

TRAVAUX PRATIQUE :
Création et configuration d'une Blockchain
Privée Ethereum

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

Pour créer un réseau privé Ethereum, nous devons d'abord installer Ethereum dans notre système. Dans cette section du didacticiel du réseau privé Ethereum, vous apprendrez comment installer une Blockchain Ethereum privée sur Ubuntu.

Pour les autres systèmes d'exploitation, veuillez consulter la documentation officielle : <https://geth.ethereum.org/docs/install-and-build/installing-geth>

Pour installer Ethereum, exécutez les commandes suivantes dans un terminal Linux:

```
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
```

Commençons ensuite par la création de votre Ethereum Blockchain privé. Voici les lignes de commandes sous Linux de Geth :

<https://github.com/ethereum/go-ethereum/wiki/Command-Line-Options>

II. Création de comptes pour le réseau privé Ethereum

Avant de créer de nouveaux comptes, créons un nouvel annuaire pour notre lieu de travail. Reportez-vous aux commandes ci-dessous pour ce faire:

Créer un répertoire pour votre réseau privée « private-ethereum »

```
mkdir private-ethereum  
cd private-ethereum
```

Créer ensuite un premier compte « wallet » sur votre nœud avec la commande suivante :

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

En cas d'instanciation réussie, vous devriez voir la sortie suivante :

```
root@ubuntu-esmesudria-master:~/private-ethereum# geth --datadir ./datadir account new  
INFO [11-29|21:39:19.748] Maximum peer count          ETH=50 LES=0  
total=50  
INFO [11-29|21:39:19.748] 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: _
```

Une fois ces commandes exécutées avec succès on vous demande un mot de passe. Dans votre rapport vous expliquerez pour quelle raison et que représente ce dernier.

Vous pouvez créer un second compte comme cela deux comptes seront créés et l'adresse du compte sera affichée à l'écran. Attention à bien sauvegarder les mots de passes !

```

INFO [11-29|21:41:25.305] Maximum peer count          ETH=50 LES=0
total=50
INFO [11-29|21:41:25.305] Smartcard socket not found, disabling  err="stat /ru
n/pcscd/pcscd.comm: no such file or directory"
Your new account is locked with a password. Please give a password. Do not forge
t this password.
Password:
Repeat password:

Your new key was generated

Public address of the key: 0x2a9d1c30d9b1d09e4228c865a4ee774426cac1cc
Path of the secret key file: datadir/keystore/UTC--2020-11-29T21-41-35.132281471
Z--2a9d1c30d9b1d09e4228c865a4ee774426cac1cc

- 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 yo
ur funds!
- You must BACKUP your key file! Without the key, it's impossible to access acco
unt funds!
- You must REMEMBER your password! Without the password, it's impossible to decr
ypt the key!

```

Vérifier la création de l'ensemble de vos comptes avec la commande suivante et indiquer ce que représente la liste :

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

En cas d'instanciation réussie, vous devriez voir la sortie suivante:

```

root@ubuntu-esmesudria-master:~/private-ethereum# geth --datadir ./datadir accou
nt list
INFO [11-29|20:09:05.532] Maximum peer count          ETH=50 LES=0
total=50
INFO [11-29|20:09:05.533] Smartcard socket not found, disabling  err="stat /ru
n/pcscd/pcscd.comm: no such file or directory"
INFO [11-29|20:09:05.534] Set global gas cap          cap=25000000
Account #0: {ba0a9891e8d57d7f916902a73d8bbacf4b9fb7eb} keystore:///root/private-
ethereum/datadir/keystore/UTC--2020-11-29T12-59-25.9920515422--ba0a9891e8d57d7f9
16902a73d8bbacf4b9fb7eb
Account #1: {00532a3fbb10ae33a3484052b8a4ad13c4c83692} keystore:///root/private-
ethereum/datadir/keystore/UTC--2020-11-29T13-00-20.0714534342--00532a3fbb10ae33a
3484052b8a4ad13c4c83692
root@ubuntu-esmesudria-master:~/private-ethereum# _

```

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

Un fichier Genesis contient les propriétés qui définissent la Blockchain. Un fichier Genesis est le point de départ de la Blockchain et il est donc obligatoire de créer le fichier Genesis pour créer une Blockchain. Maintenant, créons le fichier Genesis.

