K. J. Somaiya College of Engineering, Mumbai-77

Department of Computer Engineering

Batch: A2 **Roll No.: 1911031**
Experiment No. 06

Title: Implementation of private Ethereum Blockchain environment using Geth.

Objective: To create a private Ethereum blockchain environment using Geth.

Expected Outcome of Experiment:

CO	Outcome
CO3	Apply the algorithm and techniques used in Blockchain.
CO4	Grasp the in-depth understanding of Blockchain, Smart Contracts & how it works.

Books/ Journals/ Websites referred:

1. **https://www.investopedia.com/terms/e/ethereum.asp**
2. **https://www.simplilearn.com/tutorials/blockchain-tutorial/what-is-ethereum**
3. **https://dev.to/jeffersonxavier/create-your-own-private-blockchain-using-ethereum-52o5**
4. **https://geth.ethereum.org/docs/interface/private-network**

Theory: -

What Is Ethereum and How Does It Work?

What Is Ethereum?

At its core, Ethereum is a decentralized global software platform powered by blockchain technology. It is most commonly known for its native cryptocurrency, ether (ETH).

Ethereum can be used by anyone to create any secured digital technology. It has a token designed to pay for work done supporting the blockchain, but participants can also use it to pay for tangible goods and services if accepted.

Ethereum is designed to be scalable, programmable, secure, and decentralized. It is the blockchain of choice for developers and enterprises creating technology based upon it to change how many industries operate and how we go about our daily lives.

It natively supports smart contracts, an essential tool behind decentralized applications.1 Many decentralized finance (DeFi) and other applications use smart contracts in conjunction with blockchain technology.

Learn more about Ethereum, its token ETH, and how they are an integral part of non-fungible tokens, decentralized finance, decentralized autonomous organizations, and the metaverse.

KEY TAKEAWAYS

- Ethereum is a blockchain-based platform best known for its cryptocurrency, ether (ETH).
- The blockchain technology that powers Ethereum enables secure digital ledgers to be publicly created and maintained.
- Bitcoin and Ethereum have many similarities but different long-term visions and limitations.
- Ethereum changed from proof of work to proof of stake in Septemeber 2022.2
- Ethereum is the foundation for many emerging technological advances based on blockchain.

Ethereum Founder Joe Lubin Explains What It Is & Why It Matters

How Does Ethereum Work?

Vitalik Buterin, credited with conceiving Ethereum, published a white paper to introduce it in 2014.3 The Ethereum platform was launched in 2015 by Buterin and Joe Lubin, founder of the blockchain software company ConsenSys.45

The founders of Ethereum were among the first to consider the full potential of blockchain technology beyond just enabling the secure virtual payment method.

Since the launch of Ethereum, ether as a cryptocurrency has risen to become the second-largest cryptocurrency by market value. It is outranked only by Bitcoin.6

Blockchain Technology

Ethereum, like other cryptocurrencies, involves blockchain technology. Imagine a very long chain of blocks. All of the information contained in each block is added to every newly-created block with new data. Throughout the network, an identical copy of the blockchain is distributed.

This blockchain is validated by a network of automated programs that reach a consensus on the validity of transaction information. No changes can be made to the blockchain unless the network reaches a consensus. This makes it very secure.

Consensus is reached using an algorithm commonly called a consensus mechanism. Ethereum uses the proof-of-stake algorithm, where a network of participants called validators create new blocks and work together to verify the information they contain. The blocks contain information about the state of the blockchain, a list of attestations (a validator's signature and vote on the validity of the block), transactions, and much more.

In mid-September 2022, Ethereum officially switched over to a proof-of-stake algorithm, which is cheaper and more environmentally friendly than a proof-of-work model.2

Proof-of-Stake Mechanism

Proof-of-stake differs from proof-of-work in that it doesn't require the energy-intensive computing referred to as mining to validate blocks. It uses a finalization protocol called Casper-FFG and the algorithm LMD Ghost, combined into a consensus mechanism called Gasper, which monitors consensus and defines how validators receive rewards for work or are punished for dishonesty.7

Solo validators must stake 32 ETH to activate their validation ability. Individuals can stake smaller amounts of ETH, but they are required to join a validation pool and share any rewards. A validator creates a new block and attests that the information is valid in a process called attestation, where the block is broadcast to other validators called a committee who verify it and vote for its validity.

Validators who act dishonestly are punished under proof-of-stake. Validators who attempt to attack the network are identified by Gasper, which identifies the blocks to accept and reject based on the votes of the validators.7

Dishonest validators are punished by having their staked ETH burned and being removed from the network. Burning refers to sending crypto to a wallet that has no keys, which takes them out of circulation.

Wallets

Ethereum owners use wallets to store their ether. A wallet is a digital interface that lets you access your ether stored on the blockchain. Your wallet has an address, which is similar to an email address in that it is where users send ether, much like they would an email.8

Ether is not actually stored in your wallet. Your wallet holds private keys you use as you would a password when you initiate a transaction. You receive a private key for each ether you own. This key is essential for accessing your ether. That's why you hear so much about securing keys using different storage methods.

Historic Split

One notable event in Ethereum's history is the hard fork, or split, of Ethereum and Ethereum Classic.9 In 2016, a group of network participants gained majority control of the Ethereum blockchain to steal more than $50 million worth of ether, which had been raised for a project called The DAO.1011

The raid's success was attributed to the involvement of a third-party developer for the new project. Most of the Ethereum community opted to reverse the theft by invalidating the existing Ethereum blockchain and approving a blockchain with a revised history.

However, a fraction of the community chose to maintain the original version of the Ethereum blockchain. That unaltered version of Ethereum permanently split to become the cryptocurrency Ethereum Classic (ETC).12

Ethereum vs. Bitcoin

Ethereum is often compared to Bitcoin. While the two cryptocurrencies have many similarities, there are some some important distinctions.

Ethereum is described by founders and developers as “the world’s programmable blockchain,” positioning itself as an electronic, programmable network with many applications.13 The Bitcoin blockchain, by contrast, was created only to support the bitcoin cryptocurrency.

The Ethereum platform was founded with broad ambitions to leverage blockchain technology for many diverse applications. Bitcoin was designed strictly as a payment method.
The maximum number of bitcoins that can enter circulation is 21 million.14 The amount of ETH that can be created is unlimited, although the time it takes to process a block of ETH limits how much ether can be minted each year.15 The number of Ethereum coins in circulation is more than 122 million.16

Another significant difference between Ethereum and Bitcoin is how the respective networks treat transaction processing fees. These fees, known as gas on the Ethereum network, are paid by the participants in Ethereum transactions. The fees associated with Bitcoin transactions are absorbed by the broader Bitcoin network.

Ethereum, as of September 2022, uses a proof-of-stake consensus mechanism. Bitcoin uses the energy-intensive proof-of-work consensus, which requires miners to compete for rewards.

