

**The Chinese University of Hong Kong**  
Department of Financial Technology  
**FTEC5520– Applied Blockchain & Cryptocurrency**

**Lab1 Exercise**

**Ethereum Infrastructure Setup & Practice – Part1**

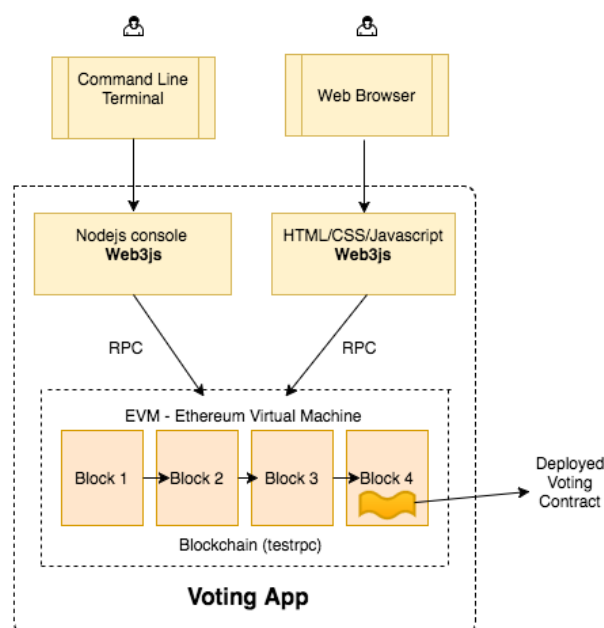
**Lab object:**

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 contract 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 for voting 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 please)
> 
```

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 please)
> 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-parity.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 path=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
```

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

```
ubuntu@ip-172-31-21-150:~/blockchain$ vim *.json
"config": {
  "chainId": 5520,
  "homesteadBlock": 0,
  "eip150Block": 0,
  "eip150Hash": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "eip155Block": 0,
  "eip158Block": 0,
  "byzantiumBlock": 0,
  "constantinopleBlock": 0,
  "petersburgBlock": 0,
  "ethash": {}
},
"nonce": "0x0",
"timestamp": "0x5da07367",
"extraData": "0x0000000000000000000000000000000000000000000000000000000000000000",
"gasLimit": "0x47b760",
"difficulty": "0x80000",
"mixHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
"coinbase": "0x0000000000000000000000000000000000000000000000000000000000000000",
"alloc": {
  "0000000000000000000000000000000000000000000000000000000000000000": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000001": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000002": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000003": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000004": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000005": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000006": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000007": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000008": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000009": {
    "balance": "0x1"
  },
  "000000000000000000000000000000000000000000000000000000000000000a": {
    "balance": "0x1"
  },
  "000000000000000000000000000000000000000000000000000000000000000b": {
    "balance": "0x1"
  },
  "000000000000000000000000000000000000000000000000000000000000000c": {
    "balance": "0x1"
  },
  "000000000000000000000000000000000000000000000000000000000000000d": {
    "balance": "0x1"
  },
  "000000000000000000000000000000000000000000000000000000000000000e": {
    "balance": "0x1"
  },
  "000000000000000000000000000000000000000000000000000000000000000f": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000010": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000011": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000012": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000013": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000014": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000015": {
    "balance": "0x1"
  },
  "0000000000000000000000000000000000000000000000000000000000000016": {
    "balance": "0x1"
  }
}
```

**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.

```
{
  "config": {
    "chainId": 5520,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip150Hash": "0x0000000000000000000000000000000000000000000000000000000000000000",
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "ethash": {}
  },
  "nonce": "0x0",
  "timestamp": "0x5da07367",
  "extraData": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "gasLimit": "0x47b760",
  "difficulty": "10",
  "mixHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "coinbase": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "alloc": {
```

#### 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.

```
1: Main net
2: Morden test net (obsolete)
3: Ropsten test net
4: Rinkeby test net
42: Kovan test net (Parity)
```

