Private Multi node Block chain:

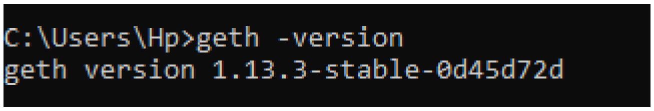
This documentation will guide steps to set up a private ethereum network with multiple node on different machine.

You can also find steps to do from https://geth.ethereum.org/docs/fundamentals/private-network.

Step 1: Install ethereum client on each machine.

We are using Geth (Go-ethereum) - You can download the software from https://geth.ethereum.org/downloads depending on your operating system.

Check whether latest version of Geth is installed:



```
C:\Users\Hp>geth -version
geth version 1.13.3-stable-0d45d72d
```

Step 2: Open command prompt terminal

Step 3: Create node folder in desired location

- mkdir node1,node2 in virtual machine(Ubuntu)

- mkdir node3,node4 in local machine

Step 4: Create account in each node folder:

https://ethereum.org/en/developers/docs/accounts/

node 1- geth --datadir "./data" account new

(ubuntu VM) ---->0x049F741544Bc885CDDd50BaFb01dfE78Fc4f23AC

Node2- geth --datadir "./data" account new

(ubuntu VM) ---->0x46Ba548AbC13366912e5D187D5Df82219eE7f132

Node 3 - geth --datadir "./data" account new

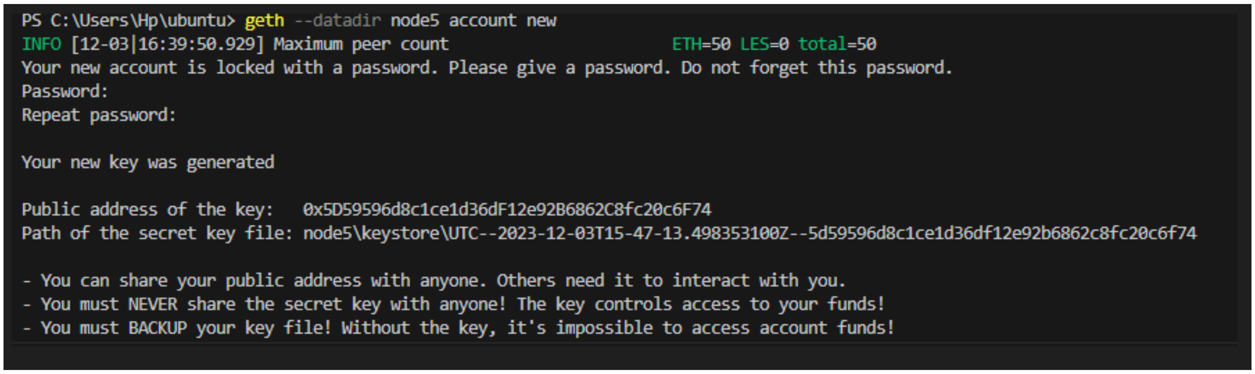
(local1) ---->0xB90bcFd7D6A04635b6e3375E1c3Dcf4d02bf798e

Node 4 - geth --datadir "./data" account new

(local2) ---->0x60f9C7346406b293a6a780663cd9A6D4e0583633

When you create account, it will ask you to enter password for the account. Please do and save the password in text file under each folder

This password will be used to unlock account when starting node later.



```
PS C:\Users\Hp\ubuntu> geth --datadir node5 account new
INFO [12-03|16:39:50.929] Maximum peer count                       ETH=50 LES=0 total=50
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:   0x5D59596d8c1ce1d36dF12e92B6862C8fc20c6F74
Path of the secret key file: node5\keystore\UTC--2023-12-03T15-47-13.498353100Z--5d59596d8c1ce1d36df12e92b6862c8fc20c6f74

- 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!
```

Step 5: Now we need to create genesis block in .json file

in ubuntu: nano genesis.json and paste your genesis block

```
{
`          "config": {
          "chainId": 123456,
          "homesteadBlock": 0,
          "eip150Block": 0,
          "eip155Block": 0,
          "eip158Block": 0,
          "byzantiumBlock": 0,
          "constantinopleBlock": 0,
          "petersburgBlock": 0,
          "istanbulBlock": 0,
          "berlinBlock": 0,
          "clique": {
             "period": 0,
		"epoch": 30000
	}
	},
	"difficulty": "0",
	"gasLimit": "15000000",
	"extradata":
"0x0000000000000000000000000000000000000000000000000000000000000000
049F741544Bc885CDDd50BaFb01dfE78Fc4f23AC46Ba548AbC13366912e5D18
7D5Df82219eE7f1320000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000
000000000000000",
	"alloc": {
	"049F741544Bc885CDDd50BaFb01dfE78Fc4f23AC": { "balance":
"9000000000000000000" },
	"46Ba548AbC13366912e5D187D5Df82219eE7f132": { "balance":
"9000000000000000000" },
	"B90bcFd7D6A04635b6e3375E1c3Dcf4d02bf798e": { "balance":
"9000000000000000000" },
	"60f9C7346406b293a6a780663cd9A6D4e0583633": { "balance":
"9000000000000000000" }


	}
    }
```

<u>You must replace the following fields accordingly</u>:

1. **Chain ID**: any desired number. It is suggested not to use chain ID that is already present in https://chainlist.org.

Reason: To prevent replay attack between the Ethereum and Ethereum Classic networks.

https://github.com/ethereum/EIPs/blob/master/EIPS/eip-155.md

Chain ID was introduced in EIP-155 or Ethereum Improvement Proposal 155 standard to prevent replay attacks between the Ethereum chain (ETH) and the Ethereum Classic chain (ETC) which both have a network ID of 1.

Chain ID is required when signing transactions, meaning transactions signed on the ETH network end up with a different hash than those signed on ETC. Before EIP-155, signed transactions on each network would look the same, and could be replayed.

2. **Choose consensus algorithm**: we choose 'Clique' proof-of-authority (PoA) consensus algorithm

https://geth.ethereum.org/docs/interacting-with-geth/rpc/ns-clique

Clique consensus is a PoA system where new blocks can be created by authorized 'signers' or sealer only.

The initial set of authorized signers is configured in the genesis block under alloc tag. Signers can be authorized and de-authorized using a voting mechanism, thus allowing the set of signers to change while the block chain operates.

Sealers take turns proposing new blocks in a round-robin fashion or another predetermined algorithm. This rotation ensures fairness and prevents any single sealer from having undue influence for an extended period. The selected sealer creates a block containing transactions, signs it with their private key, and broadcasts it to the network.

The protocol defines a voting mechanism to dynamically add new signers and remove existing ones. In Geth this can be controlled via the clique. Propose(address, authorized) method

To authorize a new signer, existing ones can propose it via clique.propose("0x...", true). When more than half the signers proposed it, the authorization comes into effect immediately and the new account can start signing blocks.

Similarly, existing signers can propose deauthorizing existing ones via clique.propose("0x...", false). Again if half + 1 signers deauthorize a signer, it is immediately removed from the list and blocks minted by it are rejected from that point onward.

For Example :

clique.propose("0x46Ba548AbC13366912e5D187D5Df82219eE7f132", true)

clique command:

List sealers : clique.getSigners()

List propositions: clique.proposals

Discard a proposition: clique.discard("{NODE_ADDRESS}")

Add a new sealer: clique.propose("{NODE_ADDRESS}", true)

Remove a sealer: clique.propose("{NODE_ADDRESS}", false)

```
"clique": {
    "period": 0,
    "epoch": 30000
}
```

period: Time frame to determine the seconds required to confirm a block

epoch: how many cycles before the algorithm reset.

3. **Extradata:** The extra Data field allows the Clique protocol to identify the initial set of authorized signers and their order

```
 "extradata":
"0x0000000000000000000000000000000000000000000000000000000000000000
{ INITIAL_SIGNER_ADDRESS
}00000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000"
```

Replace the 2 signer addresses (without '0x')

4. **Difficulty**: in PoA usually refers to the interval between block creations or the block time.

