The Chinese University of Hong Kong

Department of Financial Technology

FTEC5520– Applied Blockchain & Cryptocurrency

Lab1 Exercise

Ethereum Infrastructure Setup & Practice – Part1

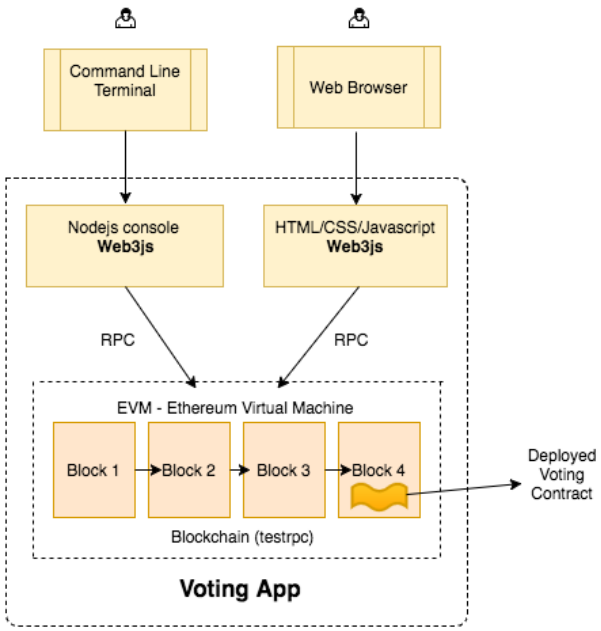
Lab object:

1. Set up the Ethereum development environment.
2. Generate the configuration file (.json) of genesis block by tool.
3. Initialize and configure a private blockchain network
4. Create account and manage account balance
5. Develop & deploy a simple Solidity smart contract.
6. Interact with the deployed contract in NodeJS console.

Introduction:

1. **Etherum** is an Open Source Blockchain-based computing platform provides Decentralized Virtual Machine aka EVM used to build smart contract.
2. **Smart contract** is a computer program that enables the capability to execute and enforce an agreement. Now terms of the smart contact are recorded in a computer language e.g. Solidity, JAVA, Node, Python, Go etc. as a set of instruction.
3. **Solidity** is a high level object oriented language for Smart contracts and is compiled to bytecode that is executable on the EVM.
4. **EVM (Ethereum Virtual Machine)** contains the Stack of protocols that defines a platform for **DAPP** aka decentralized apps e.g. Simple Smart Contract forvoting we used to demo on private blockchain network instead of the real worldwide one.

Note: due to time limitation we won't cover how to built a Web dApp but only CLI:



Task 1 Install Ethereum (Geth)

This section will take you through complete process of the installation and configuration of the Ethereum blockchain on Ubuntu Linux with Geth CLI tool. Pls follow the steps below:

Pls always use **ftec5520** as the password to avoid confusing yourself.

Install Software Properties Common (may has already exit)

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

Add Ethereum repo

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

Update sources:

```
sudo apt-get update
```

```
sudo apt update
```

Install Ethereum & Geth using apt-get:

```
sudo apt install ethereum
```

Or Install geth Puppeth

```
sudo apt-get install geth puppeth
```

Install necessary components

```
sudo apt-get install gcc g++ make
```

Create a directory to hold the blockchain network files

```
mkdir blockchain
```

Access the directory for upcoming operations

```
cd blockchain
```

Task 2 Create a Genesis File

The genesis.json file determines the characteristics of the first block in your private blockchain the genesis block.

We highly recommend you use **puppeth** – a CLI tool for Ethereum Private Network Management.

```
ubuntu@ip-172-31-21-150:~/blockchain$ puppeth
+-----------------------------------------------------------+
| Welcome to puppeth, your Ethereum private network manager |
|                                                           |
| This tool lets you create a new Ethereum network down to  |
| the genesis block, bootnodes, miners and ethstats servers |
| without the hassle that it would normally entail.         |
|                                                           |
| Puppeth uses SSH to dial in to remote servers, and builds |
| its network components out of Docker containers using the |
| docker-compose toolset.                                   |
+-----------------------------------------------------------+

Please specify a network name to administer (no spaces, hyphens or capital letters pl
ease)
>
```

Then follow the guidance to input information (pls use **ftec5520** as the network name).

```
+------------------------------------------------------------+
| Welcome to puppeth, your Ethereum private network manager  |
|                                                            |
| This tool lets you create a new Ethereum network down to   |
| the genesis block, bootnodes, miners and ethstats servers  |
| without the hassle that it would normally entail.          |
|                                                            |
| Puppeth uses SSH to dial in to remote servers, and builds  |
| its network components out of Docker containers using the  |
| docker-compose toolset.                                    |
+------------------------------------------------------------+

Please specify a network name to administer (no spaces, hyphens or capital letters please)
> ftec5520

Sweet, you can set this via --network=ftec5520 next time!

INFO [10-14|02:58:35.684] Administering Ethereum network           name=ftec5520
WARN [10-14|02:58:35.685] No previous configurations found         path=/home/ubuntu/.puppeth/ftec5520

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
> 1

Which accounts should be pre-funded? (advisable at least one)
> 0x

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

Specify your chain/network ID if you want an explicit one (default = random)
>
INFO [10-14|03:00:22.732] Configured new genesis block
```