The Future of Ethereum

Ethereum’s transition to the proof-of-stake protocol, which enables users to validate transactions and mint new ETH based on their ether holdings, is part of a significant upgrade to the Ethereum platform. Previously called Eth2, this upgrade is now referred to only as Ethereum. However, Ethereum now has two layers. The first layer is the execution layer, where transactions and validations occur. The second layer is the consensus layer, where attestations and the consensus chain is maintained.17

The upgrade added capacity to the Ethereum network to support its growth, which will eventually help to address chronic network congestion problems that have driven up gas fees.218

To address scalability, Ethereum is continuing development of "sharding." Sharding will divide the Ethereum database amongst its network. This idea is similar to cloud computing, where many computers handle the workload to reduce computational time. These smaller database sections will be called shards, and shards will be worked on by those who have staked ETH. Shards will allow more validators to work at the same time, reducing the amount of time needed to reach consensus through a process called sharding consensus.19

Sharding is expected to be implemented sometime in 2023.

Use in Gaming

Ethereum is also being implemented into gaming and virtual reality. Decentraland is a virtual world that uses the Ethereum blockchain to secure items contained within that world. Land, avatars, wearables, buildings, and environments are all tokenized through the blockchain to create ownership.21

Axie Infinity is another game that uses blockchain technology and has its own cryptocurrency called Smooth Love Potion (SLP), used for rewards and transactions within the game.2223

Non-Fungible Tokens

Non-fungible tokens (NFTs) gained popularity in 2021. NFTs are tokenized digital items created using Ethereum.24 Generally speaking, tokenization gives one digital asset a specific digital token that identifies it and stores it on the blockchain.

This establishes ownership because the encrypted data stores the owner's wallet address. The NFT can be traded or sold and is viewed as a transaction on the blockchain. The transaction is verified by the network and ownership is transferred.

NFTs are being developed for all sorts of assets. For example, sports fans can buy a sports token—also called fan tokens—of their favorite athletes, which can be treated like trading cards. Some of these NFTs are pictures that resemble a trading card, and some of them are videos of a memorable or historic moment in the athlete's career.

The applications you may use in the metaverse, such as your wallet, a dApp, or the virtual world and buildings you visit, are likely to have been built on Ethereum.

The Development of DAOs

Decentralized Autonomous Organizations (DAOs), which are a collaborative method for making decisions across a distributed network, are being developed.25

For example, imagine that you created a venture capital fund and raised money through fund-raising, but you want decision-making to be decentralized and distributions to be automatic and transparent.

A DAO could use smart contracts and applications to gather the votes from the fund members and buy into ventures based on the majority of the group's votes, then automatically distribute any returns. The transactions could be viewed by all parties, and there would be no third-party involvement in handling any funds.

The part that cryptocurrency will play in the future is still vague. However, Ethereum appears to have a significant, upcoming role in personal and corporate finance and many aspects of our modern lives.

Ethereum Features

- Ether: This is Ethereum's cryptocurrency.
- Smart contracts: Ethereum allows the development and deployment of these types of contracts.
- Ethereum Virtual Machine: Ethereum provides the underlying technology—the architecture and the software—that understands smart contracts and allows you to interact with it.
- Decentralized applications (Dapps): A decentralized application is called a Dapp (also spelled DAPP, App, or DApp) for short. Ethereum allows you to create consolidated applications, called decentralized applications.
- Decentralized autonomous organizations (DAOs): Ethereum allows you to create these for democratic decision-making.

These are Ethereum's essential features. Before going deep into the Ethereum tutorial, let's discuss each of these features in more detail.

1. Ether

Ether (ETH) is Ethereum's cryptocurrency. It is the fuel that runs the network. It is used to pay for the computational resources and the transaction fees for any transaction executed on the Ethereum network. Like Bitcoins, ether is a peer-to-peer currency. Apart from being used to pay for transactions, ether is also used to buy gas, which is used to pay for the computation of any transaction made on the Ethereum network.

Also, if you want to deploy a contract on Ethereum, you will need gas, and you would have to pay for that gas in ether. So gas is the execution fee paid by a user for running a transaction in Ethereum. Ether can be utilized for building decentralized applications, building smart contracts, and making regular peer-to-peer payments.

2. Smart Contracts

Smart contracts are revolutionizing how traditional contracts work, which is why you need to use the tutorial to become more familiar with them. A smart contract is a simple

computer program that facilitates the exchange of any asset between two parties. It could be money, shares, property, or any other digital asset that you want to exchange. Anyone on the Ethereum network can create these contracts. The contract consists primarily of the terms and conditions mutually agreed on between the parties (peers).

The smart contract's primary feature is that once it is executed, it cannot be altered, and any transaction done on top of a smart contract is registered permanently—it is immutable. So even if you modify the smart contract in the future, the transactions correlated with the original contract will not get altered; you cannot edit them.

The verification process for the smart contracts is carried out by anonymous parties in the network without the need for a centralized authority, and that's what makes any smart contract execution on Ethereum a decentralized execution.

The transfer of any asset or currency is done in a transparent and trustworthy manner, and the identities of the two entities are secure on the Ethereum network. Once the transaction is successfully done, the accounts of the sender and receiver are updated accordingly, and in this way, it generates trust between the parties.

Smart Contracts Vs. Traditional Contract Systems

In conventional contract systems, you sign an agreement, then you trust and hire a third party for its execution. The problem is that in this type of process, data tampering is possible. With smart contracts, the agreement is coded in a program.

A centralized authority does not verify the result; it is confirmed by the participants on the Ethereum blockchain-based network. Once a contract is executed, the transaction is registered and cannot be altered or tampered, so it removes the risk of any data manipulation or alteration.

Let's take an example in which someone named Zack has given a contract of $500 to someone named Elsa for developing his company's website. The developers code the agreement of the smart contract using Ethereum's programming language.

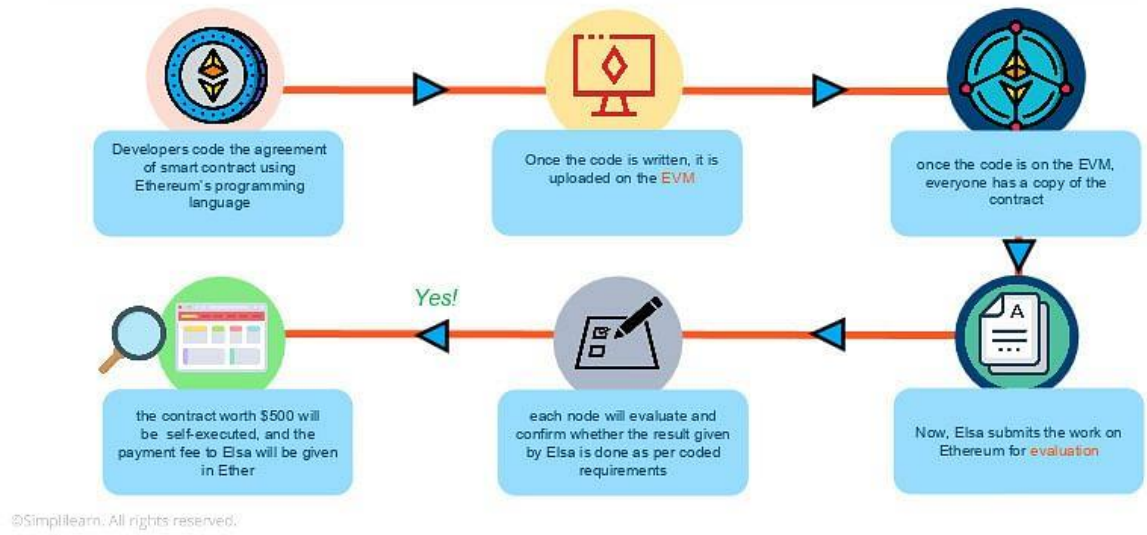
The smart contract has all the conditions (requirements) for building the website. Once the code is written, it is uploaded and deployed on the Ethereum Virtual Machine (EVM).