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

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.json
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.conf: no such file or directory"
ERROR[10-13|08:10:47.434] Failed to enumerate USB devices          hub=ledger vendor=11415 failed
count=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 failed
count=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 failed
count=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 failed
count=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 failed
count=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 failed
count=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/blockchain/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 gcsiz=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/blockchain/chaindata 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 gcsiz=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$
```

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$

```

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
```

**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 blockchain \  
console
```

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

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

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

```
ubuntu@ip-172-31-21-150:~/blockchain$ chmod +x startnode.sh  
ubuntu@ip-172-31-21-150:~/blockchain$ ls -l startnode.sh  
-rwxrwxr-x 1 ubuntu ubuntu 350 Oct 13 08:17 startnode.sh
```

Run the startnode.sh using “./” only if you finished the last step:

```
ubuntu@ip-172-31-21-150:~/blockchain$ ls  
ftec5520-aleth.json  ftec5520-parity.json  geth      password.sec  
ftec5520-harmony.json ftec5520.json         keystore  startnode.sh  
ubuntu@ip-172-31-21-150:~/blockchain$
```

```
./startnode.sh
```

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

**Not:** Alternatively, you can also use this command directly regardless the last two steps:

```
sh startnode.sh
```



```
kes@kesWin:/$ sudo geth --cache 512 --ipcpath ~/Library/Ethereum/geth.ipc --networkid 1024 --datadir blockchain console -nousb
[sudo] password for kes:
INFO [10-10|15:48:10.891] Maximum peer count           ETH=50 LES=0 total=50
INFO [10-10|15:48:10.892] Smartcard socket not found, disabling err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [10-10|15:48:10.904] Starting peer-to-peer node     instance=Geth/v1.9.6-stable-bd059680/linux-amd64/go1.11.5
INFO [10-10|15:48:10.904] Allocated trie memory caches   clean=128.00MiB dirty=128.00MiB
INFO [10-10|15:48:10.904] Allocated cache and file handles database=/blockchain/geth/chaindata/cache=256.00MiB handles=32768
INFO [10-10|15:48:10.999] Opened ancient database        database=/blockchain/geth/chaindata/ancient
INFO [10-10|15:48:11.001] Initialised chain configuration config="{ChainID: 1 Homestead: 1150000 DAO: 1920000 DAOSupport: true EIP150: 2463000 EIP155: 2675000 EIP158: 2675000 Byzantium: 4370000 Constantinople: 7280000 Petersburg: 7280000 Istanbul: <nil> Engine: ethash}"
INFO [10-10|15:48:11.001] Disk storage enabled for ethash caches dir=/blockchain/geth/ethash count=3
INFO [10-10|15:48:11.001] Disk storage enabled for ethash DAGs dir=/home/kes/.ethash count=2
INFO [10-10|15:48:11.001] Initialising Ethereum protocol versions=[63] network=1024 dbversion=7
INFO [10-10|15:48:11.024] Loaded most recent local header number=0 hash=d4e567...cb8fa3 td=17179869184 age=50y5mo4w
INFO [10-10|15:48:11.024] Loaded most recent local full block number=0 hash=d4e567...cb8fa3 td=17179869184 age=50y5mo4w
INFO [10-10|15:48:11.025] Loaded most recent local fast block number=0 hash=d4e567...cb8fa3 td=17179869184 age=50y5mo4w
INFO [10-10|15:48:11.025] Loaded local transaction journal transactions=0 dropped=0
INFO [10-10|15:48:11.027] Regenerated local transaction journal transactions=0 accounts=0
INFO [10-10|15:48:11.038] Allocated fast sync bloom size=256.00MiB
INFO [10-10|15:48:11.092] New local node record seq=3 id=6d69c53859e366a3 ip=127.0.0.1 udp=30303 tcp=30303
INFO [10-10|15:48:11.093] Started P2P networking self=enode://4b3863a169d7ae4ad456442a6e0afb5ef1c1c7641530e176046b9eb44298d6bb532378307f3837ee711d8a0d28be45421a64cb577102735f2d0c26b35ba002ad@127.0.0.1:30303
INFO [10-10|15:48:11.095] IPC endpoint opened url=/home/kes/Library/Ethereum/geth.ipc
WARN [10-10|15:48:11.195] Served eth_coinbase reqid=3 t=243.3µs err="etherbase must be explicitly specified"
Welcome to the Geth JavaScript console!

instance: Geth/v1.9.6-stable-bd059680/linux-amd64/go1.11.5
at block: 0 (Thu, 01 Jan 1970 08:00:00 CST)
datadir: /blockchain
modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

> INFO [10-10|15:48:11.211] Initialized fast sync bloom items=12356 errorrate=0.000 elapsed=170.810ms
INFO [10-10|15:48:20.411] New local node record seq=4 id=6d69c53859e366a3 ip=137.189.98.7 udp=30303 tcp=30303
```