Commencez par créer un fichier nommé **genesis.json**

```
nano genesis.json
```

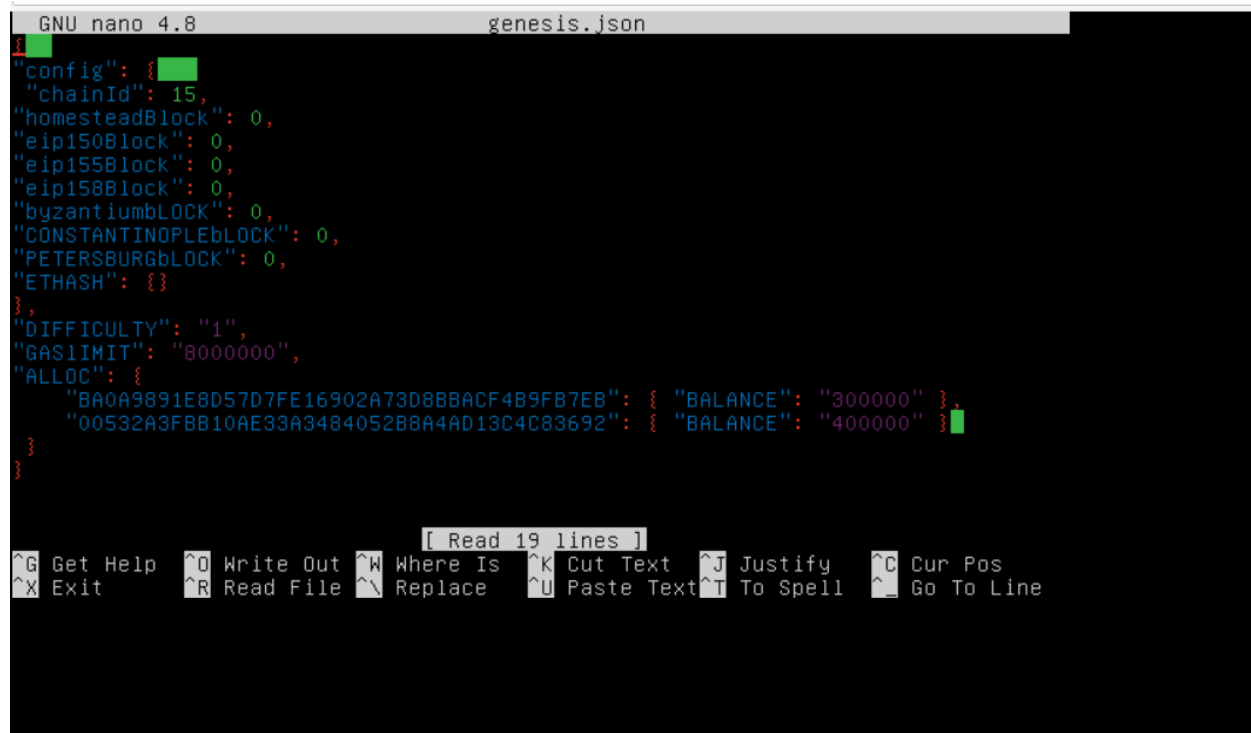
Et maintenant, copiez et collez le code suivant dans ce fichier:

Remarque : Dans le **code** ci-dessus, remplacez l'adresse sous la section d'allocation par l'adresse des comptes que vous avez créés à l'étape précédente.

```
{
  "config": {
    "chainId": 28112020,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "ethash": {}
  },
  "difficulty": "1",
  "gasLimit": "8000000",
  "alloc": {
    "b3d9fac3f9d338af629aa7ff1880d7cce06fdb60": { "balance":
"3000000000000000000000" }
  }
}
```

Dans votre rapport pouvez vous détailler à quoi corresponds chaque paramètre du genesis bloc.