If a sealer fails to propose a block within their assigned time slot, the next sealer in line can propose a block

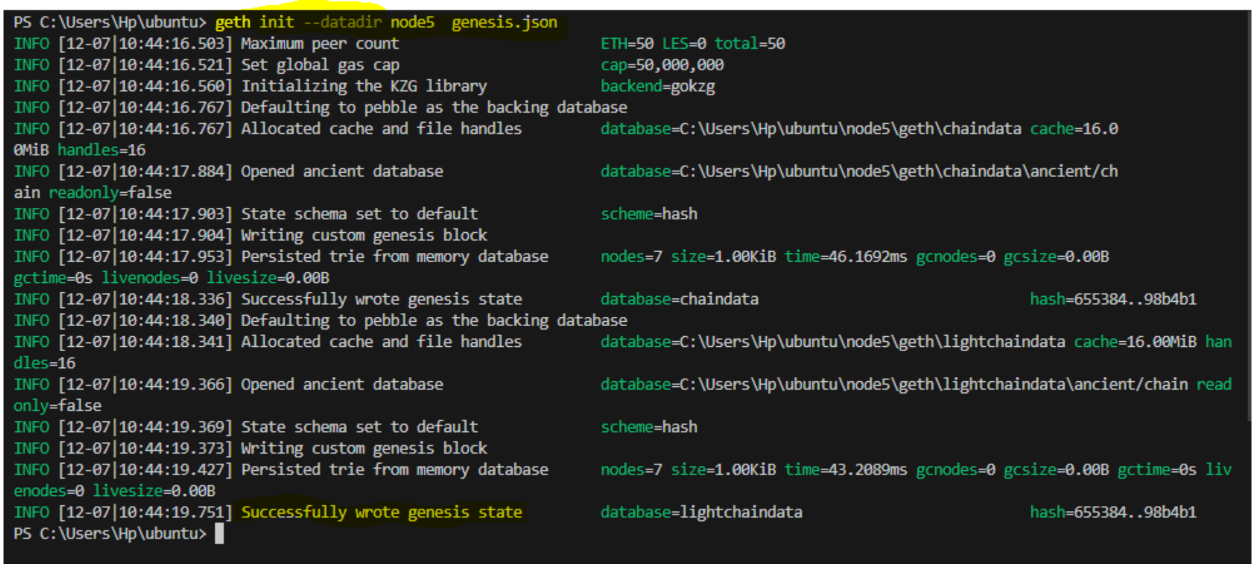
This approach ensures that blocks are created at regular intervals, and the block time is more predictable than in POW networks.

5. **Gas Limit**: Gas is a unit used to measure the computational work required to execute operations or smart contracts on the Ethereum Virtual Machine (EVM).

6. **alloc**: is the initial allocation of ether per account

Step6: Initialize each node with genesis block

Node folder> geth --datadir ./data init ../genesis.json



```
PS C:\Users\Hp\ubuntu> geth init --datadir node5  genesis.json
INFO [12-07|10:44:16.503] Maximum peer count                       ETH=50 LES=0 total=50
INFO [12-07|10:44:16.521] Set global gas cap                       cap=50,000,000
INFO [12-07|10:44:16.560] Initializing the KZG library             backend=gokzg
INFO [12-07|10:44:16.767] Defaulting to pebble as the backing database
INFO [12-07|10:44:16.767] Allocated cache and file handles         database=C:\Users\Hp\ubuntu\node5\geth\chaindata cache=16.0
0MiB handles=16
INFO [12-07|10:44:17.884] Opened ancient database                  database=C:\Users\Hp\ubuntu\node5\geth\chaindata\ancient/ch
ain readonly=false
INFO [12-07|10:44:17.903] State schema set to default              scheme=hash
INFO [12-07|10:44:17.904] Writing custom genesis block
INFO [12-07|10:44:17.953] Persisted trie from memory database      nodes=7 size=1.00KiB time=46.1692ms gcnodes=0 gcsize=0.00B
gctime=0s livenodes=0 livesize=0.00B
INFO [12-07|10:44:18.336] Successfully wrote genesis state         database=chaindata                              hash=655384..98b4b1
INFO [12-07|10:44:18.340] Defaulting to pebble as the backing database
INFO [12-07|10:44:18.341] Allocated cache and file handles         database=C:\Users\Hp\ubuntu\node5\geth\lightchaindata cache=16.00MiB han
dles=16
INFO [12-07|10:44:19.366] Opened ancient database                  database=C:\Users\Hp\ubuntu\node5\geth\lightchaindata\ancient/chain read
only=false
INFO [12-07|10:44:19.369] State schema set to default              scheme=hash
INFO [12-07|10:44:19.373] Writing custom genesis block
INFO [12-07|10:44:19.427] Persisted trie from memory database      nodes=7 size=1.00KiB time=43.2089ms gcnodes=0 gcsize=0.00B gctime=0s liv
enodes=0 livesize=0.00B
INFO [12-07|10:44:19.751] Successfully wrote genesis state         database=lightchaindata                         hash=655384..98b4b1
PS C:\Users\Hp\ubuntu>
```

Step7: Start boot node:

https://ethereum.org/en/developers/docs/nodes-and-clients/bootnodes/

Boot node serve as starting point for other nodes to discover and join the network. It can be dismissed once the nodes are started.

To create bootnode > bootnode -genkey { NAME_OF_THE_KEY }.key

To start bootnode > bootnode -nodekey { KEY_NAME } -verbosity 7 -addr "127.0.0.1:30301"

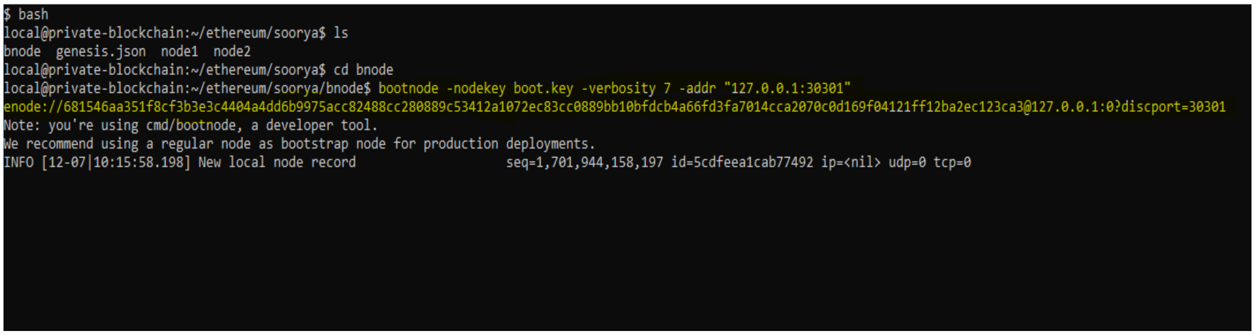
bootnode -nodekey boot.key -verbosity 7 -addr "127.0.0.1:30301"

You can choose any port number but avoid public block chain port.

For example:

Ethereum Classic (ETC) & Ethereum chain uses port: 30303

Once bootnode started, you can find its enode from the terminal



```
$ bash
local@private-blockchain:~/ethereum/soorya$ ls
bnode  genesis.json  node1  node2
local@private-blockchain:~/ethereum/soorya$ cd bnode
local@private-blockchain:~/ethereum/soorya/bnode$ bootnode -nodekey boot.key -verbosity 7 -addr "127.0.0.1:30301"
enode://681546aa351f8cf3b3e3c4404a4dd6b9975acc82488cc280889c53412a1072ec83cc0889bb10bfdcb4a66fd3fa7014cca2070c0d169f04121ff12ba2ec123ca3@127.0.0.1:0?discport=30301
Note: you're using cmd/bootnode, a developer tool.
We recommend using a regular node as bootstrap node for production deployments.
INFO [12-07|10:15:58.198] New local node record                    seq=1,701,944,158,197 id=5cdfeea1cab77492 ip=<nil> udp=0 tcp=0
```

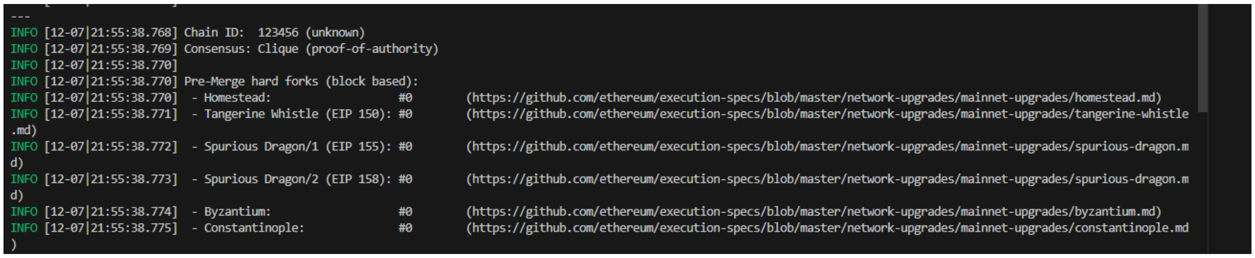
Step8: Start each Node in network:

Node1 command: geth --datadir "./data" --port 30304 --boot nodes enode://{ YOUR_VALUE } --rpc.enabledeprecatedpersonal --allow-insecure-unlock --http --authrpc.port "8551" --http.corsdomain="https://remix.ethereum.org" --http.api web3,eth,debug,personal,net --networkid { NETWORK_ID } --unlock { ADDRESS_NODE1 } --password { PASSWORD_FILE_NAME_EXTENSION } --mine --miner.etherbase= { SIGNER_ADDRESS }

