TP : Lancement d’une Blockchain sur un serveur distant et connexions des nœuds distants.

▼ Préambule : Se connecter a un serveur distant en SSH

```
ssh <user>@<ip>
ssh <user>@<ip>'s password: 🔑
```

👉 user: voir dans googledrive
ip: voir dans googledrive
mot de pass: voir dans googledrive

⚠️ Lorsque vous tapez votre mot de passe aucun caractère ne s’affiche, c’est normal

▼ Première étape : mise en place de la blockchain

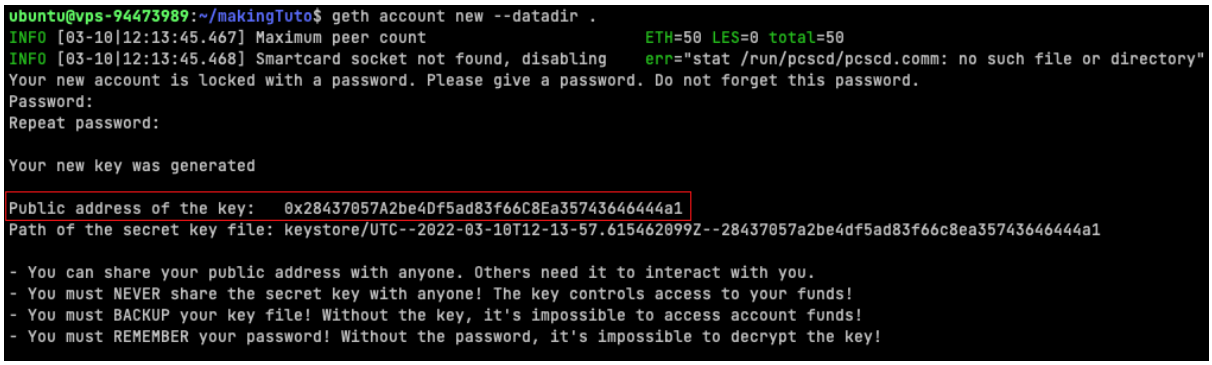
Faire un dossier de travail:

```
mkdir <nom_dossier>
cd <nom_dossier>
```

Créer un compte:

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

- Mot de passe a inscrire deux fois
- Conserver son adresse publique



- Ajouter votre mot de passe dans un fichier pour une utilisation plus fiable:

```
echo '<mot de passe>' > pwd.txt
```

Créer un fichier Genesis (⚠️ uniquement sur le noeud principal):

```
puppeth
> <nom blockchain>

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

Which accounts are allowed to seal? (mandatory at least one)
> <Public address of the key>
> 0x
```

```
Which accounts should be pre-funded? (advisable at least one)
> <Public address of the key>
> 0x

Should the precompile-addresses (0x1 .. 0xff) be pre-funded with 1 wei? (advisable yes)
> yes

Specify your chain/network ID if you want an explicit one (default = random)
> <chain id>

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

 1. Modify existing configurations
 2. Export genesis configurations
 3. Remove genesis configuration
> 2

Which folder to save the genesis specs into? (default = current)
  Will create tutokovan.json, tutokovan-aleth.json, tutokovan-harmony.json, tutokovan-parity.json
>
```

Initialisation de genesis:

```
geth --datadir . init <nom blockchain>.json
```

▼ Deuxième étape : lancement de la blockchain

```
geth --datadir . --syncmode 'full' --networkid "4242" --port "30303" --http --http.addr '127.0.
0.1' --http.port "8545" --http.api 'personal,eth,net,web3,txpool,miner,admin,clique' --nodiscov
er --mine --miner.gaslimit '9000000000000' --allow-insecure-unlock --unlock <Public address of
 the key> --password "pwd.txt"
```

⚠️ Parametres a changer
--networkid "<chaine id>"
--unlock <Public address of the key>
--password "pwd.txt"

⚠️ Lorsque la blockchain est lancée, lancer un autre terminal pour les prochaines manipulations.

▼ Troisième étape éventuelle : Agir sur la blockchain

- Lancer la console:

```
geth attach http://127.0.0.1:8545
```

- Voir les comptes connectés à la blockchain:

```
eth.accounts
```

- Voir la balance d’un compte:

```
eth.getBalance("<Public address of the key>")
```

