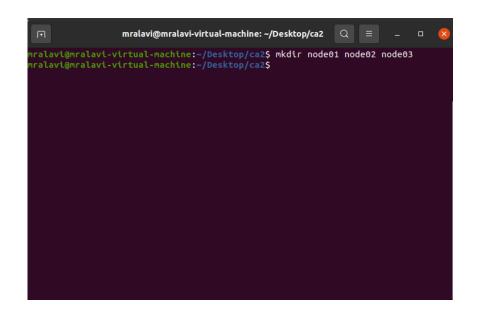
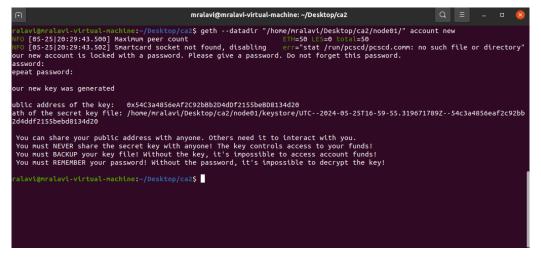
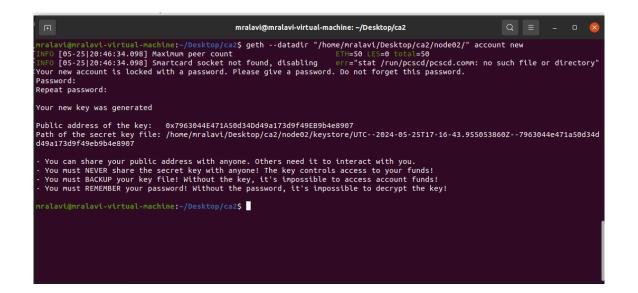
CA2 – Part 2

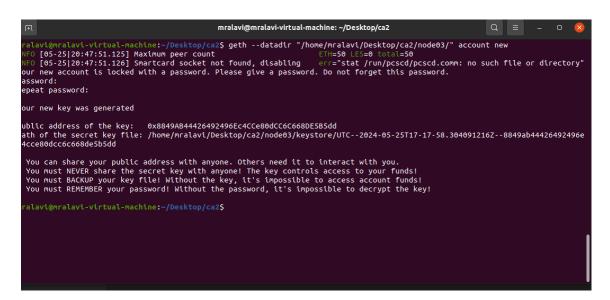
Mohammadreza Alavi - 810100253

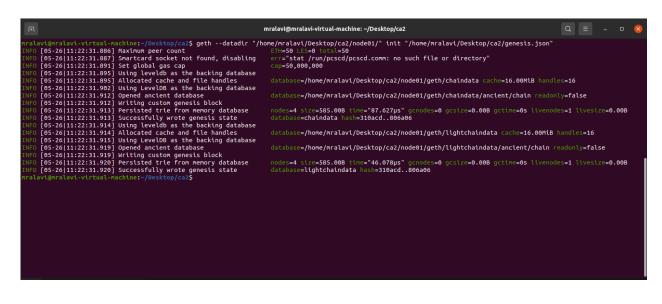
```
"config": {
            "chainId": 15,
            "homesteadBlock": 0,
            "eip155Block": 0,
            "eip158Block": 0
        "difficulty": "400000",
        "gasLimit": "2100000",
        "alloc": {
11
            "[Account #1 Address]": {"balance": "1000000000810100253"},
            "[Account #2 Address]": {"balance": "2000000000810100253"},
12
            "[Account #3 Address]": {"balance": "1500000000810100253"}
13
14
15
```