EVM is a runtime compiler to execute a smart contract. Once the code is deployed on the EVM, every participant on the network has a copy of the contract. When Elsa

submits the work on Ethereum for evaluation, each node on the Ethereum network will evaluate and confirm whether the result given by Elsa has been done as per the coding requirements.

Once the result is approved and verified, the contract worth $500 will be self-executed, and the payment will be paid to Elsa in ether. Zack's account will be automatically debited, and Elsa will be credited with $500 in ether.

The Ethereum tutorial video includes a demo on the deployment of an Ethereum smart contract.



3. Ethereum Virtual Machine

EVM, as mentioned above in this Ethereum tutorial, is designed to operate as a runtime environment for compiling and deploying Ethereum-based smart contracts. EVM is the engine that understands the language of smart contracts, which are written in the Solidity language for Ethereum. EVM is operated in a sandbox environment—basically, you can deploy your stand-alone environment, which can act as a testing and development environment. You can then test your smart contract (use it) "n" number of times, verify it, and once you are satisfied with the performance and the functionality of the smart contract, you can deploy it on the Ethereum main network.

Any programming language in the smart contract is compiled into the bytecode, which the EVM understands. This bytecode can be read and executed using the EVM. Solidity is one of the most popular languages for writing a smart contract. Once you write your

smart contract in Solidity, that contract gets converted into the bytecode and gets deployed on the EVM, thereby guaranteeing security from cyberattacks.

a) How Does EVM Work?

Suppose person A wants to pay person B 10 ethers. The transaction will be sent to the EVM using a smart contract for a fund transfer from A to B. To validate the transaction; the Ethereum network will perform the proof-of-work consensus algorithm

The miner nodes on Ethereum will validate this transaction—whether the identity of A exists or not, and if A has the requested amount to transfer. Once the transaction is confirmed, the ether will be debited from A's wallet and will be credited to B's wallet, and during this process, the miners will charge a fee to validate this transaction and will earn a reward.

All the nodes on the Ethereum network execute smart contracts using their respective EVMs.

b) Proof of Work

Every node in the Ethereum network has:

- The entire history of all the transactions—the entire chain
- The history of the smart contract, which is the address at which the smart contract is deployed, along with the transactions associated with the smart contract
- The handle to the current state of the smart contract

The goal of the miners on the Ethereum network is to validate the blocks. For each block of a transaction, miners use their computational power and resources to get the appropriate hash value by varying the nonce. The miners will vary the nonce and pass it through a hashing algorithm—in Ethereum, it is the Ethash algorithm.

This produces a hash value that should be less than the predefined target as per the proof-of-work consensus. If the hash value generated is less than the target value, then the block is considered to be verified, and the miner gets rewarded.

When the proof of work is solved, the result is broadcast and shared with all the other nodes to update their ledger. If other nodes accept the hashed block as valid, then the block gets added to the Ethereum main blockchain, and as a result, the miner receives a

reward, which as of today stands at three ethers. Plus, the miner gets the transaction fees that have been generated for verifying the block. All the transactions that are aggregated in the block—the cumulative transaction fees associated with all the transactions are also rewarded to the miner.

c) Proof of Stake

In Ethereum, a process called proof of stake is also under development. It is an alternative to proof of work and is meant to be a solution to minimize the use of expensive resources spent on mining using proof of work. In proof of stake, the miner—who is the validator—can validate the transactions based on the number of crypto coins he or she holds before actually starting the mining.

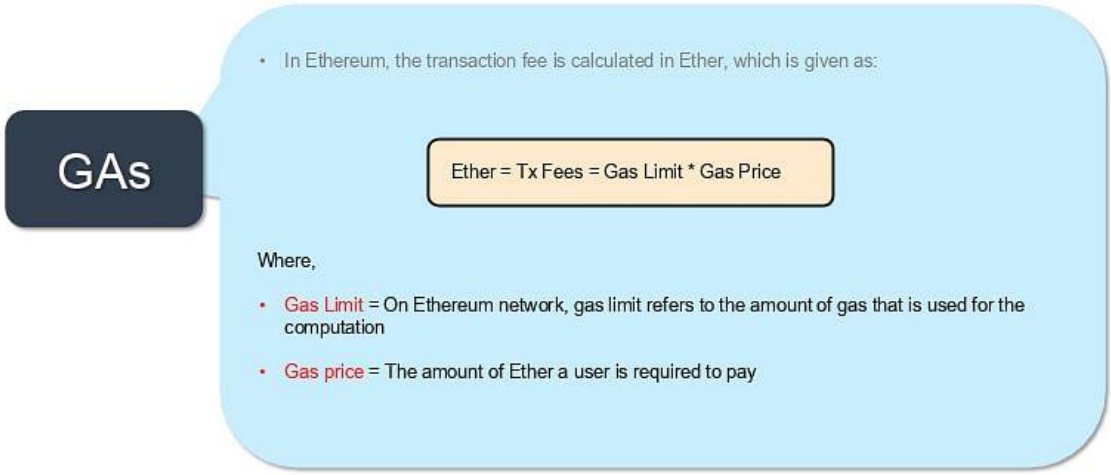
So, based on the accumulation of crypto coins the miner has beforehand, he or she has a higher probability of mining the block. However, proof of stake is not widely used as of now compared to proof of work.

d) Gas

Just like we need fuel to run a car, we need gas to run applications on the Ethereum network. To perform any transaction within the Ethereum network, a user must make a payment, in this case paying out ethers, to get a transaction done, and the intermediary monetary value is called gas.

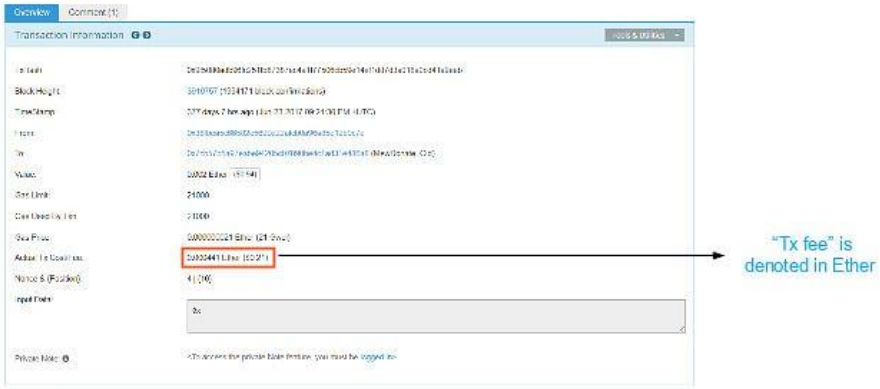
On the Ethereum network, gas is a unit that measures the computational power required to run a smart contract or a transaction. So, if you must do a transaction that updates the blockchain, you would have to shell out gas, and that gas costs ethers.