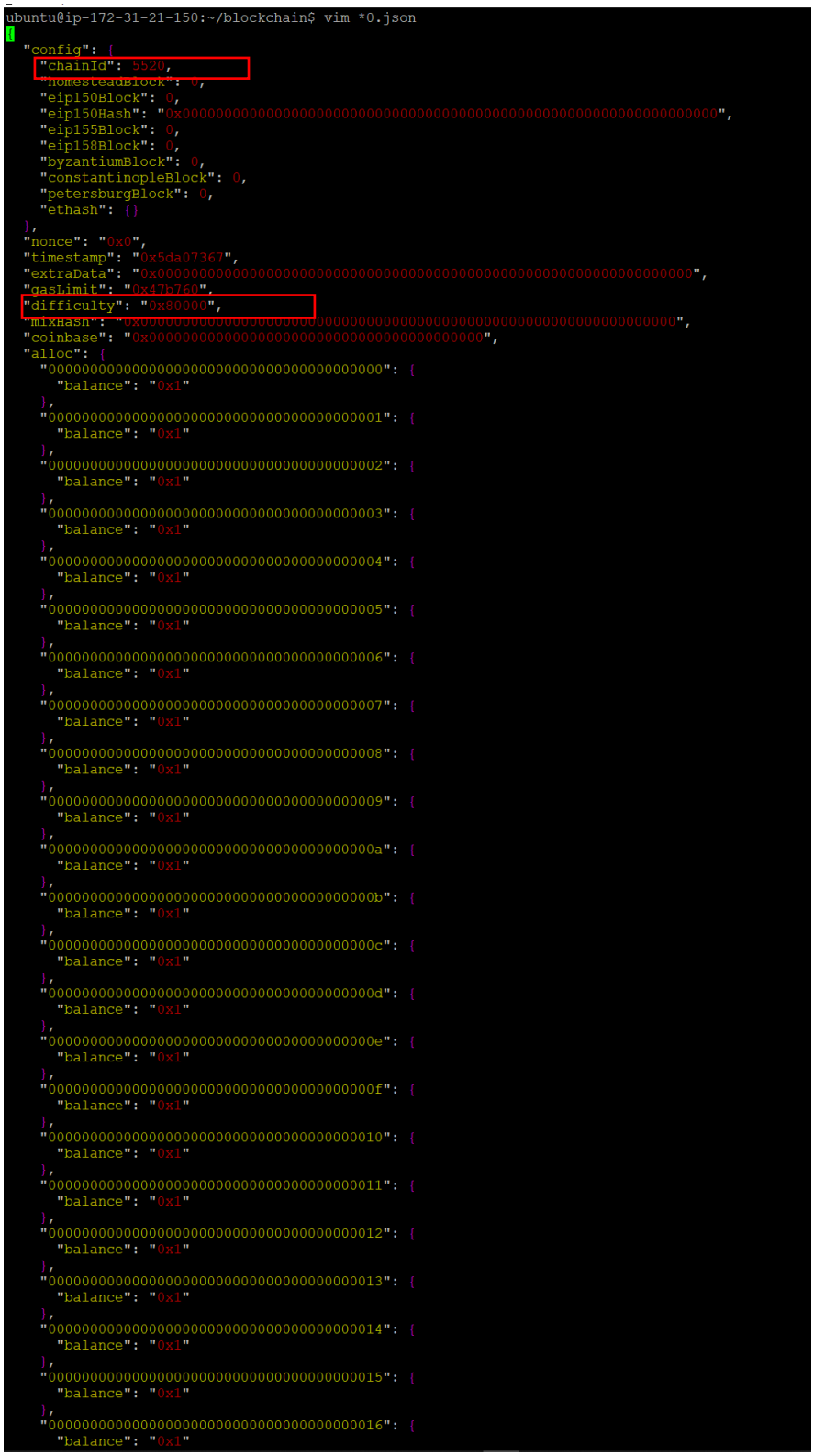
```
Please specify a network name to administer (no spaces, hyphens or capital letters pl
ease)
> ftec5520

Sweet, you can set this via --network=ftec5520 next time!

INFO [10-13|07:45:20.591] Administering Ethereum network           name=ftec5520
INFO [10-13|07:45:20.594] No remote machines to gather stats from

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 ftec5520.json, ftec5520-aleth.json, ftec5520-harmony.json, ftec5520-par
ity.json
>
INFO [10-13|07:45:49.514] Saved native genesis chain spec          path=ftec5520.json
INFO [10-13|07:45:49.515] Saved genesis chain spec                 client=aleth path=
ftec5520-aleth.json
INFO [10-13|07:45:49.517] Saved genesis chain spec                 client=parity path
=ftec5520-parity.json
INFO [10-13|07:45:49.518] Saved genesis chain spec                 client=harmony pat
h=ftec5520-harmony.json
```

Then, press **Ctrl+C** to exit.

Use ls to show the new generated .json files:

```
ubuntu@ip-172-31-21-150:~/blockchain$ ls
ftec5520-aleth.json  ftec5520-harmony.json  ftec5520-parity.json  ftec5520.json
ubuntu@ip-172-31-21-150:~/blockchain$
```

Open the **ftec5520.json** via vim, you will see these contents:

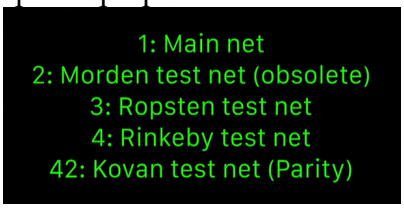


difficulty - This value determines how difficult it will be to mine a block. The lower the value (10-10000 for example) the lower the difficulty. Keeping this low for testing and development purposes will allow blocks to be mined quickly. For comparison, the Ethereum genesis file had a difficulty of 17179869184 which is really high.

To make the further mining step easier, please modify the **"difficulty"** value from the default one to **"10"** (not 0x10 but 10) to make mining in your blockchain network easier to find ether.