En cas d'instanciation réussie, vous devriez voir la sortie suivante:



```
GNU nano 4.8 genesis.json
{
  "config": {
    "chainId": 15,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "CONSTANTINOPLEBlock": 0,
    "PETERSBURGBlock": 0,
    "ETHASH": {}
  },
  "difficulty": "1",
  "gasLimit": "8000000",
  "alloc": {
    "BA0A9891E8D57D7FE16902A73D8BBACF4B9FB7EB": { "BALANCE": "300000" },
    "00532A3FBB10AE33A3484052B8A4AD13C4C83692": { "BALANCE": "400000" }
  }
}
```

[Read 19 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line

Enregistrez-le et quittez.

Le fichier Genesis est prêt. Avant de démarrer la Blockchain, nous devons instancier le répertoire de données. Le répertoire de données est le répertoire où sont stockées les données liées à la Blockchain. Pour instancier le répertoire de données, exécutez la commande suivante:

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

En cas d'instanciation réussie, vous devriez voir la sortie suivante:

```

root@ubuntu-esmesudria-master:~/private-ethereum# geth init --datadir data genesis.json
INFO [11-29|13:51:30.064] Maximum peer count          ETH=50 LES=0 total=50
INFO [11-29|13:51:30.065] Smartcard socket not found, disabling err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [11-29|13:51:30.066] Set global gas cap          cap=25000000
INFO [11-29|13:51:30.067] Allocated cache and file handles database=/root/.private-ethereum/data/GETH/chaindata cache=16.00MiB handles=16
INFO [11-29|13:51:30.073] Writing custom genesis block
INFO [11-29|13:51:30.075] Persisted trie from memory database nodes=4 size=466.00B time="84.868µs" gcnodes=0 gcsiz=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [11-29|13:51:30.098] Successfully wrote genesis state database=chaindata hash="d92b52...42a3c9"
INFO [11-29|13:51:30.099] Allocated cache and file handles database=/root/.private-ethereum/data/GETH/lightchaindata cache=16.00MiB handles=16
INFO [11-29|13:51:30.102] Writing custom genesis block
INFO [11-29|13:51:30.103] Persisted trie from memory database nodes=4 size=466.00B time="67.559µs" gcnodes=0 gcsiz=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [11-29|13:51:30.104] Successfully wrote genesis state database=lightchaindata hash="d92b52...42a3c9"
root@ubuntu-esmesudria-master:~/private-ethereum#

```

IV. Configuration du Bootnode uniquement sur le serveur dans un premier temps :

Avec tous les nœuds que vous souhaitez exécuter initialisés à l'état de genèse souhaité, vous devrez démarrer un nœud d'amorçage que d'autres peuvent utiliser pour se trouver dans votre réseau et / ou sur Internet. La première fois qu'un nœud se connecte au réseau, il utilise l'un des nœuds de démarrage prédéfinis.

Grâce à ces nœuds de démarrage, un nœud peut rejoindre le réseau et trouver d'autres nœuds. Dans le cas d'un cluster privé, ces nœuds de démarrage prédéfinis ne sont pas d'une grande utilité. Par conséquent, go-ethereum propose une implémentation de bootnode qui peut être configurée et exécutée dans votre réseau privé.

```
> bootnode
```

En cas d'instanciation réussie, vous devriez voir la sortie suivante:

```

root@ubuntu-esmesudria-master:~/private-ethereum# bootnode
Fatal: Use -nodekey or -nodekeyhex to specify a private key

```

```
bootnode --genkey=boot.key
```

```
bootnode --nodekey=boot.key
```

Vous obtiendrez le lien enode correspondant à votre bootnode. Attention vérifier bien l'adresse IP associé dans notre cas il a été défini localement 127.0.0.1 ou [] sur le port 30301.