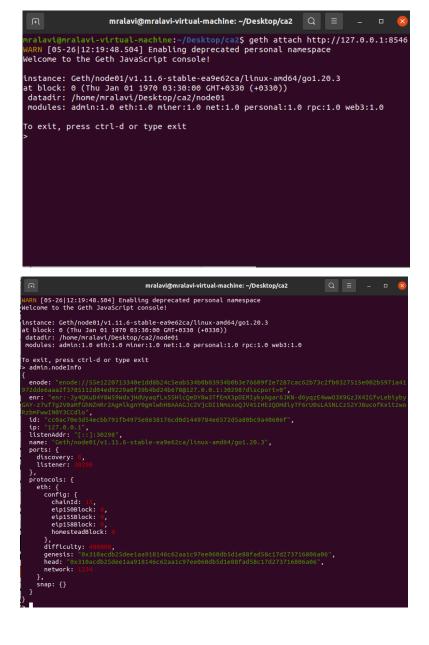




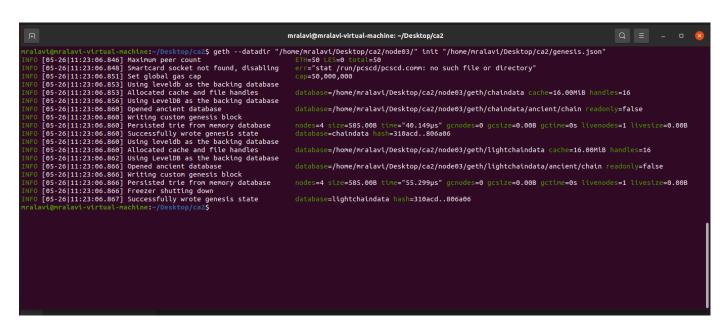


```
| The control of the
```

```
| The content of the
```