Configuration Parameters

1. chainID - The unique identifier of your blockchain. Note that this is a numeric value only.
   Note: avoid using these network ids (1, 2, 3, 4, 42) because they have been used for special purpose.

2. homesteadBlock - This is a version of ethereum which we won't be using. Leave the value as 0.
3. eip155Block - This is a block which you probably don't need to fork. Leave the value as 0.
4. eip158Block - This is a block which you probably don't need to fork. Leave the value as 0.
5. byzantiumBlock - This is the first version of Ethereum's major update. We won't be using this so leave the value as 0.
6. gasLimit - The total amount of gas that can be used in a block.
7. alloc - Allows the allocation of ETH to specific account addresses.
   Note: the content of "alloc" should be public address of the accounts. You've created any account up to now, so please leave it as blank **alloc{}**

Question 1

1. What do you need to change in commands if you want to connect to real public world Ethereum network? (1 mark)

Note: **puppeth** is a private Ethereum network manager, pls consider another approach Please list the main steps and explain why.

Task3: Initialize the blockchain create the primary account

Use **geth init** to initialize a new blockchain database in the directory created just now:

```
geth --datadir ~/blockchain init ~/blockchain/ftec5520.json
```

```
ubuntu@ip-172-31-21-150:~/blockchain$ geth --datadir ~/blockchain/ init ~/blockchain/ftec5520.js
on
INFO [10-13|08:10:47.433] Maximum peer count                       ETH=50 LES=0 total=50
INFO [10-13|08:10:47.434] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.co
mm: no such file or directory"
ERROR[10-13|08:10:47.434] Failed to enumerate USB devices          hub=ledger vendor=11415 failc
ount=1 err="failed to initialize libusb: libusb: unknown error [code -99]"
ERROR[10-13|08:10:47.434] Failed to enumerate USB devices          hub=trezor vendor=21324 failc
ount=1 err="failed to initialize libusb: libusb: unknown error [code -99]"
ERROR[10-13|08:10:47.434] Failed to enumerate USB devices          hub=trezor vendor=4617  failc
ount=1 err="failed to initialize libusb: libusb: unknown error [code -99]"
ERROR[10-13|08:10:47.434] Failed to enumerate USB devices          hub=ledger vendor=11415 failc
ount=2 err="failed to initialize libusb: libusb: unknown error [code -99]"
ERROR[10-13|08:10:47.434] Failed to enumerate USB devices          hub=trezor vendor=21324 failc
ount=2 err="failed to initialize libusb: libusb: unknown error [code -99]"
ERROR[10-13|08:10:47.435] Failed to enumerate USB devices          hub=trezor vendor=4617  failc
ount=2 err="failed to initialize libusb: libusb: unknown error [code -99]"
INFO [10-13|08:10:47.435] Allocated cache and file handles         database=/home/ubuntu/blockch
ain/geth/chaindata cache=16.00MiB handles=16
INFO [10-13|08:10:47.485] Writing custom genesis block
INFO [10-13|08:10:47.499] Persisted trie from memory database      nodes=354 size=50.49KiB time=
2.321123ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [10-13|08:10:47.502] Successfully wrote genesis state         database=chaindata
                  hash=7edfac…3c16e1
INFO [10-13|08:10:47.502] Allocated cache and file handles         database=/home/ubuntu/blockch
ain/geth/lightchaindata cache=16.00MiB handles=16
INFO [10-13|08:10:47.516] Writing custom genesis block
INFO [10-13|08:10:47.524] Persisted trie from memory database      nodes=354 size=50.49KiB time=
1.634712ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [10-13|08:10:47.524] Successfully wrote genesis state         database=lightchaindata
                  hash=7edfac…3c16e1
ubuntu@ip-172-31-21-150:~/blockchain$
```

Or

```
geth --datadir /home/ubuntu/blockchain init \
/home/ubuntu/blockchain/ftec5520.json
```

Use tree command to visualize the files initialized by **geth init** just now:

```
sudo apt install tree
tree .
```

```
ubuntu@ip-172-31-21-150:~/blockchain$ tree
.
├── ftec5520-aleth.json
├── ftec5520-harmony.json
├── ftec5520-parity.json
├── ftec5520.json
├── geth
│   ├── chaindata
│   │   ├── 000001.log
│   │   ├── CURRENT
│   │   ├── LOCK
│   │   ├── LOG
│   │   └── MANIFEST-000000
│   └── lightchaindata
│       ├── 000001.log
│       ├── CURRENT
│       ├── LOCK
│       ├── LOG
│       └── MANIFEST-000000
└── keystore

4 directories, 14 files
ubuntu@ip-172-31-21-150:~/blockchain$
```

We not only need to initialize a new blockchain network, but also need to create the primary account:

```
geth --datadir ~/blockchain account new
```

```
ubuntu@ip-172-31-21-150:~/blockchain$ ls
ftec5520-aleth.json  ftec5520-harmony.json  ftec5520-parity.json  ftec5520.json  geth  keystore
ubuntu@ip-172-31-21-150:~/blockchain$ geth --datadir ~/blockchain/ account new
INFO [10-13|08:13:33.861] Maximum peer count                       ETH=50 LES=0 total=50
INFO [10-13|08:13:33.861] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.co
mm: 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:

Your new key was generated

Public address of the key:   0xFBec10f019B85Bb59c2972412362353CeBaCeDf9
Path of the secret key file: /home/ubuntu/blockchain/keystore/UTC--2019-10-13T08-13-40.191118532
Z--fbec10f019b85bb59c2972412362353cebacedf9

- 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 your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!

ubuntu@ip-172-31-21-150:~/blockchain$
```

Or you can use the full path:

```
geth --datadir /home/ubuntu/blockchain account new
```

You're required to set a password (recommend use **ftec5520** as the password for easier memory) as the password for this account.