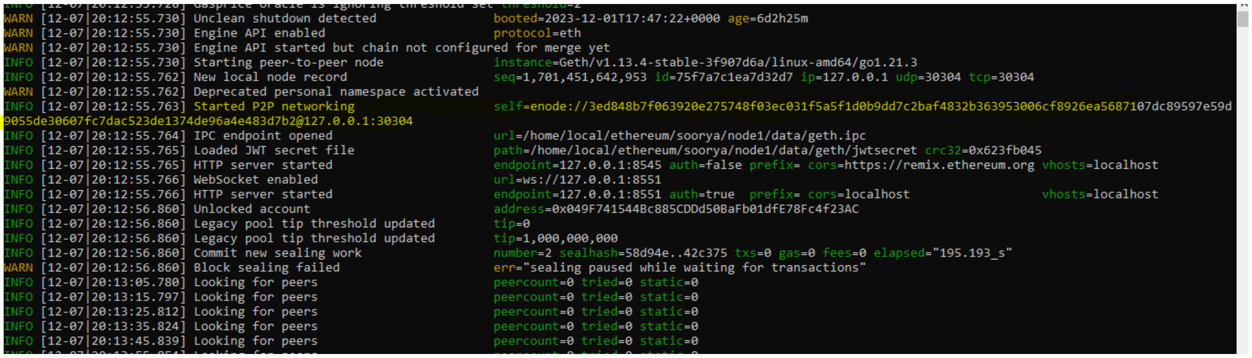
Replace each placeholder field

geth --datadir "./data" --port 30304 --bootnodes enode://681546aa351f8cf3b3e3c4404a4dd6b9975acc82488cc280889c53412a1072ec83cc0889bb10bfdcb4a66fd3fa7014cca2070c0d169f04121ff12ba2ec123ca3@127.0.0.1:0?discport=30301 --rpc.enabledeprecatedpersonal --allow-insecure-unlock --http --authrpc.port "8551" --http.corsdomain="https://remix.ethereum.org" --http.api web3,eth,debug,personal,net --networkid 123456 --unlock 0x049F741544Bc885CDDd50BaFb01dfE78Fc4f23AC --password password.txt --mine --miner.etherbase=0x049F741544Bc885CDDd50BaFb01dfE78Fc4f23AC

Once node1 start:

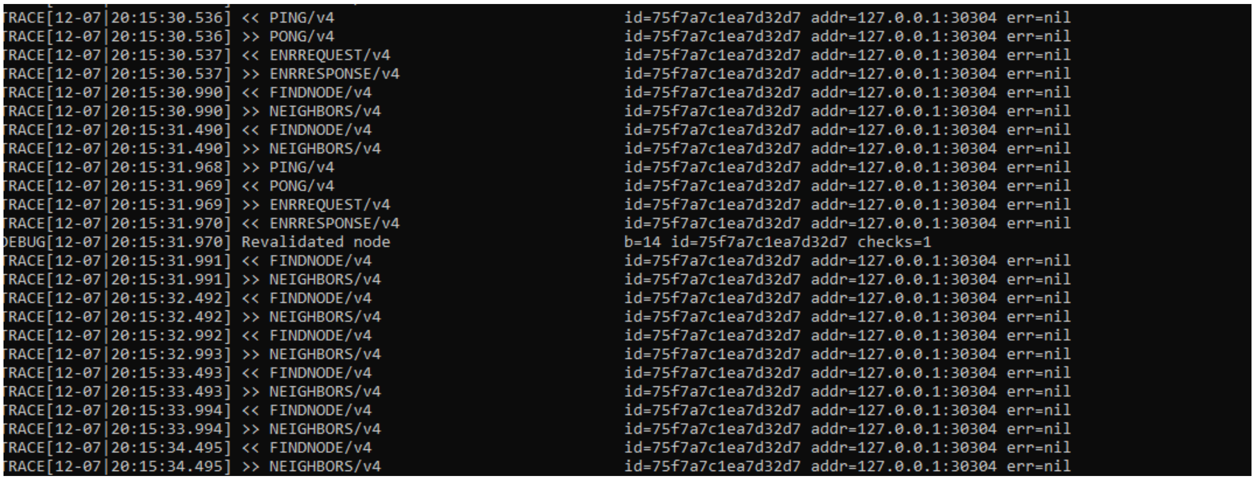


```
---
INFO [12-07|21:55:38.768] Chain ID:  123456 (unknown)
INFO [12-07|21:55:38.769] Consensus: Clique (proof-of-authority)
INFO [12-07|21:55:38.770]
INFO [12-07|21:55:38.770] Pre-Merge hard forks (block based):
INFO [12-07|21:55:38.770]  - Homestead:                   #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/homestead.md)
INFO [12-07|21:55:38.771]  - Tangerine Whistle (EIP 150): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/tangerine-whistle
.md)
INFO [12-07|21:55:38.772]  - Spurious Dragon/1 (EIP 155): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.m
d)
INFO [12-07|21:55:38.773]  - Spurious Dragon/2 (EIP 158): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.m
d)
INFO [12-07|21:55:38.774]  - Byzantium:                   #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/byzantium.md)
INFO [12-07|21:55:38.775]  - Constantinople:              #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/constantinople.md
)
```



```
WARN [12-07|20:12:55.730] Unclean shutdown detected                booted=2023-12-01T17:47:22+0000 age=6d2h25m
WARN [12-07|20:12:55.730] Engine API enabled                       protocol=eth
WARN [12-07|20:12:55.730] Engine API started but chain not configured for merge yet
INFO [12-07|20:12:55.730] Starting peer-to-peer node               instance=Geth/v1.13.4-stable-3f907d6a/linux-amd64/go1.21.3
INFO [12-07|20:12:55.762] New local node record                    seq=1,701,451,642,953 id=75f7a7c1ea7d32d7 ip=127.0.0.1 udp=30304 tcp=30304
WARN [12-07|20:12:55.762] Deprecated personal namespace activated
INFO [12-07|20:12:55.763] Started P2P networking                   self=enode://3ed848b7f063920e275748f03ec031f5a5f1d0b9dd7c2baf4832b363953006cf8926ea5687107dc89597e59d
9055de30607fc7dac523de1374de96a4e483d7b2@127.0.0.1:30304
INFO [12-07|20:12:55.764] IPC endpoint opened                      url=/home/local/ethereum/soorya/node1/data/geth.ipc
INFO [12-07|20:12:55.765] Loaded JWT secret file                   path=/home/local/ethereum/soorya/node1/data/geth/jwtsecret crc32=0x623fb045
INFO [12-07|20:12:55.765] HTTP server started                      endpoint=127.0.0.1:8545 auth=false prefix= cors=https://remix.ethereum.org vhosts=localhost
INFO [12-07|20:12:55.766] WebSocket enabled                        url=ws://127.0.0.1:8551
INFO [12-07|20:12:55.766] HTTP server started                      endpoint=127.0.0.1:8551 auth=true  prefix= cors=localhost                   vhosts=localhost
INFO [12-07|20:12:56.860] Unlocked account                         address=0x049F741544Bc885CDDd50BaFb01dfE78Fc4f23AC
INFO [12-07|20:12:56.860] Legacy pool tip threshold updated        tip=0
INFO [12-07|20:12:56.860] Legacy pool tip threshold updated        tip=1,000,000,000
INFO [12-07|20:12:56.860] Commit new sealing work                  number=2 sealhash=58d94e..42c375 txs=0 gas=0 fees=0 elapsed="195.193_s"
WARN [12-07|20:12:56.860] Block sealing failed                     err="sealing paused while waiting for transactions"
INFO [12-07|20:13:05.780] Looking for peers                        peercount=0 tried=0 static=0
INFO [12-07|20:13:15.797] Looking for peers                        peercount=0 tried=0 static=0
INFO [12-07|20:13:25.812] Looking for peers                        peercount=0 tried=0 static=0
INFO [12-07|20:13:35.824] Looking for peers                        peercount=0 tried=0 static=0
INFO [12-07|20:13:45.839] Looking for peers                        peercount=0 tried=0 static=0
```

You can find node1 port in boot node terminal:

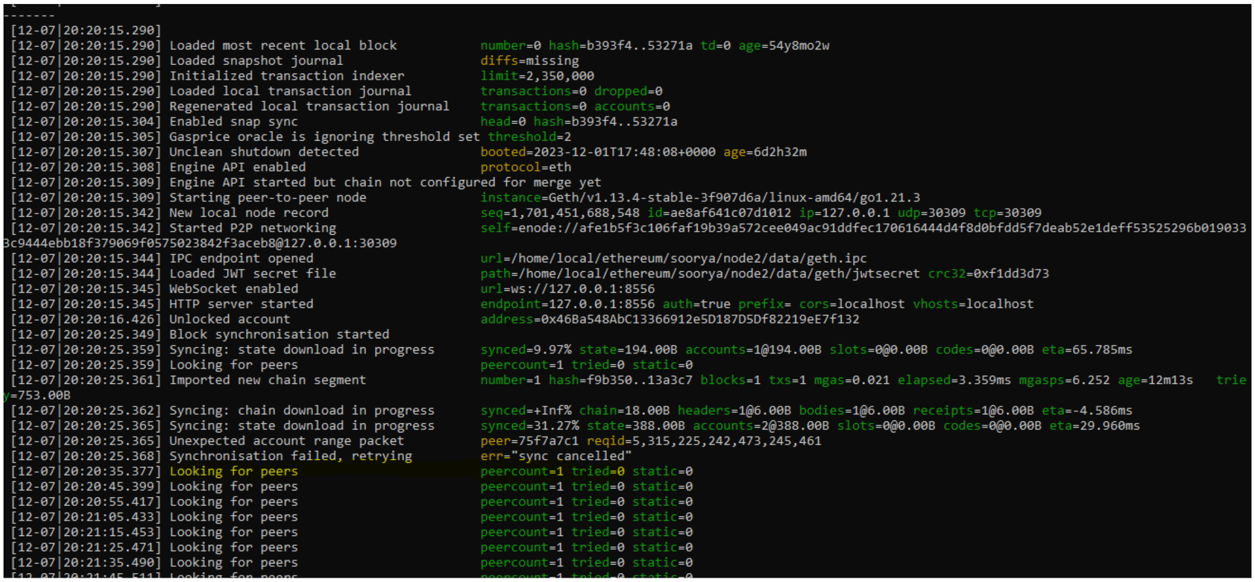


```
RACE[12-07|20:15:30.536] << PING/v4                               id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:30.536] >> PONG/v4                               id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:30.537] << ENRREQUEST/v4                         id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:30.537] >> ENRRESPONSE/v4                        id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:30.990] << FINDNODE/v4                           id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:30.990] >> NEIGHBORS/v4                          id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:31.490] << FINDNODE/v4                           id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:31.490] >> NEIGHBORS/v4                          id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:31.968] >> PING/v4                               id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:31.969] << PONG/v4                               id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:31.969] >> ENRREQUEST/v4                         id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:31.970] << ENRRESPONSE/v4                        id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
EBUG[12-07|20:15:31.970] Revalidated node                         b=14 id=75f7a7c1ea7d32d7 checks=1
RACE[12-07|20:15:31.991] << FINDNODE/v4                           id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:31.991] >> NEIGHBORS/v4                          id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:32.492] << FINDNODE/v4                           id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:32.492] >> NEIGHBORS/v4                          id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:32.992] << FINDNODE/v4                           id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:32.993] >> NEIGHBORS/v4                          id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:33.493] << FINDNODE/v4                           id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:33.493] >> NEIGHBORS/v4                          id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:33.994] << FINDNODE/v4                           id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:33.994] >> NEIGHBORS/v4                          id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:34.495] << FINDNODE/v4                           id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
RACE[12-07|20:15:34.495] >> NEIGHBORS/v4                          id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
```

To start node2 > geth --datadir "./data" --port 30306 --bootnodes enode://{ YOUR_VALUE } --authrpc.port 8556 --networkid { NETWORK_ID } --unlock { ADDRESS_NODE2 } --password { PASSWORD_FILE_WITH_EXTENSION }

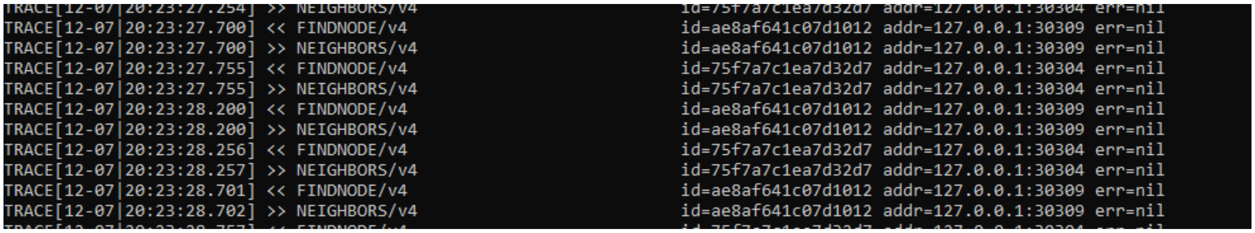
Replace each placeholder field

geth --datadir "./data" --port 30309 --bootnodes enode://681546aa351f8cf3b3e3c4404a4dd6b9975acc82488cc280889c53412a1072ec83cc0889bb10bfdcb4a66fd3fa7014cca2070c0d169f04121ff12ba2ec123ca3@127.0.0.1:0?discport=30301 --authrpc.port 8556 --networkid 123456 --unlock 0x46Ba548AbC13366912e5D187D5Df82219eE7f132 --password password.txt



```
-------
 [12-07|20:20:15.290]
 [12-07|20:20:15.290] Loaded most recent local block           number=0 hash=b393f4..53271a td=0 age=54y8mo2w
 [12-07|20:20:15.290] Loaded snapshot journal                  diffs=missing
 [12-07|20:20:15.290] Initialized transaction indexer          limit=2,350,000
 [12-07|20:20:15.290] Loaded local transaction journal         transactions=0 dropped=0
 [12-07|20:20:15.290] Regenerated local transaction journal    transactions=0 accounts=0
 [12-07|20:20:15.304] Enabled snap sync                        head=0 hash=b393f4..53271a
 [12-07|20:20:15.305] Gasprice oracle is ignoring threshold set threshold=2
 [12-07|20:20:15.307] Unclean shutdown detected                booted=2023-12-01T17:48:08+0000 age=6d2h32m
 [12-07|20:20:15.308] Engine API enabled                       protocol=eth
 [12-07|20:20:15.309] Engine API started but chain not configured for merge yet
 [12-07|20:20:15.309] Starting peer-to-peer node               instance=Geth/v1.13.4-stable-3f907d6a/linux-amd64/go1.21.3
 [12-07|20:20:15.342] New local node record                    seq=1,701,451,688,548 id=ae8af641c07d1012 ip=127.0.0.1 udp=30309 tcp=30309
 [12-07|20:20:15.342] Started P2P networking                   self=enode://afe1b5f3c106faf19b39a572cee049ac91ddfec170616444d4f8d0bfdd5f7deab52e1deff53525296b019033
3c9444ebb18f379069f0575023842f3aceb8@127.0.0.1:30309
 [12-07|20:20:15.344] IPC endpoint opened                      url=/home/local/ethereum/soorya/node2/data/geth.ipc
 [12-07|20:20:15.344] Loaded JWT secret file                   path=/home/local/ethereum/soorya/node2/data/geth/jwtsecret crc32=0xf1dd3d73
 [12-07|20:20:15.345] WebSocket enabled                        url=ws://127.0.0.1:8556
 [12-07|20:20:15.345] HTTP server started                      endpoint=127.0.0.1:8556 auth=true prefix= cors=localhost vhosts=localhost
 [12-07|20:20:16.426] Unlocked account                         address=0x46Ba548AbC13366912e5D187D5Df82219eE7f132
 [12-07|20:20:25.349] Block synchronisation started
 [12-07|20:20:25.359] Syncing: state download in progress      synced=9.97% state=194.00B accounts=1@194.00B slots=0@0.00B codes=0@0.00B eta=65.785ms
 [12-07|20:20:25.359] Looking for peers                        peercount=1 tried=0 static=0
 [12-07|20:20:25.361] Imported new chain segment               number=1 hash=f9b350..13a3c7 blocks=1 txs=1 mgas=0.021 elapsed=3.359ms mgasps=6.252 age=12m13s   trie
y=753.00B
 [12-07|20:20:25.362] Syncing: chain download in progress      synced=+Inf% chain=18.00B headers=1@6.00B bodies=1@6.00B receipts=1@6.00B eta=-4.586ms
 [12-07|20:20:25.365] Syncing: state download in progress      synced=31.27% state=388.00B accounts=2@388.00B slots=0@0.00B codes=0@0.00B eta=29.960ms
 [12-07|20:20:25.365] Unexpected account range packet          peer=75f7a7c1 reqid=5,315,225,242,473,245,461
 [12-07|20:20:25.368] Synchronisation failed, retrying         err="sync cancelled"
 [12-07|20:20:35.377] Looking for peers                        peercount=1 tried=0 static=0
 [12-07|20:20:45.399] Looking for peers                        peercount=1 tried=0 static=0
 [12-07|20:20:55.417] Looking for peers                        peercount=1 tried=0 static=0
 [12-07|20:21:05.433] Looking for peers                        peercount=1 tried=0 static=0
 [12-07|20:21:15.453] Looking for peers                        peercount=1 tried=0 static=0
 [12-07|20:21:25.471] Looking for peers                        peercount=1 tried=0 static=0
 [12-07|20:21:35.490] Looking for peers                        peercount=1 tried=0 static=0
 [12-07|20:21:45.511] Looking for peers                        peercount=1 tried=0 static=0
```

