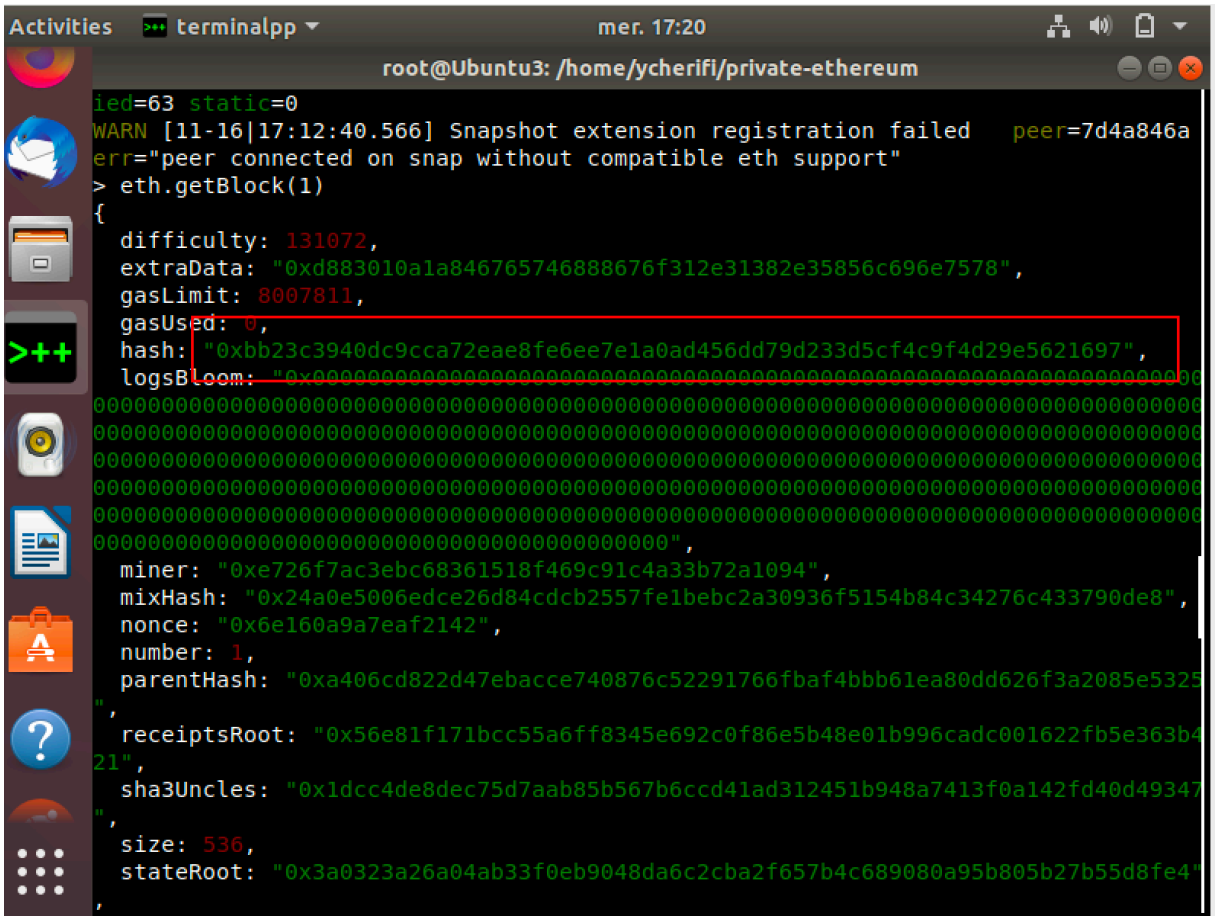


Maintenant, Vérifions que les deux premiers blocks sont identiques à celle de notre Blockchain Privée sur le server avec la commande suivante : eth.getBlock(0) et eth.getBlock(1)

Block 0 :

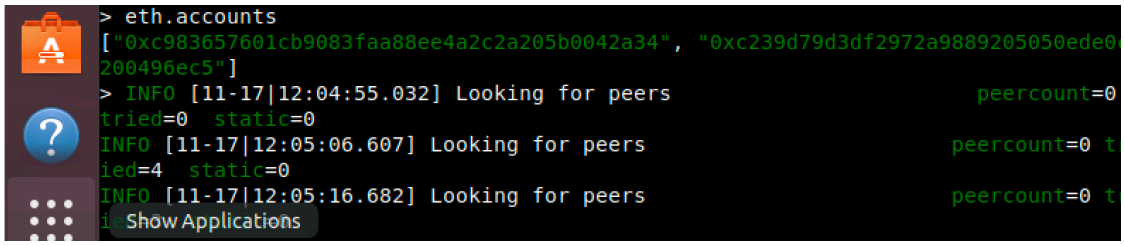


Block 1 :



Les 2 block 0 et 1, sont bien identiques à ceux du serveur en vérifiant le hash.

Ensuite nous allons vérifier si nous avons bien créé un compte avec la commande :eth.accounts



Mais après plusieurs minutes d'attente rien ne se passe !