Run this command in CLI to enter the Geth console:

```
geth --nousb --cache 512 --ipcpath ~/Library/Ethereum/geth.ipc --
networkid 5520 --datadir blockchain console
```

Or you can use the full path:

```
geth --nousb --cache 512 --ipcpath ~/Library/Ethereum/geth.ipc --
networkid 5520 --datadir /home/ubuntu/blockchain console 2>>ftec552-
part1.log
```

Trick: For enter console quickly avoiding input such a complicated command every time, you may create a **.sh** file startnode.sh to save the command below for fast starting the network:

```
touch startnode.sh && vim startnode.sh
```

Copy and paste the below **geth console** command to the **startnode.sh**

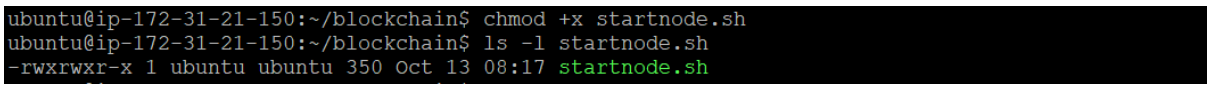
```
geth --nousb --cache 512 \
        --ipcpath ~/Library/Ethereum/geth.ipc \
        --networkid 5520 \
        --datadir /home/ubuntu/blockchain \
        Console
```

(Optional) Redirect the standard error output into a file named ftec5520-lab1.log. Then you can find the log file named *ftec5520-lab1-part1.log* **in the same directory of** *startnode.h*.

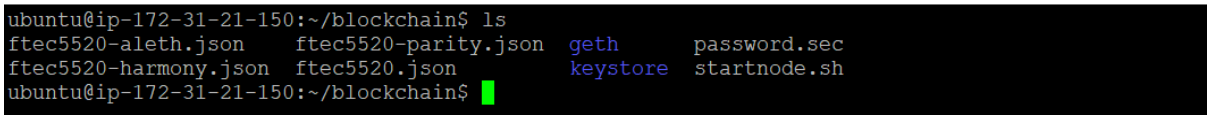
```
geth --nousb --cache 512 \
        --ipcpath ~/Library/Ethereum/geth.ipc \
        --networkid 5520 \
        --datadir /home/ubuntu/blockchain \
        console 2>>ftec-lab1-part1.log
```

Exit the vim editor using ": wq" and give the run permission to startnode.sh

```
chmod +x startnode.sh
```



Run the startnode.sh using "**./** " only if you finished the last step:

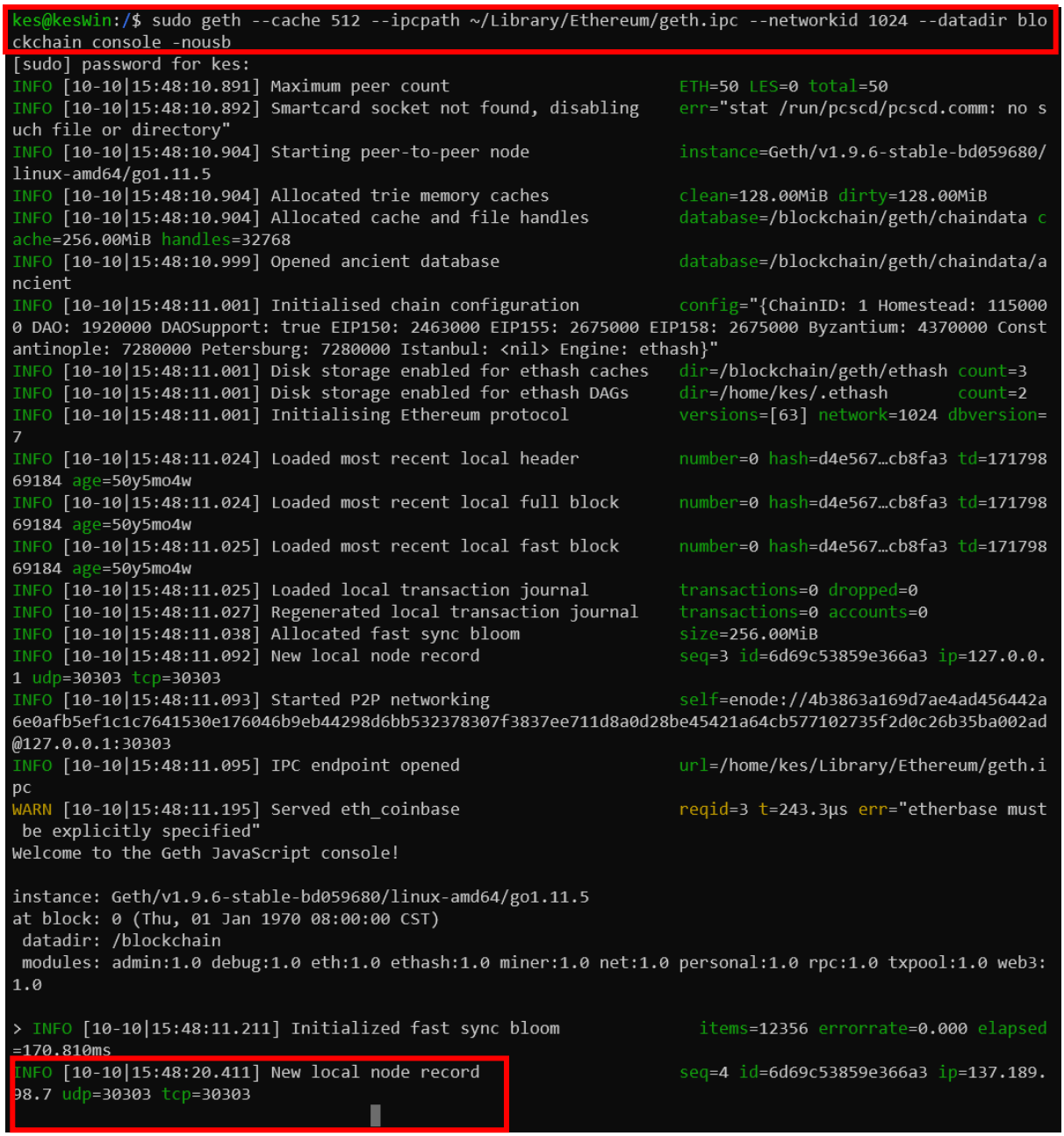



```
./startnode.sh
```

Then you may be required to input the password (ftec5520) of Ubuntu VM you set when you created the VM (if applicable).

Alternatively, you can also use this command directly regardless the last two steps:

```
sh startnode.sh
```

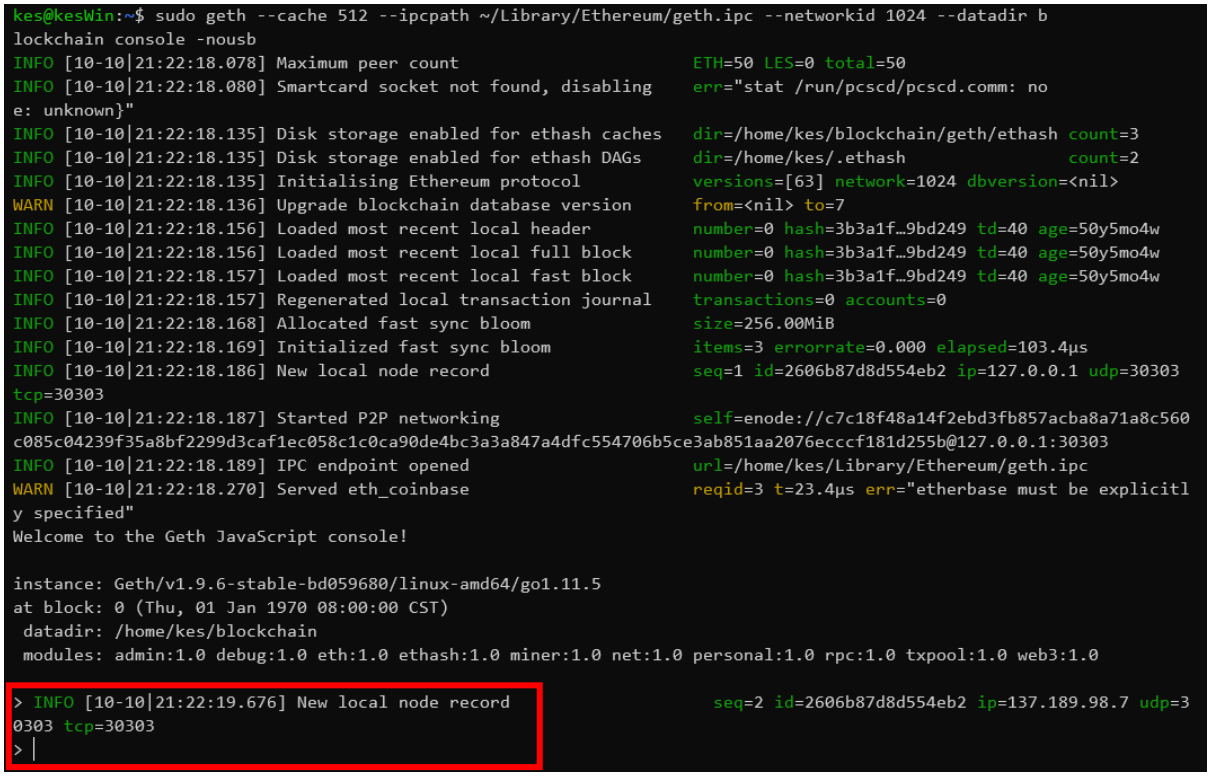


At the first time to run **geth console**, it will create new local node record and keep running. Please press **F5 or Enter and you will enter interactive mode**. Please input exit and you will go back to terminal. Run **startnode.sh** again to enter interactive console mode.

Note: pls ignore the "networkid 1024" in the screenshots, because I didn't use 5520 in this case.

(Optional) If you have redirected the stderr into a file in previous step, you can open a new window, enter the ~/blockchain directory and use tail command to show the progress in real time.

```
tail -f ftec5520-lab1-part1.log
```



Task4: Create Accounts & Check Balances

Task 4.1 Create Accounts:

Now you can input web3 code to operate on your blockchain network.

The following steps will take you through the process of creating new accounts which you can use on your private Ethereum blockchain. These are private accounts that are specific to your blockchain only.

1. View all accounts by running this command in the Geth console:

   `web3.eth.accounts`

   You will see a list of all existing accounts. In this example, there are two accounts.