In Ethereum, the transaction fees are calculated using a formula (see screenshot below). For every transaction, there is gas and its correlated gas price. The transaction fees equal the amount of gas required to execute a transaction multiplied by the gas price. “Gas limit” refers to the amount of gas used for the computation and the amount of ether a user is required to pay for the gas.



Below is a screenshot from the Ethereum network showing the transaction cost. You can see for this particular transaction, the gas limit was 21,000, the gas used by the transaction was 21,000, and the gas price was 21 Gwei, which is the lowest denomination of ether. So, 21 Gwei * 21,000 gave the actual transaction fees: 0.000441 ethers, or about 21 cents as of today. As mentioned, the transaction fee goes to the miner, who has validated the transaction.

For example, below is a screenshot of Ether transaction, where the cost of transaction fee is shown



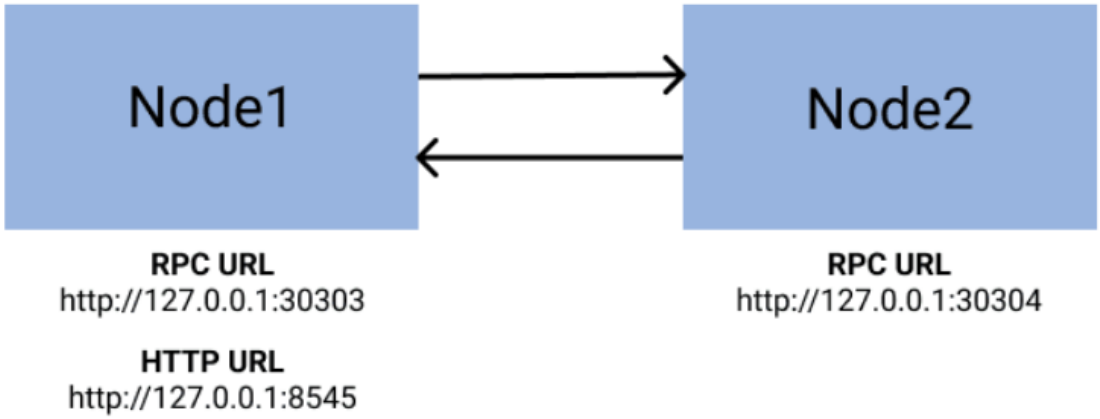
Note: "Tx Fee" is paid by users to miners

To understand the gas limit and price, let's consider an example using a car. Suppose your vehicle has a mileage of 10 kilometers per liter and petrol costs $1 per liter. Under these parameters, driving a car for 50 kilometers would cost you five liters of petrol, which is worth $5. Similarly, to perform an operation or to run code on Ethereum, you need to obtain a certain amount of gas, like petrol, and the gas has a per-unit price, called gas price.

Implementation Details:

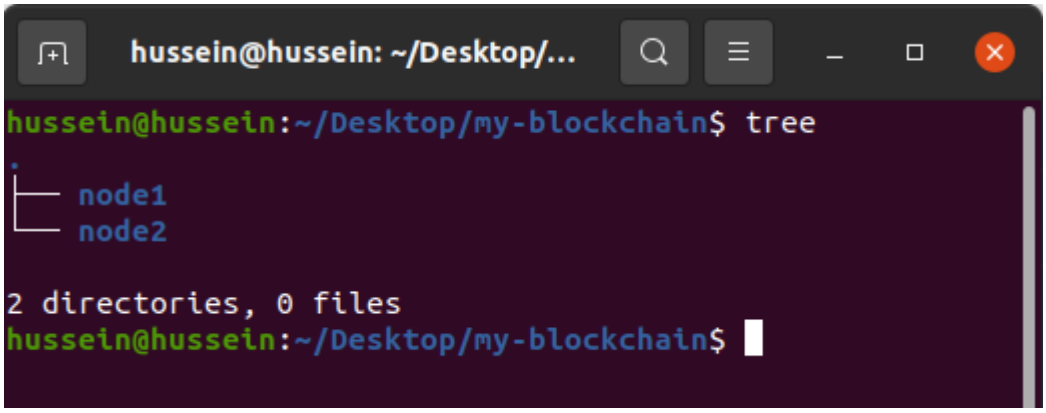
1. Enlist all the Steps followed and various options explored

- In this experiment we will configure a network something like this.



- Downloaded geth from go ethereum official website.

- Creating the required folders for the two nodes in our blockchain



- Now creating account for the two nodes that is node 1 and node 2, both of which reciever an initial ether balance.

```
hussein@hussein: ~/Desktop/my-blockchain
hussein@hussein:~/Desktop/my-blockchain$ geth --datadir node1 account new
INFO [12-01|21:36:49.410] Maximum peer count                       ETH=50 LES=0 total=50
INFO [12-01|21:36:49.410] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or direct
ory"
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:   0x61fad26D7F94d2d97Fa9e46D828A7a2C87f655b5
Path of the secret key file: node1/keystore/UTC--2022-12-01T16-06-54.665284891Z--61fad26d7f94d2d97fa9e46d828a7a2c87f655b5

- 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!

hussein@hussein:~/Desktop/my-blockchain$
```

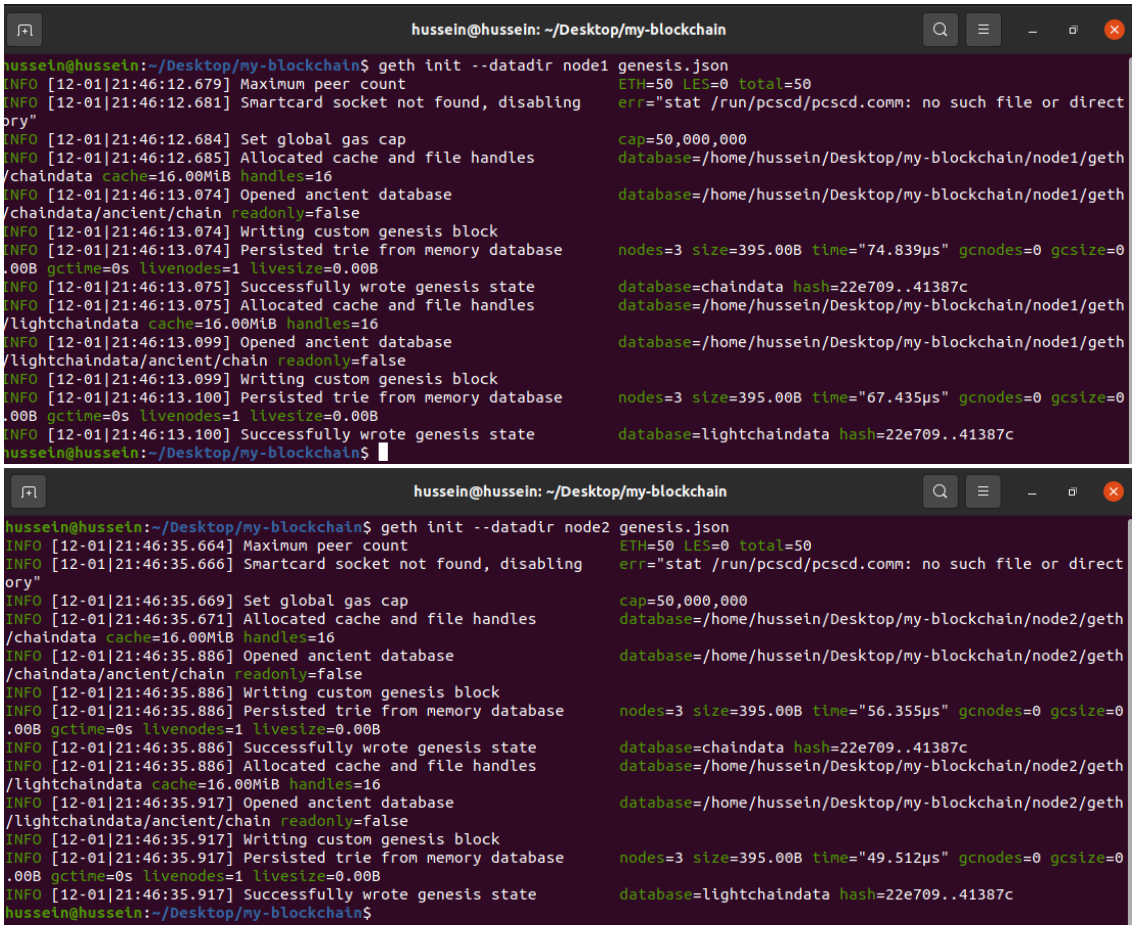
```
hussein@hussein: ~/Desktop/my-blockchain
hussein@hussein:~/Desktop/my-blockchain$ geth --datadir node2 account new
INFO [12-01|21:39:52.954] Maximum peer count                       ETH=50 LES=0 total=50
INFO [12-01|21:39:52.954] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or direct
ory"
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:   0x1eFB630d71826b718b2CBb91C062C8d79b00Fb9a
Path of the secret key file: node2/keystore/UTC--2022-12-01T16-09-57.313033490Z--1efb630d71826b718b2cbb91c062c8d79b00fb9a

- 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!

hussein@hussein:~/Desktop/my-blockchain$
```