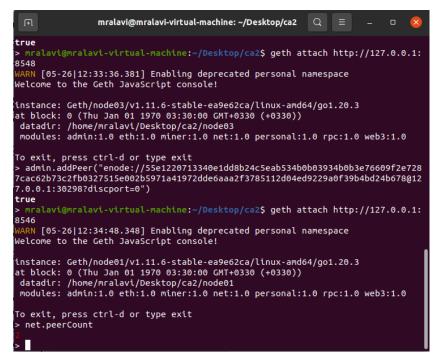


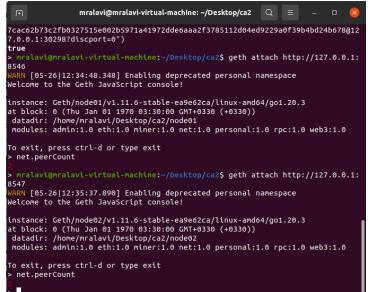
```
mralavi@mralavi-virtual-machine: ~/Desktop/ca2
                                                                                                                                                                                                                                                                                             Q = -
 mralavi@mralavi-virtual-machine:~/Desktop/ca2$ geth --datadir "/home/mralavi/Desktop/ca2/node02/" init/home/mralavi/Desktop/ca2/genesis.json
invalid command: "init/home/mralavi/Desktop/ca2$genesis.json"
mralavi@mralavi-virtual-machine:~/Desktop/ca2$ geth --datadir "/home/mralavi/Desktop/ca2/node02/" init "/home/mralavi/Desktop/ca2/genesis.json"
database=/home/mralavi/Desktop/ca2/node02/geth/chaindata cache=16.00MiB handles=16
                                                                                                                                   database=/home/mralavi/Desktop/ca2/node02/geth/chaindata/ancient/chain readonly=false
         "Vignralavi-virtual-Machine:-/Desktop/ca2$ geth --datadir "/
[05-26|11:21:24.874| Maxhuum peer count
[05-26|11:21:24.875] Snartcard socket not found, disabling
[05-26|11:21:24.879] Set global gas cap
[05-26|11:21:24.880] Sing leveldb as the backing database
[05-26|11:21:24.880] Allocated cache and file handles
[05-26|11:21:24.885] Using LevelDB as the backing database
[05-26|11:21:24.885] Opened ancient database
[05-26|11:21:24.886] Writing custom genesis block
[05-26|11:21:24.887] Successfully wrote genesis state
[05-26|11:21:24.888] Successfully wrote genesis state
[05-26|11:21:24.888] Successfully wrote genesis state
[05-26|11:21:24.888] Using leveldb as the backing database
[05-26|11:21:24.888] Using LevelDB as the backing database
                                                                                                                                   database=/home/mralavi/Desktop/ca2/node02/geth/chaindata cache=16.00MiB handles=16
                                                                                                                                   database=/home/mralavi/Desktop/ca2/node02/geth/chaindata/ancient/chain_readonly=false
                                                                                                                                   nodes=4 size=585.00B time="147.455µs" q
database=chaindata hash=310acd..806a06
                                                                                                                                                                                  ne="147.455µs" gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
                                                                                                                                   database=/home/mralavi/Desktop/ca2/node02/geth/lightchaindata_cache=16.00MiB_handles=16
         [05-26]11:21:24.893] Attocated cache and file handles [05-26]11:21:24.891] Using LevelDB as the backing database [05-26]11:21:24.894] Opened ancient database [05-26]11:21:24.894] Writing custom genesis block [05-26]11:21:24.894] Persisted trie from memory database [05-26]11:21:24.895] Successfully wrote genesis state avigmralavi-virtual-machine:-/Desktop/ca25
                                                                                                                                    database=/home/mralavi/Desktop/ca2/node02/geth/lightchaindata/ancient/chain readonly=false
                                                                                                                                   nodes=4 size=585.00B time="44.107µs" gcnodedatabase=lightchaindata hash=310acd..806a06
                                                                                                                                                                                                                     odes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
```



```
mralavi@mralavi-virtual-machine: ~/Desktop/ca2
        eip158Block:
        homesteadBlock:
      difficulty:
      genesis: "0x310acdb25dee1aa918146c62aa1c97ee060db5d1e88fad58c17d27371680
      head: "0x310acdb25dee1aa918146c62aa1c97ee060db5d1e88fad58c17d273716806a0
      network:
    snap: {}
 mralavi@mralavi-virtual-machine:~/Desktop/ca2$ geth attach http://127.0.0.1:
8547
    [05-26|12:32:20.651] Enabling deprecated personal namespace
Welcome to the Geth JavaScript console!
instance: Geth/node02/v1.11.6-stable-ea9e62ca/linux-amd64/go1.20.3
at block: 0 (Thu Jan 01 1970 03:30:00 GMT+0330 (+0330))
datadir: /home/mralavi/Desktop/ca2/node02
 modules: admin:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 web3:1.0
To exit, press ctrl-d or type exit
 admin.addPeer("enode://55e1220713340e1dd8b24c5eab534b0b03934b0b3e76609f2e728
7cac62b73c2fb0327515e002b5971a41972dde6aaa2f3785112d04ed9229a0f39b4bd24b678@12
7.0.0.1:30298?discport=0")
```

```
Q ≡
                 mralavi@mralavi-virtual-machine: ~/Desktop/ca2
 ARN [05-26|12:32:20.651] Enabling deprecated personal namespace
Welcome to the Geth JavaScript console!
instance: Geth/node02/v1.11.6-stable-ea9e62ca/linux-amd64/go1.20.3
at block: 0 (Thu Jan 01 1970 03:30:00 GMT+0330 (+0330))
datadir: /home/mralavi/Desktop/ca2/node02
modules: admin:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 web3:1.0
To exit, press ctrl-d or type exit
admin.addPeer("enode://55e1220713340e1dd8b24c5eab534b0b03934b0b3e76609f2e728
7cac62b73c2fb0327515e002b5971a41972dde6aaa2f3785112d04ed9229a0f39b4bd24b678@12
7.0.0.1:30298?discport=0")
true
• mralavi@mralavi-virtual-machine:~/Desktop/ca2$ geth attach http://127.0.0.1:
8548
   N [05-26|12:33:36.381] Enabling deprecated personal namespace
Welcome to the Geth JavaScript console!
instance: Geth/node03/v1.11.6-stable-ea9e62ca/linux-amd64/go1.20.3
at block: 0 (Thu Jan 01 1970 03:30:00 GMT+0330 (+0330))
datadir: /home/mralavi/Desktop/ca2/node03
modules: admin:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 web3:1.0
To exit, press ctrl-d or type exit
> admin.addPeer("enode://55e1220713340e1dd8b24c5eab534b0b03934b0b3e76609f2e728
7cac62b73c2fb0327515e002b5971a41972dde6aaa2f3785112d04ed9229a0f39b4bd24b678@12
7.0.0.1:30298?discport=0")
true
```