2. **[Alternative]** In the command line, run account list to show all existing accounts:

   `geth account list`

```
kes@kesWin:~/blockchain$ geth account list
INFO [10-10|21:47:05.522] Maximum peer count                       ETH=50 LES=0 total=50
INFO [10-10|21:47:05.523] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or
directory"
ERROR[10-10|21:47:05.523] Failed to enumerate USB devices          hub=ledger vendor=11415 failcount=1 err="failed t
o initialize libusb: libusb: unknown error [code -99]"
ERROR[10-10|21:47:05.523] Failed to enumerate USB devices          hub=trezor vendor=21324 failcount=1 err="failed t
o initialize libusb: libusb: unknown error [code -99]"
ERROR[10-10|21:47:05.524] Failed to enumerate USB devices          hub=trezor vendor=4617  failcount=1 err="failed t
o initialize libusb: libusb: unknown error [code -99]"
ERROR[10-10|21:47:05.524] Failed to enumerate USB devices          hub=ledger vendor=11415 failcount=2 err="failed t
o initialize libusb: libusb: unknown error [code -99]"
ERROR[10-10|21:47:05.524] Failed to enumerate USB devices          hub=trezor vendor=21324 failcount=2 err="failed t
o initialize libusb: libusb: unknown error [code -99]"
ERROR[10-10|21:47:05.525] Failed to enumerate USB devices          hub=trezor vendor=4617  failcount=2 err="failed t
o initialize libusb: libusb: unknown error [code -99]"
kes@kesWin:~/blockchain$
```

Note: Pls ignore the ERRORs. The errors are due to we don't use hardware wallet here.

3. **[Only if your account were not created]** In the Geth console, run this command to c reate an account:

   ```
   personal.newAccount("Here is my awesome random pass phrase!")
   ```

The value in **" "** is your password phrase of this account. Please keep it safe.

Or you can call this function and follow the printed guidance to steup the password.

```
personal.newAccount()
```

At this step, you are creating a random pass phrase for the generation of your account. Record and keep your pass phrase for this account in case you need it later.

```
> personal.newAccount("Here is my awesome random pass phrase!")
INFO [10-10|21:52:35.729] Your new key was generated               address=0xcf65B09ABA37256800DD10284344fDef36Bf532
2
WARN [10-10|21:52:35.730] Please backup your key file!             path=/home/kes/blockchain/keystore/UTC--2019-10-1
0T13-52-34.037252300Z--cf65b09aba37256800dd10284344fdef36bf5322
WARN [10-10|21:52:35.730] Please remember your password!
"0xcf65b09aba37256800dd10284344fdef36bf5322"
>
```

Note: In this example, the value of "address" is the public account address of this account.

You can create another account using same function but different pass phrase.

```
> personal.newAccount("Here is my another random pass phrase!")
INFO [10-10|21:53:39.344] Your new key was generated               address=0x42f5D8A0ebDC1239B7FFa623AF455f580eD3393
a
WARN [10-10|21:53:39.348] Please backup your key file!             path=/home/kes/blockchain/keystore/UTC--2019-10-1
0T13-53-37.811607500Z--42f5d8a0ebdc1239b7ffa623af455f580ed3393a
WARN [10-10|21:53:39.348] Please remember your password!
"0x42f5d8a0ebdc1239b7ffa623af455f580ed3393a"
>
```

You can observe that the "address" is different from the previous one.

Task 4.2 Check Balances:

The following steps will show you how to check account balances using the Geth CLI and assumes you are in the Geth console when executing these commands.

1. Set the primary account (assuming one account which is account '0' has been created) using

   ```
   var primaryAccount = web3.eth.accounts[0]
   ```

2. View the primary account balance

```
web3.eth.getBalance(primaryAccount)
```

```
> var primaryAccount = web3.eth.accounts[0]
undefined
> web3.eth.getBalance(primaryAccount)
0
>
```

Up to now, the balance of primaryAccount or web3.eth.accounts[0] is still 0.

Task 5: Mining for Ether & Transfer Ether

The following steps will take you through the process of mining for Ether on your own private blockchain which you have created by following the previous steps in this tutorial. However, this will not be real world Ether but only fake Ether only for your blockchain network.

Task 5.1 Mining for Ether:

1. Use **miner.start()** command to mining for ether in your own private network

   You can observe that there're 6 threads for mining ether.