- Now create a genesis.json file with the required configurations that are needed for the ethereum network to be created

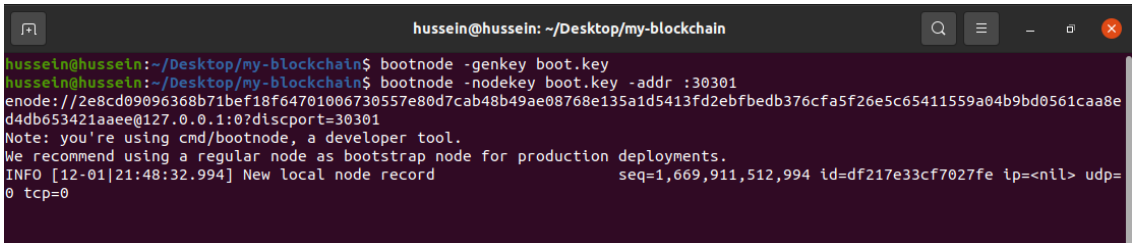
*genesis.json

```
{
  "config": {
    "chainId": 12345,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "istanbulBlock": 0,
    "berlinBlock": 0,
    "clique": {
      "period": 5,
      "epoch": 30000
    }
  },
  "difficulty": "1",
  "gasLimit": "8000000",
  "extradata":
"0x0000000000000000000000000000000000000000000000000000000000000000007df9a875a174b3bc565e6424a00
  "alloc": {
    "0x61fad26D7F94d2d97Fa9e46D828A7a2C87f655b5": { "balance": "1000" },
    "0x1eFB630d71826b718b2CBb91C062C8d79b00Fb9a": { "balance": "2000" }
  }
}
```

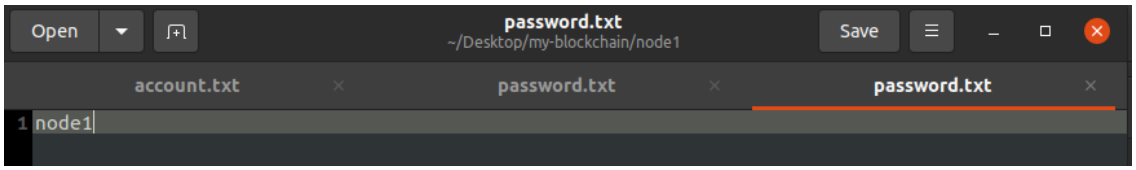
- Now link the two nodes with the configurations mentioned in the genesis.json file