Now, Boot node terminal will show two node address:



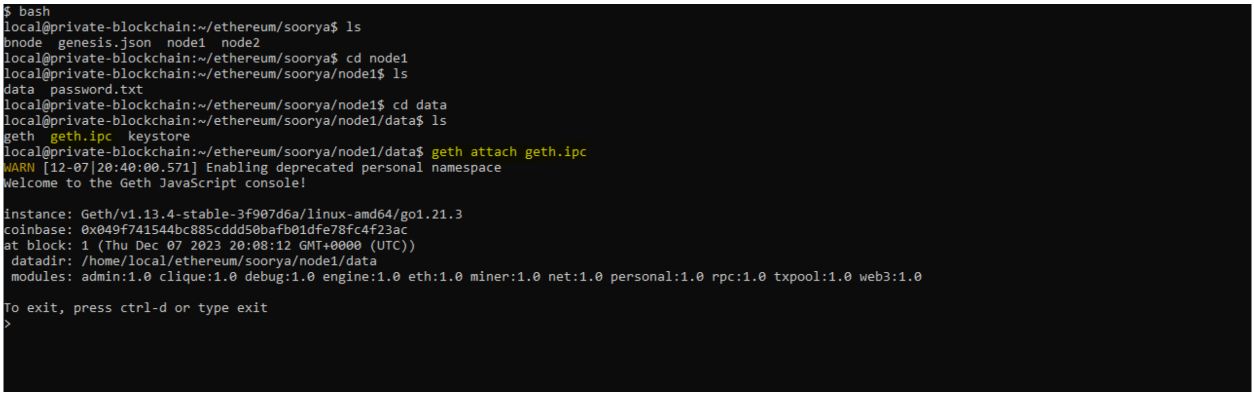
```
TRACE[12-07|20:23:27.254] >> NEIGHBORS/v4                 id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
TRACE[12-07|20:23:27.700] << FINDNODE/v4                  id=ae8af641c07d1012 addr=127.0.0.1:30309 err=nil
TRACE[12-07|20:23:27.700] >> NEIGHBORS/v4                 id=ae8af641c07d1012 addr=127.0.0.1:30309 err=nil
TRACE[12-07|20:23:27.755] << FINDNODE/v4                  id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
TRACE[12-07|20:23:27.755] >> NEIGHBORS/v4                 id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
TRACE[12-07|20:23:28.200] << FINDNODE/v4                  id=ae8af641c07d1012 addr=127.0.0.1:30309 err=nil
TRACE[12-07|20:23:28.200] >> NEIGHBORS/v4                 id=ae8af641c07d1012 addr=127.0.0.1:30309 err=nil
TRACE[12-07|20:23:28.256] << FINDNODE/v4                  id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
TRACE[12-07|20:23:28.257] >> NEIGHBORS/v4                 id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
TRACE[12-07|20:23:28.701] << FINDNODE/v4                  id=ae8af641c07d1012 addr=127.0.0.1:30309 err=nil
TRACE[12-07|20:23:28.702] >> NEIGHBORS/v4                 id=ae8af641c07d1012 addr=127.0.0.1:30309 err=nil
TRACE[12-07|20:23:28.757] << FINDNODE/v4                  id=75f7a7c1ea7d32d7 addr=127.0.0.1:30304 err=nil
```

To start node3 & node 4

First we need find enode of signer node.

To do that we have to open geth javascript console:

geth attach geth.ipc command from node1 folder



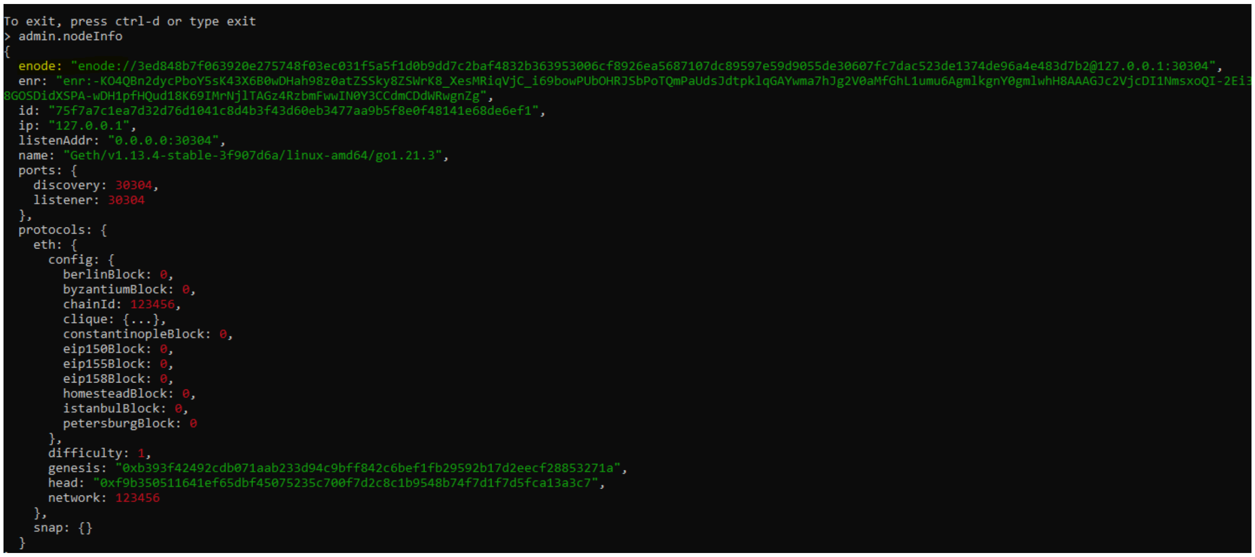
```
$ bash
local@private-blockchain:~/ethereum/soorya$ ls
bnode  genesis.json  node1  node2
local@private-blockchain:~/ethereum/soorya$ cd node1
local@private-blockchain:~/ethereum/soorya/node1$ ls
data  password.txt
local@private-blockchain:~/ethereum/soorya/node1$ cd data
local@private-blockchain:~/ethereum/soorya/node1/data$ ls
geth  geth.ipc  keystore
local@private-blockchain:~/ethereum/soorya/node1/data$ geth attach geth.ipc
WARN [12-07|20:40:00.571] Enabling deprecated personal namespace
Welcome to the Geth JavaScript console!

instance: Geth/v1.13.4-stable-3f907d6a/linux-amd64/go1.21.3
coinbase: 0x049f741544bc885cddd50bafb01dfe78fc4f23ac
at block: 1 (Thu Dec 07 2023 20:08:12 GMT+0000 (UTC))
 datadir: /home/local/ethereum/soorya/node1/data
 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
>
```

To Find enode:

Enter admin.nodeInfo from javascript console



```
To exit, press ctrl-d or type exit
> admin.nodeInfo
{
  enode: "enode://3ed848b7f063920e275748f03ec031f5a5f1d0b9dd7c2baf4832b363953006cf8926ea5687107dc89597e59d9055de30607fc7dac523de1374de96a4e483d7b2@127.0.0.1:30304",
  enr: "enr:-KO4QBn2dycPboY5sK43X6B0wDHah98z0atZSSky8ZSWrK8_XesMRiqVjC_i69bowPUbOHRJSbPoTQmPaUdsJdtpklqGAYwma7hJg2V0aMfGhL1umu6AgmlkgnY0gmlwhH8AAAGJc2VjcDI1NmsxoQI-2Ei3
8GOSDidXSPA-wDH1pfHQud18K69IMrNjlTAGz4RzbmFwwIN0Y3CCdmCDdWRwgnZg",
  id: "75f7a7c1ea7d32d76d1041c8d4b3f43d60eb3477aa9b5f8e0f48141e68de6ef1",
  ip: "127.0.0.1",
  listenAddr: "0.0.0.0:30304",
  name: "Geth/v1.13.4-stable-3f907d6a/linux-amd64/go1.21.3",
  ports: {
    discovery: 30304,
    listener: 30304
  },
  protocols: {
    eth: {
      config: {
        berlinBlock: 0,
        byzantiumBlock: 0,
        chainId: 123456,
        clique: {...},
        constantinopleBlock: 0,
        eip150Block: 0,
        eip155Block: 0,
        eip158Block: 0,
        homesteadBlock: 0,
        istanbulBlock: 0,
        petersburgBlock: 0
      },
      difficulty: 1,
      genesis: "0xb393f42492cdb071aab233d94c9bff842c6bef1fb29592b17d2eecf28853271a",
      head: "0xf9b350511641ef65dbf45075235c700f7d2c8c1b9548b74f7d1f7d5fca13a3c7",
      network: 123456
    },
    snap: {}
  }
```

.