   ```
   miner.start(6)
   ```

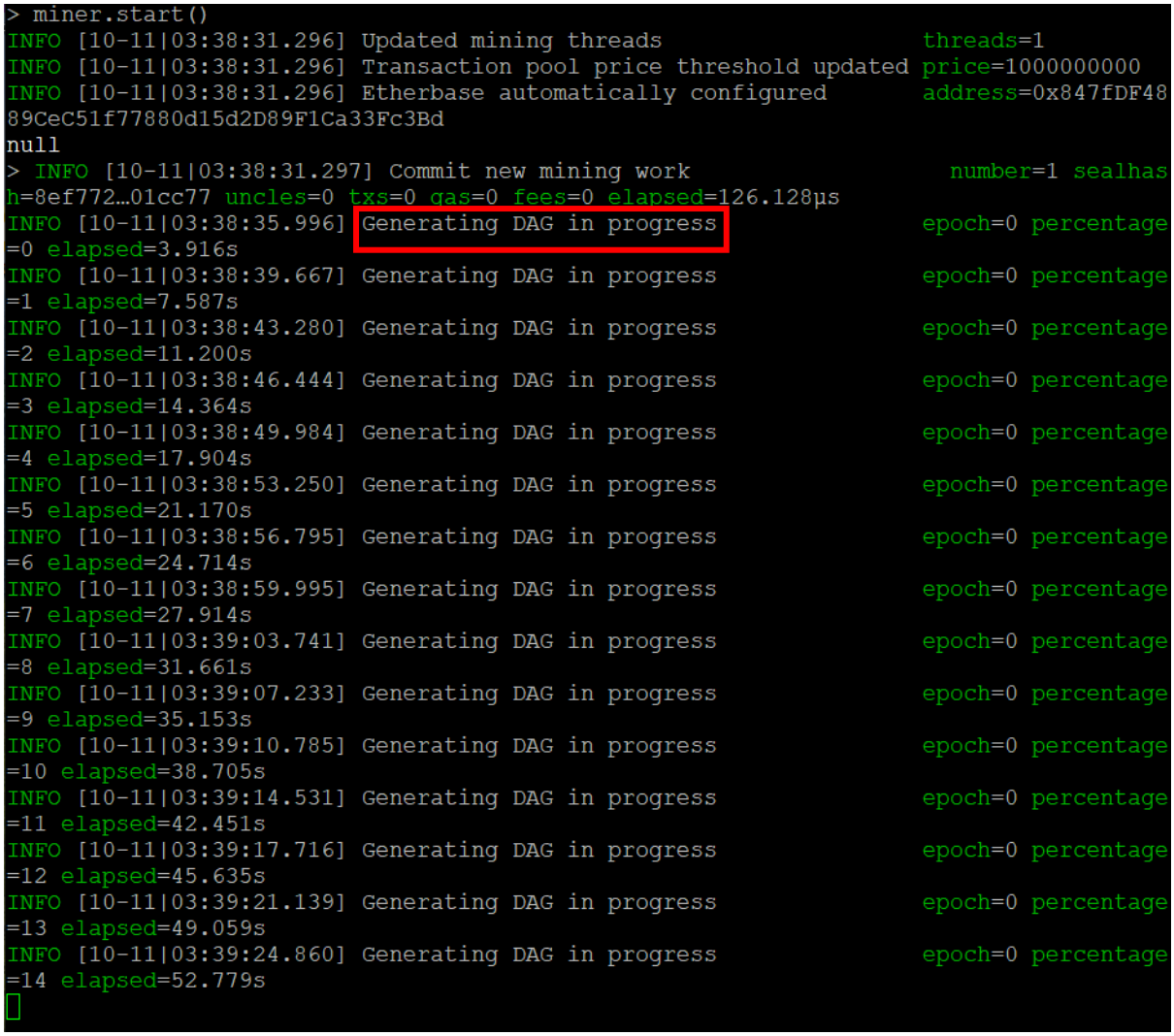
```
> miner.start()
INFO [10-10|22:03:52.174] Updated mining threads                   threads=6
INFO [10-10|22:03:52.174] Transaction pool price threshold updated price=1000000000
INFO [10-10|22:03:52.174] Etherbase automatically configured       address=0xcf65B09ABA37256800DD10284344fDef36Bf532
2
null
> INFO [10-10|22:03:52.175] Commit new mining work                  number=1 sealhash=c89bde…739800 uncles=0 txs=0
gas=0 fees=0 elapsed=560µs
INFO [10-10|22:03:52.839] Successfully sealed new block           number=1 sealhash=c89bde…739800 hash=300cc9…245c0
3 elapsed=663.526ms
INFO [10-10|22:03:52.839] 🔨 mined potential block                 number=1 hash=300cc9…245c03
INFO [10-10|22:03:52.840] Commit new mining work                  number=2 sealhash=7fc602…a569a6 uncles=0 txs=0 ga
s=0 fees=0 elapsed=688.1µs
INFO [10-10|22:03:52.849] Successfully sealed new block           number=2 sealhash=7fc602…a569a6 hash=2e87b1…00e30
4 elapsed=8.953ms
INFO [10-10|22:03:52.849] 🔨 mined potential block                 number=2 hash=2e87b1…00e304
INFO [10-10|22:03:52.849] Mining too far in the future            wait=2s
```

Usually, it will cost a long time a get 1st block ether. If you can see the Another example when haven't get any mining result:

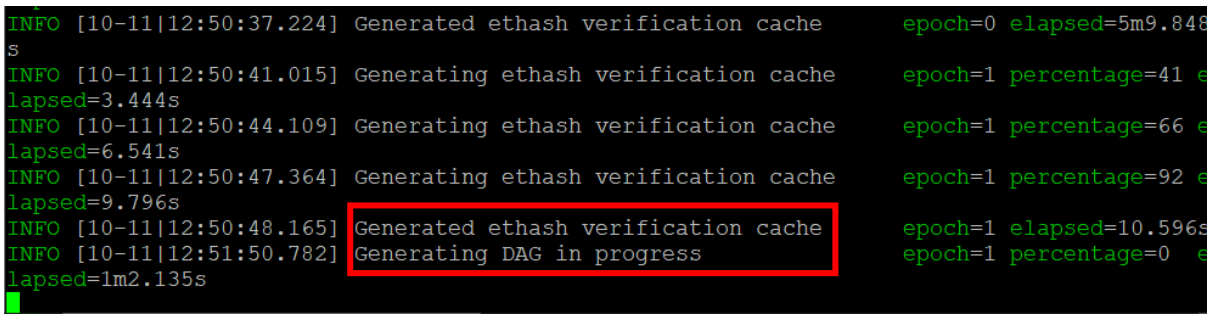
You can use this variable to check the growth of blockchain.

```
web3.eth.blockNumber
```

Note that if you redirect the stderr in previous steps, you won't observe any dynamic output on screen here. Pls use "tail -f ftec55200-lab1-part1.log" to see the real-time progress.



……



The contents in red squares indicate that the mining process is running well.

2. Use miner.stop() command to stop mining.
   Note: cause the screen are scrolling all the time due to mining, you may copy this command to filpboard and then paste it use keyboard shorcut.