En cas d'instanciation réussie, vous devriez voir la sortie suivante:

```
root@ubuntu-esmesudria-master:~/private-ethereum# bootnode --nodekey=boot.key
enode://68064426a4b8f37469a6bf6ac5f84decea57ccffef1f958a00549af39c3e7610d00b91a3
645cb55041a4554b0e43e49f949200333c4e784306904d60d0852c12@127.0.0.1:0?discport=30
301
```

Ensuite vérifier que vous posséder bien la même adresse comme identifiant :

```
$ bootnode --nodekey=boot.key --writeaddress
```

En cas d'instanciation réussie, vous devriez voir la sortie suivante:

```
root@ubuntu-esmesudria-master:~/private-ethereum# bootnode --nodekey boot.key --
writeaddress
68064426a4b8f37469a6bf6ac5f84decea57ccffef1f958a00549af39c3e7610d00b91a3645cb550
41a4554b0e43e49f949200333c4e784306904d60d0852c12
```

V. Configurez votre Ethereum Private Blockchain et commencer à miner uniquement sur le serveur dans un premier temps.

Miner sur le réseau public Ethereum est une tâche complexe car elle n'est possible qu'en utilisant des GPU, nécessitant une instance ethminer activée OpenCL ou CUDA.

Cependant, dans un réseau privé, une seule instance de mineur de processeur est plus que suffisante à des fins pratiques car elle peut produire un flux stable de blocs aux intervalles corrects sans avoir besoin de ressources lourdes (envisagez de fonctionner sur un seul thread, pas besoin de plusieurs).

```
geth --mine --networkid 28112020 --http.port 30303 --nat extip:172.16.254.4 --
miner.etherbase=0xVotrePremierCompte --datadir data console
```



```

root@ubuntu-s-server-genesis:~/private-ethereum# geth --networkid 28112020 --http.port 30303 --nat extip:178.62.246.138 --datadir private-ethereum console
WARN [12-01|21:11:45.481] Sanitizing cache to Go's GC limits
INFO [12-01|21:11:45.484] Maximal peer count:
INFO [12-01|21:11:45.484] Smartcard socket not found, disabling
INFO [12-01|21:11:45.485] Set global gas cap
INFO [12-01|21:11:45.486] Allocated trie memory caches
INFO [12-01|21:11:45.488] Allocated cache and file handles
INFO [12-01|21:11:45.507] Opened ancient database
INFO [12-01|21:11:45.508] Initialized chain configuration
INFO [12-01|21:11:45.509] Disk storage enabled for ethash caches
INFO [12-01|21:11:45.509] Disk storage enabled for ethash DAGs
INFO [12-01|21:11:45.509] Initializing Ethereum protocol
WARN [12-01|21:11:45.509] Upgrade blockchain database version
INFO [12-01|21:11:45.511] Loaded most recent local header
INFO [12-01|21:11:45.511] Loaded most recent local full block
INFO [12-01|21:11:45.511] Loaded most recent local fast block
INFO [12-01|21:11:45.512] Regenerated local transaction journal
INFO [12-01|21:11:45.540] Allocated fast sync bloom
INFO [12-01|21:11:45.543] Starting peer-to-peer node
INFO [12-01|21:11:45.546] Initialized fast sync bloom
INFO [12-01|21:11:45.554] Mapped network port
INFO [12-01|21:11:45.555] Mapped network port
INFO [12-01|21:11:45.555] New local node record
INFO [12-01|21:11:45.558] IPC endpoint opened
INFO [12-01|21:11:45.571] Started P2P networking
c0fa31c69330384f4998178.62.246.138:30303
INFO [12-01|21:11:45.634] Etherbase automatically configured
Welcome to the Geth JavaScript console!

instance: Geth/v1.9.24-stable-cc05b050/linux-amd64/go1.15.5
coinbase: 0xb3d9fac3f9d338af629aa7ff1880d7cce06fdb60
at block: 0 (Thu Jan 01 1970 00:00:00 GMT+0000 (UTC))
datadir: /root/.private-ethereum/private-ethereum
modules: admin:0 debug:0 ethash:0 ethashall:0 metrics:0 net:0 personal:0 rpc:0 txpool:0 web3:0

provided=1024 updated=327
ETH=50 IDE=0 LOCAL=50
err="stat /run/pcsd/pcsd.comm: no such file or directory"
cap=25000000
clean=81.00MiB dirty=81.00MiB
database=/root/.private-ethereum/private-ethereum/ethash/chaindata cache=163.00MiB handles=524288
database=/root/.private-ethereum/private-ethereum/ethash/chaindata/ancient
config={ChainID: 28112020 Homestead: 0 DAO: <nil> DAOsupport: false EIP150: 0 EIP155: 0 EIP158: 0 Byzantium: 0 Constantinople: 0 Petersburg: 0 Istanbul: <nil>, Muir Glacier: <nil>, YOLO v2: <nil>, Engine: ethash}
dir=/root/.private-ethereum/private-ethereum/ethash count=3
dir=/root/.ethash count=2
versions="[65 64 63]" network=28112020 dbversion=<nil>
from=<nil> to=8
number=0 hash="7b340c-be6c2c" td=1 age=51y7mo3w
number=0 hash="7b340c-be6c2c" td=1 age=51y7mo3w
number=0 hash="7b340c-be6c2c" td=1 age=51y7mo3w
transactions=0 accounts=0
size=162.00MiB
instance=Geth/v1.9.24-stable-cc05b050/linux-amd64/go1.15.5
items=1 errorrate=0.000 elapsed=1.894ms
proto=udp extport=30303 intport=30303 interface=ExtIP(178.62.246.138)
proto=tcp extport=30303 intport=30303 interface=ExtIP(178.62.246.138)
seq=1 id=ef6834ef64943b17 ip=178.62.246.138 udp=30303 tcp=30303
url=/root/.private-ethereum/private-ethereum/ethash.ipc
self=node://2ffad05a9e5f7afedd1e4b7903ff6c9c9b0cc12b060a942996f3376a4a117d1d319b12f0a252181cad3b68be6cacf4435ba013902
address=0xb3d9fac3f9d338af629aa7ff1880d7cce06fdb60