Start node3 with node1(signer) enode:

geth --datadir "./data" --port 30306 --bootnodes enode://3ed848b7f063920e275748f03ec031f5a5f1d0b9dd7c2baf4832b363953006cf8926ea5687107dc89597e59d9055de30607fc7dac523de1374de96a4e483d7b2@141.45.212.243:30304 --networkid 123456 --ipcdisable --allow-insecure-unlock --

authrpc.port "8546" --unlock 0xB90bcFd7D6A04635b6e3375E1c3Dcf4d02bf798e --password password.txt

Start node4 with node1(signer) enode:

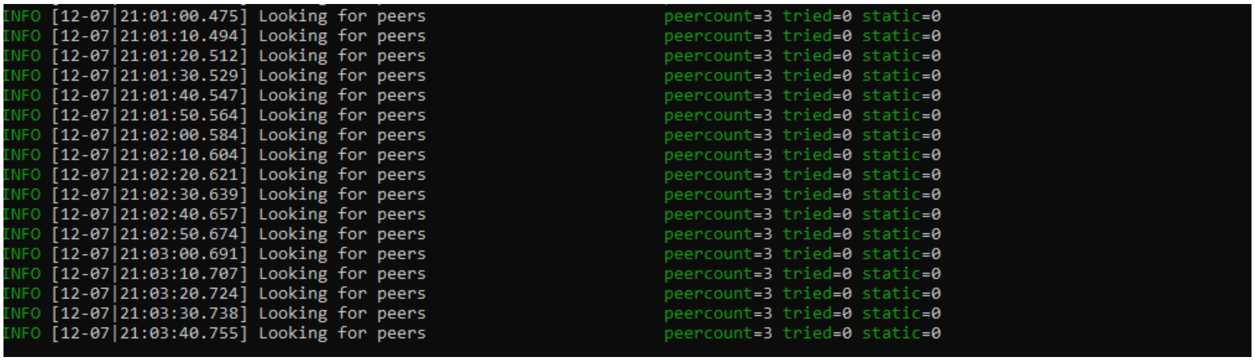
geth --datadir "./data" --port 30308 --bootnodes enode://3ed848b7f063920e275748f03ec031f5a5f1d0b9dd7c2baf4832b363953006cf8926ea5687107dc89597e59d9055de30607fc7dac523de1374de96a4e483d7b2@141.45.212.243:30304 --networkid 123456 --ipcdisable --allow-insecure-unlock --authrpc.port "8587" --unlock 0x60f9C7346406b293a6a780663cd9A6D4e0583633 --password password.txt

Make sure authrpc.port no should be different for each node.

Authrpc(remote procedure call) port serves as interface that allow other nodes or application or external program to interact with that node

When you check terminal of each node:

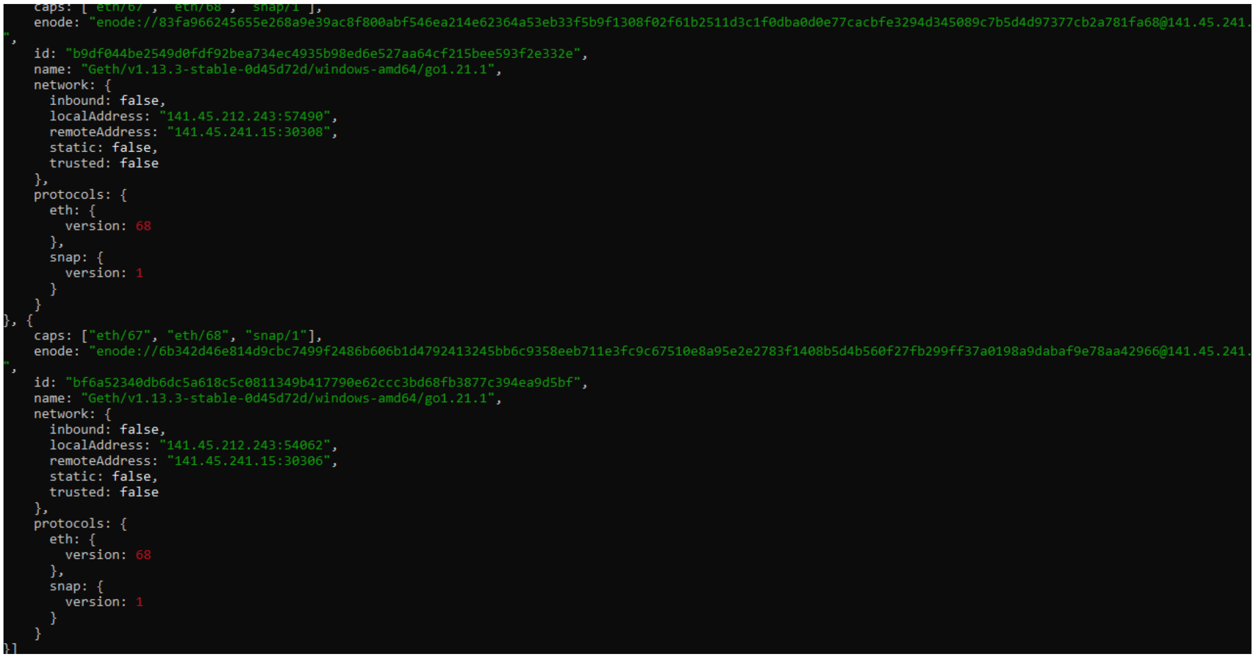
You can see no of Peer Connected



```
INFO [12-07|21:01:00.475] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:01:10.494] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:01:20.512] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:01:30.529] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:01:40.547] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:01:50.564] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:02:00.584] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:02:10.604] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:02:20.621] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:02:30.639] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:02:40.657] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:02:50.674] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:03:00.691] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:03:10.707] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:03:20.724] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:03:30.738] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:03:40.755] Looking for peers                        peercount=3 tried=0 static=0
```

You can also get information about peer from console

admin.peers



```
    caps: ["eth/67", "eth/68", "snap/1"],
    enode: "enode://83fa966245655e268a9e39ac8f800abf546ea214e62364a53eb33f5b9f1308f02f61b2511d3c1f0dba0d0e77cacbfe3294d345089c7b5d4d97377cb2a781fa68@141.45.241.
",
    id: "b9df044be2549d0fdf92bea734ec4935b98ed6e527aa64cf215bee593f2e332e",
    name: "Geth/v1.13.3-stable-0d45d72d/windows-amd64/go1.21.1",
    network: {
      inbound: false,
      localAddress: "141.45.212.243:57490",
      remoteAddress: "141.45.241.15:30308",
      static: false,
      trusted: false
    },
    protocols: {
      eth: {
        version: 68
      },
      snap: {
        version: 1
      }
    }
}, {
    caps: ["eth/67", "eth/68", "snap/1"],
    enode: "enode://6b342d46e814d9cbc7499f2486b606b1d4792413245bb6c9358eeb711e3fc9c67510e8a95e2e2783f1408b5d4b560f27fb299ff37a0198a9dabaf9e78aa42966@141.45.241.
",
    id: "bf6a52340db6dc5a618c5c0811349b417790e62ccc3bd68fb3877c394ea9d5bf",
    name: "Geth/v1.13.3-stable-0d45d72d/windows-amd64/go1.21.1",
    network: {
      inbound: false,
      localAddress: "141.45.212.243:54062",
      remoteAddress: "141.45.241.15:30306",
      static: false,
      trusted: false
    },
    protocols: {
      eth: {
        version: 68
      },
      snap: {
        version: 1
      }
    }
}]
```

Things to verify once network is up:

- Check whether all nodes are synced to current state:

  If eth.syncing is false and eth.blockNumber is non-zero, then most likely it mean node completed syncing

- If peer not connected automatically, you can add peer manually using:

  admin.addPeer() method

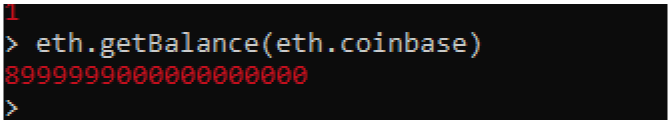
  admin.addPeer("enode://90b7cbbaee94ab6e5bc7c5e8080bf8e2dfed5047b7c19ac61ee82511bef40faf9be2066258228ce7a71ab97b508dbef3c8f50fcf31dedfd43f2f0abd7f618db9@172.129.23.46:30303?discport=0")

We have now successfully created private block chain with four nodes

Interaction between nodes:

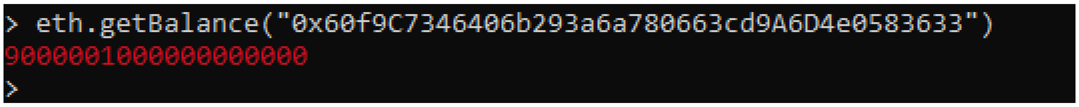
We can now try to do transaction from node1 account to node4 account(send ether):

Before that we need to check balance of node1 & node4



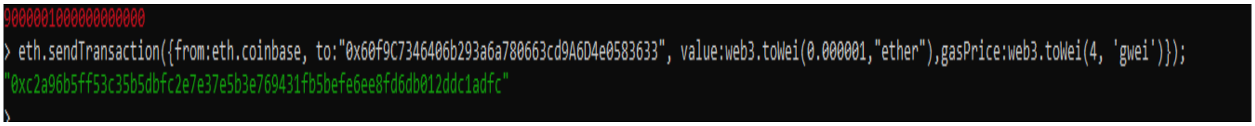
```
1
> eth.getBalance(eth.coinbase)
8999999000000000000
>
```

Node4:

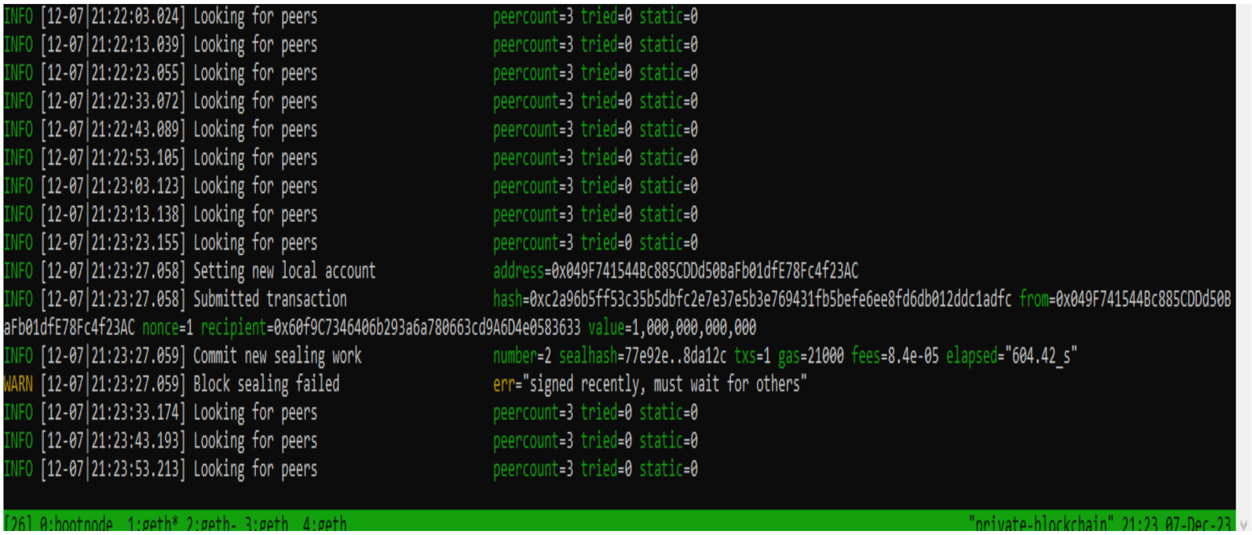


```
> eth.getBalance("0x60f9C7346406b293a6a780663cd9A6D4e0583633")
9000001000000000000
>
```

Send transaction from node1 to node4



```
9000001000000000000
> eth.sendTransaction({from:eth.coinbase, to:"0x60f9C7346406b293a6a780663cd9A6D4e0583633", value:web3.toWei(0.000001,"ether"),gasPrice:web3.toWei(4, 'gwei')});
"0xc2a96b5ff53c35b5dbfc2e7e37e5b3e769431fb5befe6ee8fd6db012ddc1adfc"
>
```

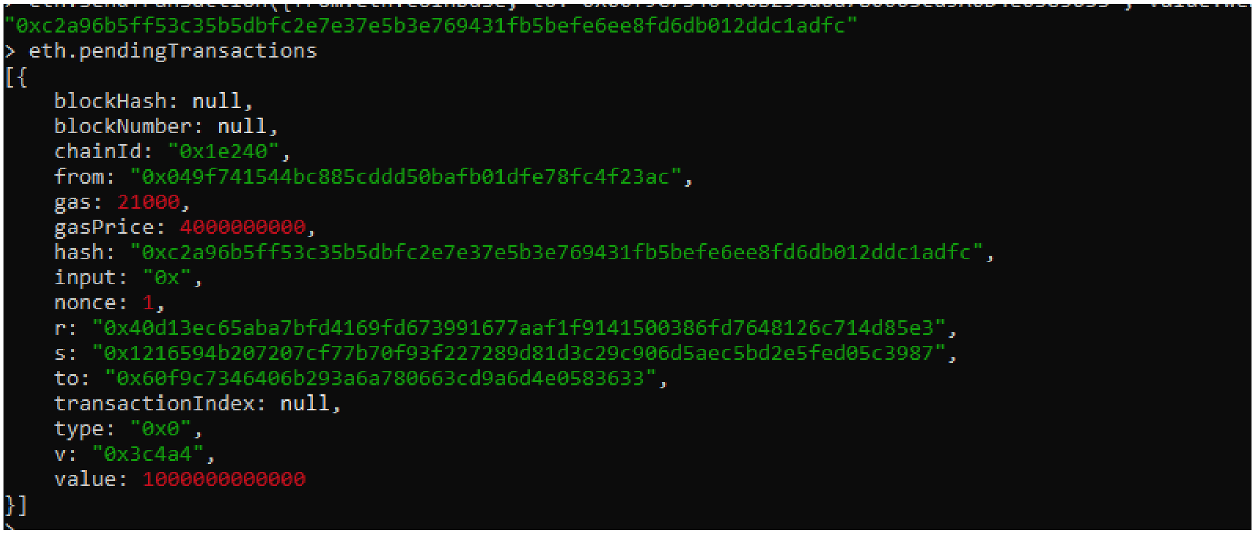


```
INFO [12-07|21:22:03.024] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:22:13.039] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:22:23.055] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:22:33.072] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:22:43.089] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:22:53.105] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:23:03.123] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:23:13.138] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:23:23.155] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:23:27.058] Setting new local account               address=0x049F741544Bc885CDDd50BaFb01dfE78Fc4f23AC
INFO [12-07|21:23:27.058] Submitted transaction                    hash=0xc2a96b5ff53c35b5dbfc2e7e37e5b3e769431fb5befe6ee8fd6db012ddc1adfc from=0x049F741544Bc885CDDd50B
aFb01dfE78Fc4f23AC nonce=1 recipient=0x60f9C7346406b293a6a780663cd9A6D4e0583633 value=1,000,000,000,000
INFO [12-07|21:23:27.059] Commit new sealing work                  number=2 sealhash=77e92e..8da12c txs=1 gas=21000 fees=8.4e-05 elapsed="604.42_s"
WARN [12-07|21:23:27.059] Block sealing failed                     err="signed recently, must wait for others"
INFO [12-07|21:23:33.174] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:23:43.193] Looking for peers                        peercount=3 tried=0 static=0
INFO [12-07|21:23:53.213] Looking for peers                        peercount=3 tried=0 static=0

[26] 0:bootnode  1:geth* 2:geth- 3:geth  4:geth                                        "private-blockchain" 21:23 07-Dec-23
```

Somehow transaction goes into pending state

You can find pending transaction using eth.pendingTrasactions



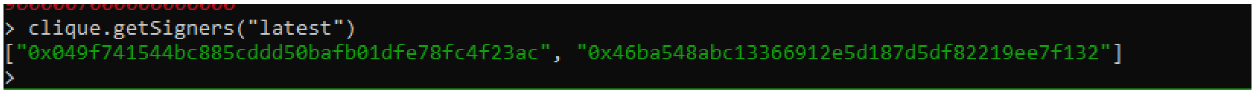
```
"0xc2a96b5ff53c35b5dbfc2e7e37e5b3e769431fb5befe6ee8fd6db012ddc1adfc"
> eth.pendingTransactions
[{
    blockHash: null,
    blockNumber: null,
    chainId: "0x1e240",
    from: "0x049f741544bc885cddd50bafb01dfe78fc4f23ac",
    gas: 21000,
    gasPrice: 4000000000,
    hash: "0xc2a96b5ff53c35b5dbfc2e7e37e5b3e769431fb5befe6ee8fd6db012ddc1adfc",
    input: "0x",
    nonce: 1,
    r: "0x40d13ec65aba7bfd4169fd673991677aaf1f9141500386fd7648126c714d85e3",
    s: "0x1216594b207207cf77b70f93f227289d81d3c29c906d5aec5bd2e5fed05c3987",
    to: "0x60f9c7346406b293a6a780663cd9a6d4e0583633",
    transactionIndex: null,
    type: "0x0",
    v: "0x3c4a4",
    value: 1000000000000
}]
```