   ```
   miner.stop()
   ```

```
33 elapsed=965.308ms
INFO [10-10|22:04:32.433] 🔗 block reached canonical chain          number=32 hash=8f8580…bd6a70
INFO [10-10|22:04:32.433] 🔨 mined potential block                  number=39 hash=e48a14…3a4d33
INFO [10-10|22:04:32.458] Commit new mining work                   number=40 sealhash=67873f…d279ca uncles=0 txs=0 g
as=0 fees=0 elapsed=24.757ms
INFO [10-10|22:04:32.979] Successfully sealed new block            number=40 sealhash=67873f…d279ca hash=4aa9c8…1509
38 elapsed=546.252ms
INFO [10-10|22:04:32.980] 🔗 block reached canonical chain          number=33 hash=7daad7…824233
INFO [10-10|22:04:32.981] 🔨 mined potential block                  number=40 hash=4aa9c8…150938
INFO [10-10|22:04:32.980] Mining too far in the future             wait=2s

> INFO [10-10|22:04:34.982] Commit new mining work                   number=41 sealhash=a63142…9cbbd3 uncles=0 txs=0
 gas=0 fees=0 elapsed=2.001s
INFO [10-10|22:04:35.147] Successfully sealed new block            number=41 sealhash=a63142…9cbbd3 hash=1dc230…ead5
4e elapsed=165.198ms
INFO [10-10|22:04:35.148] 🔗 block reached canonical chain          number=34 hash=a271a5…f8ab2f
INFO [10-10|22:04:35.148] 🔨 mined potential block                  number=41 hash=1dc230…ead54e
INFO [10-10|22:04:35.149] Commit new mining work                   number=42 sealhash=75d7fd…f73055 uncles=0 txs=0 g
as=0 fees=0 elapsed=124.8µs
INFO [10-10|22:04:35.265] Successfully sealed new block            number=42 sealhash=75d7fd…f73055 hash=1bb4f4…7b2c
24 elapsed=116.521ms
INFO [10-10|22:04:35.266] 🔗 block reached canonical chain          number=35 hash=602c62…b1f00e
INFO [10-10|22:04:35.266] 🔨 mined potential block                  number=42 hash=1bb4f4…7b2c24
INFO [10-10|22:04:35.266] Commit new mining work                   number=43 sealhash=8a36b5…ad7c1d uncles=0 txs=0 g
as=0 fees=0 elapsed=154µs
INFO [10-10|22:04:35.391] Successfully sealed new block            number=43 sealhash=8a36b5…ad7c1d hash=296e0f…1f5a
f9 elapsed=124.755ms
INFO [10-10|22:04:35.391] 🔗 block reached canonical chain          number=36 hash=b92d59…bf04be
INFO [10-10|22:04:35.391] 🔨 mined potential block                  number=43 hash=296e0f…1f5af9
INFO [10-10|22:04:35.391] Mining too far in the future             wait=2s
> miner.stop()
null
>
```

3. Use get balance function again to show the mining result:

```
web3.eth.getBalance(primaryAccount)
```

```
>web3.eth.getBalance(primaryAccount)
Null 12
>
```

Question 2

2.1 Please describe the process of mining, what are the main steps for finding a new block with Ether?

2.2 What's the content of a new mined Ethereum block, and how to identify a block in the blockchain network except using block number?

Reference

a) **How to set up Ethereum blockchain**
https://arctouch.com/blog/how-to-set-up-ethereum-blockchain/
b) **Setup private Ethereum blockchain in AWS**
https://medium.com/nxtplus/setup-private-ethereum-blockchain-in-aws-amazon-web-services-or-gcp-google-cloud-platform-abfaae779f6a
c) **Setup private Ethereum blockchain and deploy your 1st Solidity Smart Contract**
https://medium.com/blockchainbistro/set-up-a-private-ethereum-blockchain-and-deploy-your-first-solidity-smart-contract-on-the-caa8334c343d
d) **Setting up and running a private Ethereum blockchain on ubuntu**
https://steemit.com/ethereum/@nphacker/setting-up-and-running-a-private-ethereum-blockchain-on-ubuntu
e) **Setup a fully synced blockchain node**
https://www.freecodecamp.org/news/ethereum-69-how-to-set-up-a-fully-synced-blockchain-node-in-10-mins-f6318d7aad40/
f) **Setup the Go Ethereum geth version in Ubuntu Linux**
https://medium.com/@priyalwalpita/setting-up-the-go-ethereum-geth-environment-in-ubuntu-linux-67c09706b42
g) **Two-nodes setup of private Ethereum on AWS with contract deployment**
Part1: https://blockgeeks.com/two-node-setup-of-a-private-ethereum/
Part2: https://blockgeeks.com/two-node-setup-of-a-private-ethereum-on-aws-with-contract-deployment-part-2/
h) **Aws blockchain template**
https://docs.aws.amazon.com/blockchain-templates/latest/developerguide/blockchain-templates-ethereum.html
i) **Ethereum for web developers**
https://medium.com/@mvmurthy/ethereum-for-web-developers-890be23d1d0c
j) **Full Stack Hello World Voting dApp tutorial Part1**
https://medium.com/@mvmurthy/full-stack-hello-world-voting-ethereum-dapp-tutorial-part-1-40d2d0d807c2
k) **Full Stack Hello World Voting dApp tutorial part2**
https://medium.com/@mvmurthy/full-stack-hello-world-voting-ethereum-dapp-tutorial-part-1-40d2d0d807c2
l) **Full Stack Hello World Voting dApp tutorial Part3**
https://medium.com/@mvmurthy/full-stack-hello-world-voting-ethereum-dapp-tutorial-part-3-331c2712c9df