```

- **networkid**: identifiant réseau de ce réseau ethereum. Vous choisissez une valeur que vous voulez. Par exemple: olympic (0), frontier (1), morden (2), ropsten (3).

- **nat extip**: l'adresse IP publique de votre Bootnode

- **mine**: permet l'exploitation minière.

- **rpc**: active un serveur HTTP-RPC. Les applications de portefeuille peuvent se connecter à ce nœud de minage via http.

- **rpcaddr**: spécifie l'interface d'écoute du serveur HTTP-RPC (par défaut: «localhost»)

- **rpcport**: spécifie le port d'écoute du serveur HTTP-RPC (par défaut: 8545)

- **rpcapi**: spécifie les API proposées via l'interface HTTP-RPC (par défaut: "eth, net, web3")

- **rpccorsdomain**: active CORS en spécifiant une liste de domaines séparés par des virgules à partir desquels accepter les requêtes d'origine croisée. Ceci il est utile lorsque vous utilisez des éditeurs solidity basés sur un navigateur (remix) pour déployer des contrats intelligents ou des portefeuilles basés sur un navigateur. Par exemple, la suite acceptera CORS de n'importe quel domaine.

Vérifier l'état de la Blockchain et le numéro de Block qui doit être à zero avec la commande :

```
$ eth.blockNumber
```

En cas d'instanciation réussie, vous devriez voir la sortie suivante:

```
> eth.blockNumber
0
```

Vérifier que vous avez reçu à l'initialisation du genesis block vos ethers :

```
$ eth.getBalance(eth.accounts[0])
```

En cas d'instanciation réussie, vous devriez voir la sortie suivante:

```
INFO [12-01|21:18:36.594] Looking for peers                peercount=0 tried=146 static=0
> eth.getBalance(eth.accounts[0])
300000000000
```

Vérifier le genesis block

```
$ eth.getBlock(0)
```

En cas d'instanciation réussie, vous devriez voir la sortie suivante:

[illegible]

Lancer la commande de mining suivante :

```
$ miner.start(1)
```

En cas d'instanciation réussie, vous devriez voir la sortie suivante:

Vérifier la création du block 1

En cas d'instanciation réussie, vous devriez voir la sortie suivante:

Quitter la console geth avec la commande (ctrl + d) et affecter votre adresse ip à votre nœud :

En cas d'instanciation réussie, vous devriez voir la sortie suivante:

Le nœud affichera ensuite une URL enode que d'autres nœuds peuvent utiliser pour s'y connecter et échanger des informations de pair.