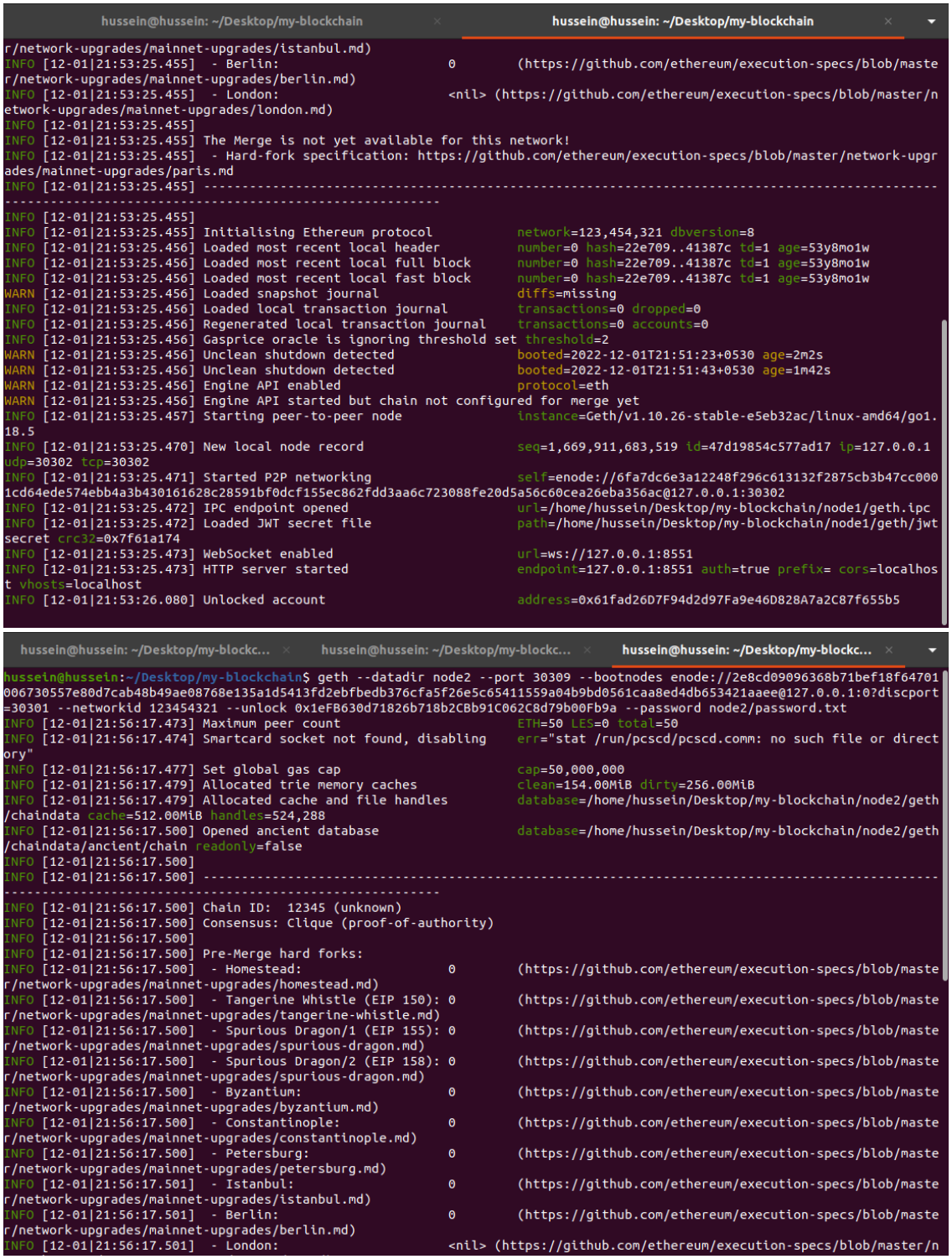


- Now create bootnode and start it on the port number 30301 and node the enode address for future use.



- Now start the first node on port number 8557 and node 2 on port number 8558





- Now the two nodes are connected to the bootnode. Now open a javascript termial by attaching the geth command to the port number 8557

```
hussein@hussein:~/Desktop/my-blockchain$ geth attach node1/geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.5
coinbase: 0x61fad26d7f94d2d97fa9e46d828a7a2c87f655b5
at block: 0 (Thu Jan 01 1970 05:30:00 GMT+0530 (IST))
 datadir: /home/hussein/Desktop/my-blockchain/node1
 modules: admin:1.0 clique:1.0 debug:1.0 engine: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 or type exit
> admin.nodeInfo
{
  enode: "enode://6fa7dc6e3a12248f296c613132f2875cb3b47cc0001cd64ede574ebb4a3b430161628c28591bf0dcf155ec862fdd3aa6c723088f
e20d5a56c60cea26eba356ac@127.0.0.1:30302",
  enr: "enr:-KO4QOSLmhTr81CJQTTBMB4dLSSe4qnXw5J2HauBKObbHrZVDXcQbiI8Efgo8D1GZJwdz6rU_qLEXj0ugISQLStloK6GAYTOfiHBg2V0aMfGhK
K2nMiAgmlkgnY0gmlwhH8AAAGJc2VjcDI1NmsxoQJvp9xuOhIkjylsYTEy8odcs7R8wAAc1k7eV067SjtDAYRzbmFwwIN0Y3CCdl6DdWRwgnZe",
  id: "47d19854c577ad17766645f3d8ce46fadb9f42b53005da07acde3feb3018a52b",
  ip: "127.0.0.1",
  listenAddr: "[::]:30302",
  name: "Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.5",
  ports: {
    discovery: 30302,
    listener: 30302
  },
  protocols: {
    eth: {
      config: {
        berlinBlock: 0,
        byzantiumBlock: 0,
        chainId: 12345,
        clique: {...},
        constantinopleBlock: 0,
        eip150Block: 0,
        eip150Hash: "0x0000000000000000000000000000000000000000000000000000000000000000",
        eip155Block: 0,
```

```
{
  enode: "enode://6fa7dc6e3a12248f296c613132f2875cb3b47cc0001cd64ede574ebb4a3b430161628c28591bf0dcf155ec862fdd3aa6c723088f
e20d5a56c60cea26eba356ac@127.0.0.1:30302",
  enr: "enr:-KO4QOSLmhTr81CJQTTBMB4dLSSe4qnXw5J2HauBKObbHrZVDXcQbiI8Efgo8D1GZJwdz6rU_qLEXj0ugISQLStloK6GAYTOfiHBg2V0aMfGhK
K2nMiAgmlkgnY0gmlwhH8AAAGJc2VjcDI1NmsxoQJvp9xuOhIkjylsYTEy8odcs7R8wAAc1k7eV067SjtDAYRzbmFwwIN0Y3CCdl6DdWRwgnZe",
  id: "47d19854c577ad17766645f3d8ce46fadb9f42b53005da07acde3feb3018a52b",
  ip: "127.0.0.1",
  listenAddr: "[::]:30302",
  name: "Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.5",
  ports: {
    discovery: 30302,
    listener: 30302
  },
  protocols: {
    eth: {
      config: {
        berlinBlock: 0,
        byzantiumBlock: 0,
        chainId: 12345,
        clique: {...},
        constantinopleBlock: 0,
        eip150Block: 0,
        eip150Hash: "0x0000000000000000000000000000000000000000000000000000000000000000",
        eip155Block: 0,
        eip158Block: 0,
        homesteadBlock: 0,
        istanbulBlock: 0,
        petersburgBlock: 0
      },
      difficulty: 1,
      genesis: "0x22e709687ccd09e4fbf58a3aee08bb03df92c57bf83228fcd1230989e541387c",
      head: "0x22e709687ccd09e4fbf58a3aee08bb03df92c57bf83228fcd1230989e541387c",
      network: 123454321
    },
    snap: {}
  }
}
>
```

```
hussein@hussein:~/Desktop/my-blockchain$ geth attach node2/geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.5
coinbase: 0x1efb630d71826b718b2cbb91c062c8d79b00fb9a
at block: 0 (Thu Jan 01 1970 05:30:00 GMT+0530 (IST))
 datadir: /home/hussein/Desktop/my-blockchain/node2
 modules: admin:1.0 clique:1.0 debug:1.0 engine: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 or type exit
> admin.nodeInfo
{
  enode: "enode://71da534a2dfbf5421021f4338040f870ffff5231b839b21762aa8169240d734c782fcfe85bfcd45105cffde2025d8ed5984ef95374f2dfee63ade1a9bca27822@127.0.0.1:9500",
  enr: "enr:-KO4QJk89Snk4IksJCBdeNM-C_cN_aBBp58UtLz5rZptzVvKdOhecbIVE4XWny2kfP34Yyx2eGqXZrdPO1ylk7Qlg8WGAYTOgURpg2V0aMfGhKK2nMiAgmlkgnY0gmlwhH8AAAGJc2VjcDI1NmsxoQJx2lNKLfv1QhAh9DOAQPhw__9SMbg5shdlqoFpJA1zTIRzbmFwwIN0Y3CCJRyDdWRwgiUc",
  id: "7f47a0fd9237ba34f650dfbd0282f17d897391c9d5f32c1a87372ea4adcba0d7",
  ip: "127.0.0.1",
  listenAddr: "[::]:9500",
  name: "Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.5",
  ports: {
    discovery: 9500,
    listener: 9500
  },
  protocols: {
    eth: {
      config: {
        berlinBlock: 0,
        byzantiumBlock: 0,
        chainId: 12345,
        clique: {...},
        constantinopleBlock: 0,
        eip150Block: 0,
        eip150Hash: "0x0000000000000000000000000000000000000000000000000000000000000000",
        eip155Block: 0,
        eip158Block: 0,
        homesteadBlock: 0,
        istanbulBlock: 0,
```