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, cause I didn’t use 5520 in this case.

```

kes@kesWin:~$ sudo geth --cache 512 --ipcpath ~/Library/Ethereum/geth.ipc --networkid 1024 --datadir b
lockchain console -nouns
INFO [10-10|21:22:18.078] Maximum peer count           ETH=50 LES=0 total=50
INFO [10-10|21:22:18.080] Smartcard socket not found, disabling err="stat /run/pcscd/pcscd.comm: no
e: unknown}"
INFO [10-10|21:22:18.135] Disk storage enabled for ethash caches dir=/home/kas/blockchain/geth/ethash count=3
INFO [10-10|21:22:18.135] Disk storage enabled for ethash DAGs  dir=/home/kas/.ethash count=2
INFO [10-10|21:22:18.135] Initialising Ethereum protocol      versions=[63] network=1024 dbversion=<nil>
WARN [10-10|21:22:18.136] Upgrade blockchain database version  from=<nil> to=7
INFO [10-10|21:22:18.156] Loaded most recent local header      number=0 hash=3b3a1f...9bd249 td=40 age=50y5mo4w
INFO [10-10|21:22:18.156] Loaded most recent local full block  number=0 hash=3b3a1f...9bd249 td=40 age=50y5mo4w
INFO [10-10|21:22:18.157] Loaded most recent local fast block  number=0 hash=3b3a1f...9bd249 td=40 age=50y5mo4w
INFO [10-10|21:22:18.157] Regenerated local transaction journal transactions=0 accounts=0
INFO [10-10|21:22:18.168] Allocated fast sync bloom           size=256.00MiB
INFO [10-10|21:22:18.169] Initialized fast sync bloom          items=3 errorrate=0.000 elapsed=103.4µs
INFO [10-10|21:22:18.186] New local node record                seq=1 id=2606b87d8d554eb2 ip=127.0.0.1 udp=30303
tcp=30303
INFO [10-10|21:22:18.187] Started P2P networking               self=enode://c7c18f48a14f2ebd3fb857acba8a71a8c560
c085c04239f35a8bf2299d3caf1ec058c1c0ca90de4bc3a3a847a4dfc554706b5ce3ab851aa2076eccc181d255b@127.0.0.1:30303
INFO [10-10|21:22:18.189] IPC endpoint opened                  url=/home/kas/Library/Ethereum/geth.ipc
WARN [10-10|21:22:18.270] Served eth_coinbase                  reqid=3 t=23.4µs err="etherbase must be explicitl
y specified"
Welcome to the Geth JavaScript console!

instance: Geth/v1.9.6-stable-bd059680/linux-amd64/go1.11.5
at block: 0 (Thu, 01 Jan 1970 08:00:00 CST)
datadir: /home/kas/blockchain
modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

> INFO [10-10|21:22:19.676] New local node record                seq=2 id=2606b87d8d554eb2 ip=137.189.98.7 udp=3
0303 tcp=30303
> |

```

## 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.

```

> web3.eth.accounts
["0xc6f5b09aba37256800dd10284344fdef36bf5322", "0x42f5d8a0ebdc1239b7ffa623af455f580ed3393a"]
> |

```

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
directory"                      err="stat /run/pcscd/pcscd.comm: no such file or
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 create 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.