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

Avec le nœud de démarrage « Bootnode » et accessible de l'extérieur (vous pouvez essayer telnet <ip> <port> pour vous assurer qu'il est bien accessible), démarrez chaque nœud geth suivant pointé vers le nœud de démarrage pour la découverte des pairs via l'indicateur --bootnodes. Il sera probablement également souhaitable de garder le répertoire de données de votre réseau privé séparé, alors spécifiez également un indicateur personnalisé --datadir.

Sur les machines PC/MAC vous devez créer un compte, votre bootnode, récupérer le fichier Genesis.json (en réalisant la commande init depuis la version Geth 1.19) et ensuite réaliser la commande ci-dessous :

Coté server:

```
geth --networkid 28112020 --nat extip:172.16.254.136 --datadir data console
```

Puis dans la console réaliser la commande suivante pour écouter tous les ports :

```
> net.listening
```

Coté client :

```
geth --networkid 28112020 --datadir data --bootnodes  
enode://2fffad05a9e5f7af6dd11e4b7903ff8c8c9b0cc812b060a942996f3376a4a1  
17d1d319b12f0a252181cad3b68be6cacf4435ba013302c0fa31c69330384f499@  
178.62.246.138:30303 console
```

Puis dans la console réaliser la commande suivante pour écouter tous les ports :

```
> net.listening
```

```
root@ubuntu:~# ./client-private-ethereum3-geth --networkid=main12020 --datadir data --bootnodes enode://7ffad05ae5f7af6dd1e4b7903ff8c5cb0cc3d2b060a54296ef376aa417d1d319b12f0a25210icaadbb6ecac44453da013002ccfa1ce6930384f4998178.62.246.138:30303 console
```

```
INFO [12-02|18:24:54.685] Maximum peer count
INFO [12-02|18:24:54.685] Smartcard socket not found, disabling
INFO [12-02|18:24:54.686] Set global gas cap
INFO [12-02|18:24:54.686] Allocated trie memory caches
INFO [12-02|18:24:54.686] Allocated cache and file handles
INFO [12-02|18:24:54.701] Opened ancient database
INFO [12-02|18:24:54.702] Initialized chain configuration
INFO [12-02|18:24:54.702] Genesis: <nil>, Muir Glacier: <nil>, YOLO v2: <nil>, Engine: ethash<nil>
INFO [12-02|18:24:54.702] Disk storage enabled for ethash caches
INFO [12-02|18:24:54.702] Disk storage enabled for ethash DAGs
INFO [12-02|18:24:54.702] Initializing Ethereum protocol
WARN [12-02|18:24:54.702] Upgrade blockchain database version
INFO [12-02|18:24:54.703] Loaded most recent local header
INFO [12-02|18:24:54.703] Loaded most recent local full block
INFO [12-02|18:24:54.703] Loaded most recent local fast block
INFO [12-02|18:24:54.703] Regenerated local transaction journal
INFO [12-02|18:24:54.711] Allocated fast sync bloom
INFO [12-02|18:24:54.718] Starting peer-to-peer node
INFO [12-02|18:24:54.719] Initialized fast sync bloom
INFO [12-02|18:24:54.725] New local node record
INFO [12-02|18:24:54.726] Started P2P networking
faed2ac7f836259be168127101:1:30303
INFO [12-02|18:24:54.726] IPC endpoint opened
INFO [12-02|18:24:54.770] Ethersize automatically configured
Welcome to the Geth JavaScript console!

instance: Geth/v1.9.24-stable-cc05b050/linux-amd64/gol.15.5
coinbase: 0xd02ec416c4185e8d488fe5bd0488695f7773
at blocks: 0 (eth can ol 1970 00:00:00 CMT+0000 (UTC))
datadir: /root/private-ethereum3/data
modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 network:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d
> INFO [12-02|18:24:55.948] New local node record
INFO [12-02|18:24:56.042] IPC endpoint started
INFO [12-02|18:25:04.820] InnoDB engine started
mysql -u root -p882ms processes:1 sending:nao transaction:0 undo:0 log:0 innodb:0 user:root@localhost
```

```
INFO [12-02-18:24:55.948] New local node record seq=2 id=id970c7cf9dfeel9 ip=64.225.79.7 seq=30303 tcp=30303
INFO [12-02-18:25:04.727] Block synchronisation started
INFO [12-02-18:25:04.820] Imported new state entries count=1 elapsed=1.982ms processed=1 pending=0 trieretry=0 coderetry=0 duplicate=0 unexpected=0
INFO [12-02-18:25:04.932] Looking for peers peercount=1 tried=166 static=0
INFO [12-02-18:25:05.688] Imported new block headers count=192 elapsed=754.236ms number=192 hash="b664ac...32b276" age=19h47m52s
INFO [12-02-18:25:05.691] Downloader queue stats receiptTasks=0 blockTasks=0 itemSize=572.00b throttle=8192
INFO [12-02-18:25:05.694] Imported new block receipts count=192 elapsed=1.080ms number=192 hash="b664ac...32b276" age=19h47m52s size=768.00B
INFO [12-02-18:25:05.691] Imported new block headers count=192 elapsed=29.180ms number=384 hash="343081...422262" age=19h35m37s
INFO [12-02-18:25:05.698] Imported new block receipts count=192 elapsed=68.091ms number=384 hash="343081...422262" age=19h35m37s size=768.00B
INFO [12-02-18:25:05.946] Imported new block headers count=2048 elapsed=725.799ms number=2432 hash="40c44d...fa499e" age=1h54m50s size=8.00KiB
INFO [12-02-18:25:06.127] Imported new block receipts count=2048 elapsed=18.035ms number=2432 hash="40c44d...fa499e" age=1h54m50s size=8.00KiB
INFO [12-02-18:25:07.272] Imported new block headers count=448 elapsed=1.317s number=2880 hash="8a0654...9e5b10" age=5m10s size=8.00KiB
INFO [12-02-18:25:07.280] Imported new block receipts count=448 elapsed=2.126ms number=2880 hash="8a0654...9e5b10" age=5m10s size=1.75KiB
```

```
eth.getBlock(0)
```