- Now unlocking account associated to node 1

```
hussein@hussein: ...  hussein@hussein: ...  hussein@hussein: ...  hussein@hussein: ...  hussein@hussein: ...
INFO [12-01|22:12:35.690] ------------------------------------------------------------------------------------------------------------------------
INFO [12-01|22:12:35.690]
INFO [12-01|22:12:35.691] Initialising Ethereum protocol           network=123,454,321 dbversion=8
INFO [12-01|22:12:35.693] Loaded most recent local header          number=0 hash=22e709..41387c td=1 age=53y8mo1w
INFO [12-01|22:12:35.693] Loaded most recent local full block      number=0 hash=22e709..41387c td=1 age=53y8mo1w
INFO [12-01|22:12:35.693] Loaded most recent local fast block      number=0 hash=22e709..41387c td=1 age=53y8mo1w
INFO [12-01|22:12:35.694] Loaded local transaction journal         transactions=0 dropped=0
INFO [12-01|22:12:35.694] Regenerated local transaction journal    transactions=0 accounts=0
INFO [12-01|22:12:35.695] Gasprice oracle is ignoring threshold set threshold=2
WARN [12-01|22:12:35.695] Unclean shutdown detected                booted=2022-12-01T21:51:23+0530 age=21m12s
WARN [12-01|22:12:35.695] Unclean shutdown detected                booted=2022-12-01T21:51:43+0530 age=20m52s
WARN [12-01|22:12:35.696] Engine API enabled                       protocol=eth
WARN [12-01|22:12:35.696] Engine API started but chain not configured for merge yet
INFO [12-01|22:12:35.697] Starting peer-to-peer node               instance=Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.
18.5
INFO [12-01|22:12:35.728] New local node record                    seq=1,669,911,683,523 id=47d19854c577ad17 ip=127.0.0.1
udp=30302 tcp=30302
INFO [12-01|22:12:35.730] Started P2P networking                   self=enode://6fa7dc6e3a12248f296c613132f2875cb3b47cc000
1cd64ede574ebb4a3b430161628c28591bf0dcf155ec862fdd3aa6c723088fe20d5a56c60cea26eba356ac@127.0.0.1:30302
INFO [12-01|22:12:35.736] IPC endpoint opened                      url=/home/hussein/Desktop/my-blockchain/node1/geth.ipc
INFO [12-01|22:12:35.736] Loaded JWT secret file                   path=/home/hussein/Desktop/my-blockchain/node1/geth/jwt
secret crc32=0x7f61a174
INFO [12-01|22:12:35.738] WebSocket enabled                        url=ws://127.0.0.1:8551
INFO [12-01|22:12:35.738] HTTP server started                      endpoint=127.0.0.1:8551 auth=true prefix= cors=localhos
t vhosts=localhost
INFO [12-01|22:12:36.795] Unlocked account                         address=0x61fad26D7F94d2d97Fa9e46D828A7a2C87f655b5
INFO [12-01|22:12:45.812] Looking for peers                        peercount=0 tried=0 static=0
INFO [12-01|22:12:55.930] Looking for peers                        peercount=0 tried=0 static=0
INFO [12-01|22:13:05.966] Looking for peers                        peercount=0 tried=0 static=0
INFO [12-01|22:13:16.014] Looking for peers                        peercount=0 tried=0 static=0
INFO [12-01|22:13:26.112] Looking for peers                        peercount=0 tried=0 static=0
INFO [12-01|22:13:36.418] Looking for peers                        peercount=0 tried=0 static=0
INFO [12-01|22:13:46.457] Looking for peers                        peercount=0 tried=0 static=0
INFO [12-01|22:13:56.550] Looking for peers                        peercount=1 tried=1 static=0
INFO [12-01|22:14:06.595] Looking for peers                        peercount=1 tried=0 static=0
INFO [12-01|22:14:16.676] Looking for peers                        peercount=1 tried=0 static=0
```

```
hussein@hussein: ...  hussein@hussein: ...  hussein@hussein: ...  hussein@hussein: ...  hussein@hussein: ...

INFO [12-01|22:13:54.054]  - Hard-fork specification: https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md
INFO [12-01|22:13:54.054] ------------------------------------------------------------------------------------------------------------------------
INFO [12-01|22:13:54.054]
INFO [12-01|22:13:54.054] Initialising Ethereum protocol           network=123,454,321 dbversion=8
INFO [12-01|22:13:54.056] Loaded most recent local header          number=0 hash=22e709..41387c td=1 age=53y8mo1w
INFO [12-01|22:13:54.056] Loaded most recent local full block      number=0 hash=22e709..41387c td=1 age=53y8mo1w
INFO [12-01|22:13:54.056] Loaded most recent local fast block      number=0 hash=22e709..41387c td=1 age=53y8mo1w
INFO [12-01|22:13:54.056] Loaded local transaction journal         transactions=0 dropped=0
INFO [12-01|22:13:54.056] Regenerated local transaction journal    transactions=0 accounts=0
INFO [12-01|22:13:54.057] Gasprice oracle is ignoring threshold set threshold=2
WARN [12-01|22:13:54.057] Unclean shutdown detected                booted=2022-12-01T21:54:48+0530 age=19m6s
WARN [12-01|22:13:54.057] Unclean shutdown detected                booted=2022-12-01T21:55:50+0530 age=18m4s
WARN [12-01|22:13:54.057] Unclean shutdown detected                booted=2022-12-01T21:55:59+0530 age=17m55s
WARN [12-01|22:13:54.057] Unclean shutdown detected                booted=2022-12-01T21:56:17+0530 age=17m37s
WARN [12-01|22:13:54.057] Unclean shutdown detected                booted=2022-12-01T22:04:02+0530 age=9m52s
WARN [12-01|22:13:54.057] Unclean shutdown detected                booted=2022-12-01T22:13:48+0530 age=6s
WARN [12-01|22:13:54.057] Engine API enabled                       protocol=eth
WARN [12-01|22:13:54.057] Engine API started but chain not configured for merge yet
INFO [12-01|22:13:54.057] Starting peer-to-peer node               instance=Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.5
INFO [12-01|22:13:54.074] New local node record                    seq=1,669,911,889,004 id=7f47a0fd9237ba34 ip=127.0.0.1 udp=30303 tcp=30303
INFO [12-01|22:13:54.075] Started P2P networking                   self=enode://71da534a2dfbf5421021f4338040f870ffff5231b839b21762aa8169240d734c782fcfe85bfcd45105cffde2025d8ed5984ef95374f2dfee63ade1a9bca27822@127.0.0.1:30303
INFO [12-01|22:13:54.076] IPC endpoint opened                      url=/home/hussein/Desktop/my-blockchain/node2/geth.ipc
INFO [12-01|22:13:54.076] Loaded JWT secret file                   path=/home/hussein/Desktop/my-blockchain/node2/geth/jwtsecret crc32=0x141950d2
INFO [12-01|22:13:54.077] WebSocket enabled                        url=ws://127.0.0.1:8552
INFO [12-01|22:13:54.077] HTTP server started                      endpoint=127.0.0.1:8552 auth=true prefix= cors=localhost vhosts=localhost
INFO [12-01|22:13:55.173] Unlocked account                         address=0x1eFB630d71826b718b2CBb91C062C8d79b00Fb9a
INFO [12-01|22:14:04.252] Looking for peers                        peercount=1 tried=1 static=0
INFO [12-01|22:14:14.392] Looking for peers                        peercount=1 tried=0 static=0
INFO [12-01|22:14:24.569] Looking for peers                        peercount=1 tried=0 static=0
INFO [12-01|22:14:34.629] Looking for peers                        peercount=1 tried=0 static=0
```

```
> admin.peers
[{
    caps: ["eth/66", "eth/67", "snap/1"],
    enode: "enode://71da534a2dfbf5421021f4338040f870ffff5231b839b21762aa8169240d734c782fcfe85bfcd45105cffde2025d8ed5984ef95374f2dfee63ade1a9bca27822@127.0.0.1:41430",
    id: "7f47a0fd9237ba34f650dfbd0282f17d897391c9d5f32c1a87372ea4adcba0d7",
    name: "Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.5",
    network: {
      inbound: true,
      localAddress: "127.0.0.1:30302",
      remoteAddress: "127.0.0.1:41430",
      static: false,
      trusted: false
    },
    protocols: {
      eth: {
        difficulty: 1,
        head: "0x22e709687ccd09e4fbf58a3aee08bb03df92c57bf83228fcd1230989e541387c",
        version: 67
      },
      snap: {
        version: 1
      }
    }
}]
```

```
> net.peerCount
1
> admin.peers
[{
    caps: ["eth/66", "eth/67", "snap/1"],
    enode: "enode://6fa7dc6e3a12248f296c613132f2875cb3b47cc0001cd64ede574ebb4a3b430161628c28591bf0dcf155ec862fdd3aa6c723088fe20d5a56c60cea26eba356ac@127.0.0.1:30302",
    id: "47d19854c577ad17766645f3d8ce46fadb9f42b53005da07acde3feb3018a52b",
    name: "Geth/v1.10.26-stable-e5eb32ac/linux-amd64/go1.18.5",
    network: {
      inbound: false,
      localAddress: "127.0.0.1:41430",
      remoteAddress: "127.0.0.1:30302",
      static: false,
      trusted: false
    },
    protocols: {
      eth: {
        difficulty: 1,
        head: "0x22e709687ccd09e4fbf58a3aee08bb03df92c57bf83228fcd1230989e541387c",
        version: 67
      },
      snap: {
        version: 1
      }
    }
}]
>
```