➡️ Différents modules (api) ont été installés ['personal,eth,net,web3,txpool,miner,admin,clique'].
Pour voir les différentes méthodes vous pouvez simplement appeler le module et la liste sera dans l’output ou https://geth.ethereum.org/docs/rpc/server
👩‍💻 **try and enjoy**

```
ubuntu@vps-94473989:~/makingTuto$ geth attach http://127.0.0.1:8545
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.3-stable-991384a7/linux-amd64/go1.16.3
coinbase: 0x4a3d642c374b16c08234c285b0348b0893feb848
at block: 27 (Thu Mar 10 2022 12:49:19 GMT+0000 (UTC))
 datadir: /home/ubuntu/makingTuto
 modules: admin:1.0 clique:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d
> eth.accounts
["0x4a3d642c374b16c08234c285b0348b0893feb848"]
> eth.getBalance("0x4a3d642c374b16c08234c285b0348b0893feb848")
9.046256971665327767466483203803742801036717552003169065582623750618213253 12e+74
> eth
{
  accounts: ["0x4a3d642c374b16c08234c285b0348b0893feb848"],
  blockNumber: 62,
  coinbase: "0x4a3d642c374b16c08234c285b0348b0893feb848",
  compile: {
    lll: function(),
    serpent: function(),
    solidity: function()
  },
  defaultAccount: undefined,
  defaultBlock: "latest",
  gasPrice: 1000000000,
  hashrate: 0,
  mining: true,
  pendingTransactions: [],
  protocolVersion: undefined,
  syncing: false,
  call: function(),
  chainId: function(),
  contract: function(abi),
  createAccessList: function(),
  estimateGas: function(),
  fillTransaction: function(),
  filter: function(options, callback, filterCreationErrorCallback),
  getAccounts: function(callback),
  getBalance: function(),
  getBlock: function(),
  getBlockByHash: function(),
  getBlockByNumber: function(),
  getBlockNumber: function(callback),
  getBlockTransactionCount: function(),
  getBlockUncleCount: function(),
  getCode: function(),
```

▼ Quatrième étape : connexion à serveur distant

▼ Mise en place de la Blockchain

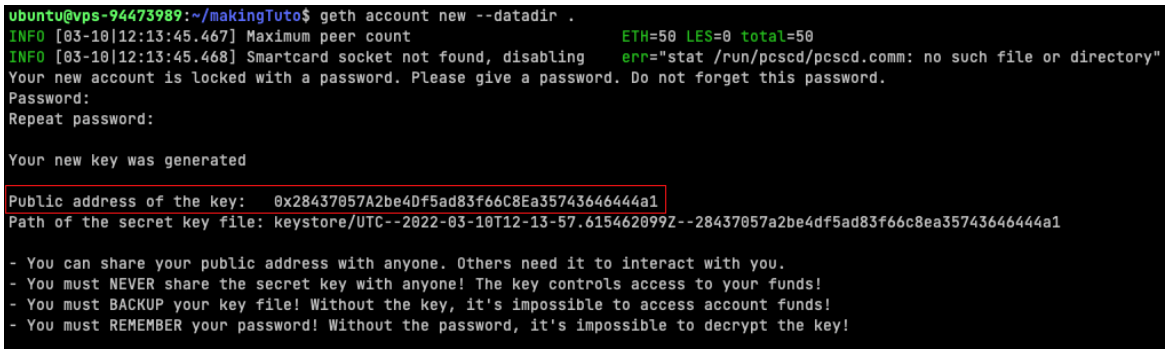
Faire un dossier de travail:

```
mkdir <nom_dossier>
cd <nom_dossier>
```

Créer un compte:

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

- Mot de passe a inscrire deux fois
- Conserver son adresse publique



- Ajouter votre mot de passe dans un fichier pour une utilisation plus fiable:

```
echo '<mot de passe>' > pwd.txt
```

Récupérer le fichier Genesis sur le serveur distant:

```
scp <user>@<ip>:<path/to/file.json> <path/local>
ssh <user>@<ip>'s password: 🔑

# scp <user>@<ip>:/home/ubuntu/makingTuto/kovantuto.json .
```

👉 user: voir dans googledrive
ip: voir dans googledrive
mot de pass: voir dans googledrive

Initialisation de la blockchain avec le fichier de genesis:

```
geth --datadir . init <genesis file>.json

# geth --datadir . init kovantuto.json
```

▼ Lancement de la blockchain:

```
geth --datadir . --syncmode 'full' --networkid "4242" --port "30303" --http --http.port "85
45" --http.api 'personal,eth,net,web3,txpool,miner,admin,clique' --nodiscover --mine --mine
r.gaslimit '9000000000000' --allow-insecure-unlock --unlock <Public address of the key> --p
assword "pwd.txt"
```

⚠️ Parametres a changer:
--networkid "<chaine id>"
--unlock <Public address of the key>
--password "pwd.txt"

⚠️ Lorsque la blockchain est lancée, lancer un autre terminal pour les prochaines manipulations.

▼ Synchronisation de la blockchain:

- Lancer la console (console noeud local):

```
geth attach --datadir .
```

- Récupérer l'enode du serveur distant (console noeud principal):

```
> admin.nodeInfo.enode
```

- Ajouter le nouveaux node en remplaçant l'ip du server (noeud local) 127.0.0.1 par (console noeud local):

```
> admin.addPeer("enode://0d8a135b20881ef3da90cdcddf771dfcd9d6bac8a22ae80142cfe2ec296ce9bc1c2b4a566eb0fe56e1a9fff5d35e8f527c086e87b637c46a65ce5cc29f4122e9@<ip>:30303?discport=0")
```

👉 ip: voir dans googledrive

- Autorisé le noeud local pour miner en PoA (console noeud principal):

```
clique.propose("<Public address of the key>", true)
```

- Verification:

```
clique.getSigners()
```

Ressources:

- https://www.youtube.com/watch?v=UQammT_Pulk
- https://geth.ethereum.org/docs/interface/command-line-options
- https://besu.hyperledger.org/en/stable/