[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:

```
eth.accounts
```

Avec le résultat ci-dessous :

```
> eth.accountsINFO [12-01|23:19:15.221] Looking for peers peercount=1 tried=78 static=0
["0xe07f7b3e5bfb1f3e426f555cc0a05224e3f50a9b", "0xcc2155b6b453f31295a99e2d0b9c271aee92a575"]
> INFO [12-01|23:19:17.033] Generating DAG in progress epoch=1 percentage=44 elapsed=2m13.545s
```

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

```
miner.start(1)
```

```
INFO [12-02|19:32:16.410] Updated mining threads threads=1
INFO [12-02|19:32:16.411] Transaction pool price threshold updated price=10000000000
null
> INFO [12-02|19:32:16.411] Commit new mining work number=3314 sealhash="3a6b1
4...bf1742" uncles=0 txs=0 gas=0 fees=0 elapsed="117.813µs"
INFO [12-02|19:32:16.666] Imported new chain segment blocks=1 txs=0 mgas=0.000 ela
psed=4.093ms mgasps=0.000 number=3314 hash="e84888...3a4198" dirty=2.15KiB
INFO [12-02|19:32:16.666] Commit new mining work number=3315 sealhash="eade41...
574435" uncles=0 txs=0 gas=0 fees=0 elapsed="100.039µs"
INFO [12-02|19:32:17.737] Successfully sealed new block number=3315 sealhash="eade41...
574435" hash="edd3d6...1d22b7" elapsed=1.070s
INFO [12-02|19:32:17.737] ⚡ mined potential block number=3315 hash="edd3d6...1d2
2b7"
INFO [12-02|19:32:17.737] Commit new mining work number=3316 sealhash="65e21b...
5d9884" uncles=0 txs=0 gas=0 fees=0 elapsed="162.952µs"
```

Bravo vous avez terminé le TP. Vous devez désormais fournir un rapport détaillant les commandes exécutées ainsi que les résultats obtenus.