- Now that the nodes are connected lets execute a transaction and see whether the balance changes or not.

```
> eth.sendTransaction({to: "0x0e1420c876855bd69fdc27c212cdc4fc993702e7", from: eth.accounts[0], value: 286})
```


```
{
  "config": {
    "chainId": 12345,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "istanbulBlock": 0,
    "berlinBlock": 0,
    "clique": {
      "period": 5,
      "epoch": 30000
    }
  },
  "difficulty": "1",
  "gasLimit": "80000",
  "extradata":
"0x0000000000000000000000000000000000000000000000000000000000000000007df9a875a174b3bc565e6424a00
  "alloc": {
    "0x74e1bEDB9621Bed872390c9b7108f56a69d8c836": { "balance": "999714" },
    "0x0E1420c876855Bd69fdc27c212cdc4Fc993702E7": { "balance": "2000286" }
  }
}
```


2. Explain your program logic, classes and methods used.

Methods:

1) **addPeer():** For the experiment, two nodes were created, and a boot node was created for fetching the logs from the network. These nodes must be connected to each other, so this method is used for the same. This method takes the enode address of another node to which the node that is running it wants to connect. After the successful execution of this step, the nodes will be connected to each other.
2) **newAccount():** Geth is a tool that is based on Ethereum, and as we know, Ethereum uses an account-based architecture. In this experiment, accounts are created for each node created. This function is used to attach an account to nodes in the network. This also provides some ethers for accounts.
3) **getBalance():** This method is used to fetch the account balance in ethers. For this experiment, it was used for verification of account balances after the transaction.
4) **unlockAccount():** This method is used to unlock a particular account for the transaction. The account address is passed as the parameter to this method.
5) **sendTransaction():** This method is used for making the transaction. This method takes an object as a parameter, which defines the from and to addresses of the transaction and the amount to transfer.

3. Explain the Importance of the approach followed by you

1. **Data coordination.** Ethereum's decentralized architecture better allocates information and trust so that network participants do not have to rely on a central entity to manage the system and mediate transactions.
2. **Rapid deployment.** With an all-in-one SaaS platform like Hyperledger Besu, enterprises can easily deploy and manage private blockchain networks instead of coding a blockchain implementation from scratch.
3. **Permissioned networks.** The ConsenSys Quorum open source protocol layer enables businesses to build on public or private Ethereum networks, ensuring your solution fits any potential regulatory and security requirements.
4. **Network size.** The mainnet proves that an Ethereum network can work with hundreds of nodes and millions of users. Most enterprise blockchain competitors are only running networks of less than 10 nodes and have no reference case for a vast and viable network. Network size is critical for enterprise consortia that are bound to outgrow a handful of nodes.
5. **Private transactions**. Enterprises can achieve granularity of privacy in Ethereum by forming private consortia with private transaction layers. On ConsenSys Quorum, private information is never broadcast to network participants. Private data is encrypted and only shared directly with relevant parties.
6. **Scalability and performance.** With Proof of Authority consensus and custom block time and gas limit, consortium networks built on Ethereum can outperform the public mainnet and scale up to hundreds of transactions per second or more depending on network configuration. Protocol-level solutions like sharding and off-chain, layer 2 scaling solutions such as Plasma and statechannels present opportunities for Ethereum to increase its throughput in the near future.
7. **Finality.** A blockchain's consensus algorithm secures confidence that the record of transactions remains tamper-proof and canonical. Ethereum offers customizable consensus mechanisms including RAFT and IBFT for different enterprise network instances, ensuring immediate transaction finality and reducing the required infrastructure that the Proof of Work algorithm demands.
8. **Incentive layer.** Ethereum's cryptoeconomic layers allow business networks to develop mechanisms that both punish nefarious activity and create rewards around activities such as verification and availability.
9. **Tokenization.** Businesses can tokenize any asset on Ethereum that has been registered in a digital format. By tokenizing assets, organizations can fractionalize previously monolithic assets (real estate), expand their line of products (provably rare art), and unlock new incentive models (crowdsourced data management).

10. **Standards.** Ethereum is where the standards are. Protocols around token design (ERC20), human-readable names (ENS), decentralized storage (Swarm), and decentralized messaging (Whisper) keep the ecosystem from balkanizing. For enterprises, the Enterprise Ethereum Alliance's Client Specification 1.0 defines the architectural components for compliant enterprise blockchain implementations. The EEA is planning to release version 2.0 of the spec soon.
11. **Interoperability and open source.** Consortia on Ethereum are not locked into the IT environment of a single vendor. Amazon Web Services customers, for example, can operate private networks with Kaleido's Blockchain Business Cloud. Like the Java community's spec-driven philosophy, the Ethereum ecosystem welcomes contributions to the codebase through Ethereum Improvement Proposals (EIPs).

Conclusion: - Understood how to create a private blockchain on the local network using Ethereum and without the help of ganache. Then we also understood how to connect multiple peers so that they can perform transactions between themselves.