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=0xcf65B09ABA3725680DD10284344fDef368f532
2
WARN [10-10|21:52:35.730] Please backup your key file!    path=/home/kas/blockchain/keystore/UTC--2019-10-1
0T13-52-34.037252300Z--cf65b09aba3725680dd10284344fdef36bf5322
WARN [10-10|21:52:35.730] Please remember your password!
"0xcf65b09aba3725680dd10284344fdef36bf5322"
>

```

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/kas/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()**

```

> 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:

```

> miner.start()
INFO [10-11|03:38:31.296] Updated mining threads          threads=1
INFO [10-11|03:38:31.296] Transaction pool price threshold updated price=1000000000
INFO [10-11|03:38:31.296] Etherbase automatically configured    address=0x847fDF48
89CeC51f77880d15d2D89F1Ca33Fc3Bd
null
> INFO [10-11|03:38:31.297] Commit new mining work          number=1 sealhas
h=8ef772...01cc77 uncles=0 txs=0 gas=0 fees=0 elapsed=126.128ps
INFO [10-11|03:38:35.996] Generating DAG in progress      epoch=0 percentage
=0 elapsed=3.916s
INFO [10-11|03:38:39.667] Generating DAG in progress      epoch=0 percentage
=1 elapsed=7.587s
INFO [10-11|03:38:43.280] Generating DAG in progress      epoch=0 percentage
=2 elapsed=11.200s
INFO [10-11|03:38:46.444] Generating DAG in progress      epoch=0 percentage
=3 elapsed=14.364s
INFO [10-11|03:38:49.984] Generating DAG in progress      epoch=0 percentage
=4 elapsed=17.904s
INFO [10-11|03:38:53.250] Generating DAG in progress      epoch=0 percentage
=5 elapsed=21.170s
INFO [10-11|03:38:56.795] Generating DAG in progress      epoch=0 percentage
=6 elapsed=24.714s
INFO [10-11|03:38:59.995] Generating DAG in progress      epoch=0 percentage
=7 elapsed=27.914s
INFO [10-11|03:39:03.741] Generating DAG in progress      epoch=0 percentage
=8 elapsed=31.661s
INFO [10-11|03:39:07.233] Generating DAG in progress      epoch=0 percentage
=9 elapsed=35.153s
INFO [10-11|03:39:10.785] Generating DAG in progress      epoch=0 percentage
=10 elapsed=38.705s
INFO [10-11|03:39:14.531] Generating DAG in progress      epoch=0 percentage
=11 elapsed=42.451s
INFO [10-11|03:39:17.716] Generating DAG in progress      epoch=0 percentage
=12 elapsed=45.635s
INFO [10-11|03:39:21.139] Generating DAG in progress      epoch=0 percentage
=13 elapsed=49.059s
INFO [10-11|03:39:24.860] Generating DAG in progress      epoch=0 percentage
=14 elapsed=52.779s

```

.....

```

INFO [10-11|12:50:37.224] Generated ethash verification cache epoch=0 elapsed=5m9.848
s
INFO [10-11|12:50:41.015] Generating ethash verification cache epoch=1 percentage=41 e
lapsed=3.444s
INFO [10-11|12:50:44.109] Generating ethash verification cache epoch=1 percentage=66 e
lapsed=6.541s
INFO [10-11|12:50:47.364] Generating ethash verification cache epoch=1 percentage=92 e
lapsed=9.796s
INFO [10-11|12:50:48.165] Generated ethash verification cache epoch=1 elapsed=10.596s
INFO [10-11|12:51:50.782] Generating DAG in progress      epoch=1 percentage=0 e
lapsed=1m2.135s

```

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
INFO [10-10|22:04:32.433] ⚡ mined potential block
INFO [10-10|22:04:32.458] Commit new mining work
as=0 fees=0 elapsed=24.757ms
INFO [10-10|22:04:32.979] Successfully sealed new block
38 elapsed=546.252ms
INFO [10-10|22:04:32.980] ⚡ block reached canonical chain
INFO [10-10|22:04:32.981] ⚡ mined potential block
INFO [10-10|22:04:32.980] Mining too far in the future