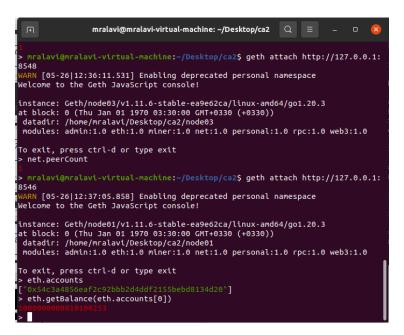


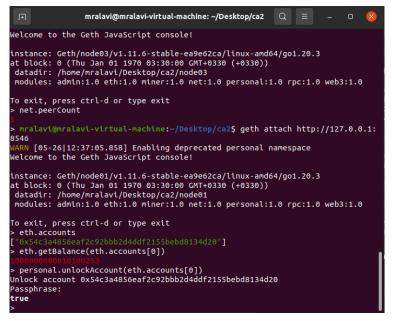
```
To exit, press ctrl-d or type exit
> net.peerCount
> mralavi@mralavi-virtual-machine:-/Desktop/ca2$ geth attach http://127.0.0.1:
8547
MARN [05-26|12:35:37.890] Enabling deprecated personal namespace
Welcome to the Geth JavaScript console!
instance: Geth/node02/v1.11.6-stable-ea9e62ca/linux-amd64/go1.20.3
at block: 0 (Thu Jan 01 1970 03:30:00 GMT+0330 (+0330))
datadir: /home/mralavi/Desktop/ca2/node02
modules: admin:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 web3:1.0

To exit, press ctrl-d or type exit
> net.peerCount

> mralavi@mralavi-virtual-machine:-/Desktop/ca2$ geth attach http://127.0.0.1:
8548
MARN [05-26|12:36:11.531] Enabling deprecated personal namespace
Welcome to the Geth JavaScript console!
instance: Geth/node03/v1.11.6-stable-ea9e62ca/linux-amd64/go1.20.3
at block: 0 (Thu Jan 01 1970 03:30:00 GMT+0330 (+0330))
datadir: /home/mralavi/Desktop/ca2/node03
modules: admin:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 web3:1.0

To exit, press ctrl-d or type exit
> net.peerCount
```





```
mralavi@mralavi-virtual-machine: -/Desktop/ca2 Q = - D &

C668DE5B5dd", value:1000})

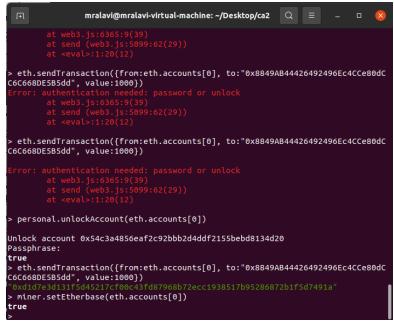
Error: authentication needed: password or unlock
    at web3.js:6365:9(39)
    at send (web3.js:5099:62(29))
    at <eval>:1:20(12)

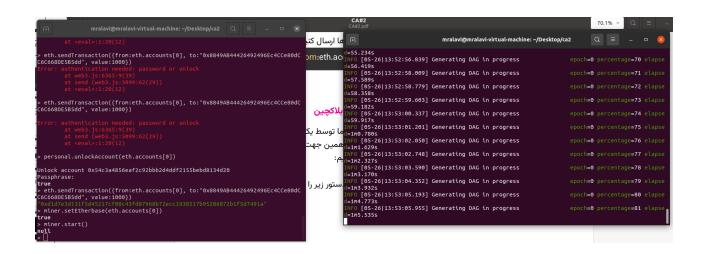
> eth.sendTransaction({from:eth.accounts[0], to:"0x8849AB44426492496Ec4Cce80dC
C6C668DE5B5dd", value:1000})

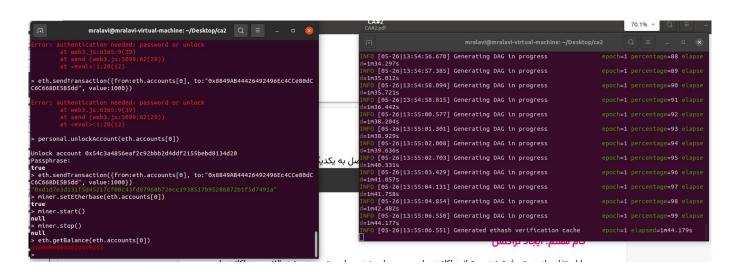
Error: authentication needed: password or unlock
    at web3.js:6365:9(39)
    at send (web3.js:5099:62(29))
    at <eval>:1:20(12)

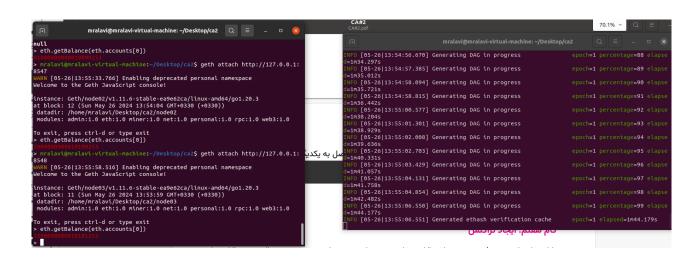
> eth.sendTransaction({from:eth.accounts[0], to:"0x8849AB44426492496Ec4Cce80dC
C6C668DE5B5dd", value:1000})

Error: authentication needed: password or unlock
    at web3.js:6365:9(39)
    at send (web3.js:5099:62(29))
    at
```