> INFO [10-10|22:04:34.982] Commit new mining work
gas=0 fees=0 elapsed=2.001s
INFO [10-10|22:04:35.147] Successfully sealed new block
4e elapsed=165.198ms
INFO [10-10|22:04:35.148] ⚡ block reached canonical chain
INFO [10-10|22:04:35.148] ⚡ mined potential block
INFO [10-10|22:04:35.149] Commit new mining work
as=0 fees=0 elapsed=124.8µs
INFO [10-10|22:04:35.265] Successfully sealed new block
24 elapsed=116.521ms
INFO [10-10|22:04:35.266] ⚡ block reached canonical chain
INFO [10-10|22:04:35.266] ⚡ mined potential block
INFO [10-10|22:04:35.266] Commit new mining work
as=0 fees=0 elapsed=154µs
INFO [10-10|22:04:35.391] Successfully sealed new block
f9 elapsed=124.755ms
INFO [10-10|22:04:35.391] ⚡ block reached canonical chain
INFO [10-10|22:04:35.391] ⚡ mined potential block
INFO [10-10|22:04:35.391] Mining too far in the future
> miner.stop()
null
> |

```

```

number=32 hash=8f8580...bd6a70
number=39 hash=e48a14...3a4d33
number=40 sealhash=67873f...d279ca uncles=0 txs=0 g
number=40 sealhash=67873f...d279ca hash=4aa9c8...1509
number=33 hash=7daad7...824233
number=40 hash=4aa9c8...150938
wait=2s
number=41 sealhash=a63142...9cbbd3 uncles=0 txs=0
number=41 sealhash=a63142...9cbbd3 hash=1dc230...ead5
number=34 hash=a271a5...f8ab2f
number=41 hash=1dc230...ead54e
number=42 sealhash=75d7fd...f73055 uncles=0 txs=0 g
number=42 sealhash=75d7fd...f73055 hash=1bb4f4...7b2c
number=35 hash=602c62...b1f00e
number=42 hash=1bb4f4...7b2c24
number=43 sealhash=8a36b5...ad7c1d uncles=0 txs=0 g
number=43 sealhash=8a36b5...ad7c1d hash=296e0f...1f5a
number=36 hash=b92d59...bf04be
number=43 hash=296e0f...1f5af9
wait=2s

```

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) **Install Ethereum on Ubuntu 16.04**  
<https://retrospect.blog/install-ethereum-on-ubuntu-16-4-xenial/>
- d) **Setup private Ethereum blockchain and deploy your 1<sup>st</sup> Solidity Smart Contract**  
<https://medium.com/blockchainbistro/set-up-a-private-ethereum-blockchain-and-deploy-your-first-solidity-smart-contract-on-the-aea8334c343d>
- e) **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>
- f) **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/>
- g) **Setting up the Go Ethereum geth version in Ubuntu Linux**  
<https://medium.com/@priyalwalpita/setting-up-the-go-ethereum-geth-environment-in-ubuntu-linux-67c09706b42>
- h) **Two-nodes setup of private Ethereum on AWS with contract deployment**  
<https://blockgeeks.com/two-node-setup-of-a-private-ethereum/>  
<https://blockgeeks.com/two-node-setup-of-a-private-ethereum-on-aws-with-contract-deployment-part-2/>
- i) **Aws blockchain template**  
<https://docs.aws.amazon.com/blockchain-templates/latest/developerguide/blockchain-templates-ethereum.html>
- j) **Ethereum for web developers**  
<https://medium.com/@mvmurthy/ethereum-for-web-developers-890be23d1d0c>
- k) **Full Stack Hello World Voting dApp tutorial part2**  
<https://medium.com/@mvmurthy/full-stack-hello-world-voting-ethereum-dapp-tutorial-part-1-40d2d0d807c2>