Question 1:

Every node works as a system and runs a client. They have the responsibility of storing and confirming the transactions. Nodes communicate with each other and store database of block chain.

Nodes in a cryptocurrency network play essential roles in maintaining the network's functionality, security, and decentralization.

Full Node:

Functions:

Store the Entire Blockchain: Full nodes keep a complete copy of the blockchain, which includes all transactions and blocks from the network's inception. This comprehensive storage allows full nodes to independently verify the entire transaction history.

Validate Transactions and Blocks: Full nodes check each transaction and block against the network's consensus rules. This includes verifying digital signatures, ensuring no double-spending occurs, and confirming the correctness and order of new blocks.

Propagate Transactions and Blocks: Full nodes disseminate validated transactions and blocks to other nodes in the network. This propagation helps maintain the network's decentralization and ensures that all nodes have up-to-date information.

Enforce Consensus Rules: Full nodes play a critical role in enforcing the network's rules by rejecting invalid transactions and blocks. This enforcement maintains the integrity and security of the blockchain.

Advantages:

High Security: By independently validating transactions and blocks, full nodes contribute significantly to the network's trustworthiness and security.

Network Stability: Full nodes help maintain the network's stability by ensuring all participants follow the same rules.

Disadvantages:

Resource Intensive: Running a full node requires significant storage space, computational power, and bandwidth, which can be costly and resource-demanding.

Light Node:

Functions:

Store Partial Blockchain Data: Light nodes store only a small portion of the blockchain, typically just the block headers, rather than the entire transaction history. This makes them much smaller and less resource-intensive.

Request Data from Full Nodes: Light nodes rely on full nodes to provide detailed transaction information. When a light node needs to verify a transaction, it requests the necessary data from full nodes.

Validate Transactions Using Simplified Payment Verification (SPV): Light nodes use SPV to verify transactions. They check the presence of a transaction in a block by verifying the block header and ensuring it is part of the longest chain, which is maintained by the full nodes.

Advantages:

Lower Resource Requirements: Light nodes require significantly less storage space, computational power, and bandwidth compared to full nodes, making them more accessible for average users and devices with limited resources.

Ease of Use: They are easier to set up and maintain, making them a more convenient option for users who do not need to run a full node.

Disadvantages:

Dependence on Full Nodes: Light nodes must trust full nodes for transaction verification, which can introduce security risks if the full nodes provide incorrect information.

Limited Validation Capability: Light nodes cannot independently verify the entire blockchain, limiting their ability to enforce all network rules.

Question 2:

Successfully sealed new block: This means that a suitable nonce has been found, and the block has been discovered. Note that this version of geth is working with PoW (Proof of Work). Additionally, other information such as the block number, block hash, and its fee are also included.

block reached canonical chain: This means that the block has been added to the canonical chain. The canonical chain is the main and valid chain.

Question 3:

Since the protocol is PoS (Proof of Stake), the power of individuals in the network is proportional to their holdings. Therefore, if someone wants to mine locally, they must first have enough money to mine a significant number of blocks. If they possess this much money, they will have a very high level of power and can effectively carry out this attack and even more

effective attacks. Additionally, this would cause all miners to lose their credibility because the previous chain would become invalidated, and they would never approve such an event.