Reason for it is:

Block sealing failed

On a clique (proof-of-authority) network this error might occur when the sealer nodes are not up. Remember that in a proof-of-authority network **the 51% of sealers must be up and running** and also connected to the network

You can find the list of signer in the network by using:



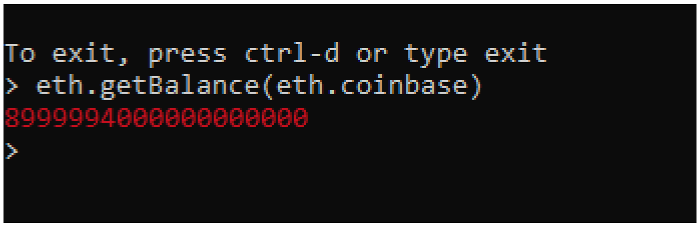
```
> clique.getSigners("latest")
["0x049f741544bc885cddd50bafb01dfe78fc4f23ac", "0x46ba548abc13366912e5d187d5df82219ee7f132"]
>
```

In our case node2 is signer - but it didn’t start to mine.so I need to restart my node2

```
geth --datadir "./data"  --port 30309 --bootnodes enode://681546aa351f8cf3b3e3c4404a4dd6b9975acc82488cc280889c53412a1072ec83cc0889bb10bfdcb4a66fd3fa7014cca2070c0d169f04121ff12ba2ec123ca3@127.0.0.1:0?discport=30301 --authrpc.port 8556 --allow-insecure-unlock --http.api web3,eth,debug,personal,net --networkid 123456 --unlock 0x46Ba548AbC13366912e5D187D5Df82219eE7f132 --password password.txt --mine --miner.etherbase=0x46Ba548AbC13366912e5D187D5Df82219eE7f132
```

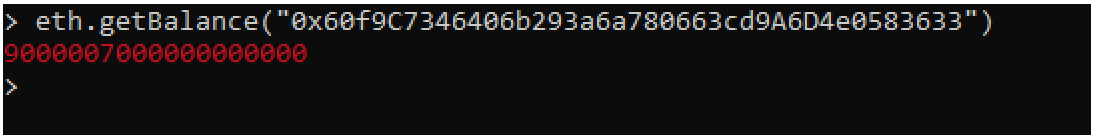
Again we will check another transaction:

Node1 Account balance:



```
To exit, press ctrl-d or type exit
> eth.getBalance(eth.coinbase)
8999994000000000000
>
```

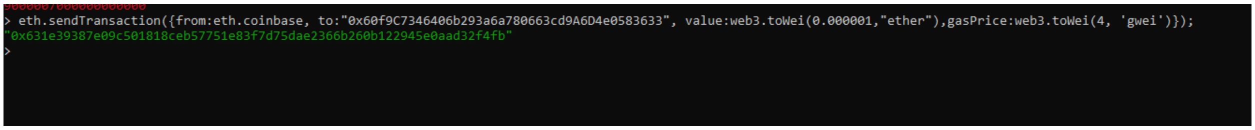
Node4 Account balance:



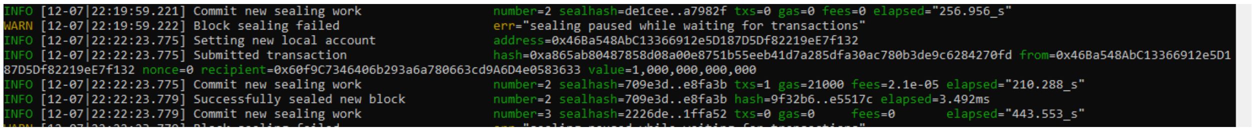
```
> eth.getBalance("0x60f9C7346406b293a6a780663cd9A6D4e0583633")
9000007000000000000
>
```

Send transaction

eth.sendTransaction({from:eth.coinbase, to:"0x60f9C7346406b293a6a780663cd9A6D4e0583633", value:web3.toWei(0.000001,"ether"),gasPrice:web3.toWei(4, 'gwei')});

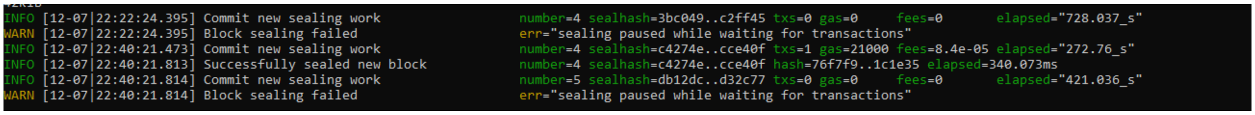


```
9000007000000000000
> eth.sendTransaction({from:eth.coinbase, to:"0x60f9C7346406b293a6a780663cd9A6D4e0583633", value:web3.toWei(0.000001,"ether"),gasPrice:web3.toWei(4, 'gwei')});
"0x631e39387e09c501818ceb57751e83f7d75dae2366b260b122945e0aad32f4fb"
>
```



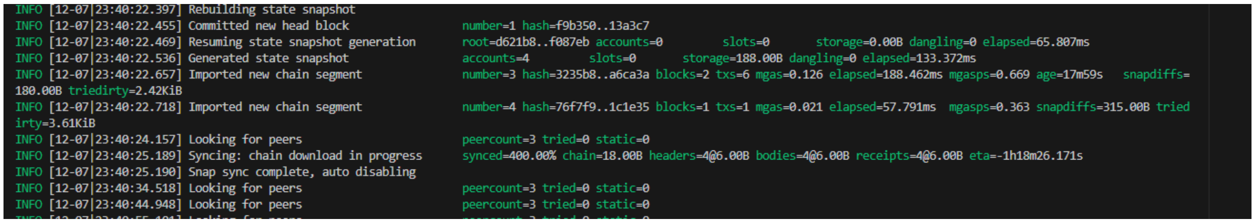
```
INFO [12-07|22:19:59.221] Commit new sealing work                  number=2 sealhash=de1cee..a7982f txs=0 gas=0 fees=0 elapsed="256.956_s"
WARN [12-07|22:19:59.222] Block sealing failed                     err="sealing paused while waiting for transactions"
INFO [12-07|22:22:23.775] Setting new local account                address=0x46Ba548AbC13366912e5D187D5Df82219eE7f132
INFO [12-07|22:22:23.775] Submitted transaction                    hash=0xa865ab80487858d08a00e8751b55eeb41d7a285dfa30ac780b3de9c6284270fd from=0x46Ba548AbC13366912e5D1
87D5Df82219eE7f132 nonce=0 recipient=0x60f9C7346406b293a6a780663cd9A6D4e0583633 value=1,000,000,000,000
INFO [12-07|22:22:23.775] Commit new sealing work                  number=2 sealhash=709e3d..e8fa3b txs=1 gas=21000 fees=2.1e-05 elapsed="210.288_s"
INFO [12-07|22:22:23.779] Successfully sealed new block            number=2 sealhash=709e3d..e8fa3b hash=9f32b6..e5517c elapsed=3.492ms
INFO [12-07|22:22:23.779] Commit new sealing work                  number=3 sealhash=2226de..1ffa52 txs=0 gas=0     fees=0       elapsed="443.553_s"
WARN [12-07|22:22:23.779] Block sealing failed                     err="sealing paused while waiting for transactions"
```

Successfully created block:



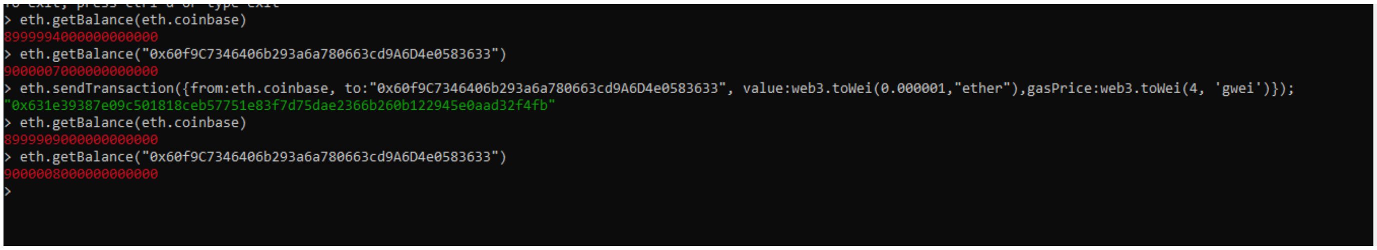
```
INFO [12-07|22:22:24.395] Commit new sealing work                  number=4 sealhash=3bc049..c2ff45 txs=0 gas=0     fees=0       elapsed="728.037_s"
WARN [12-07|22:22:24.395] Block sealing failed                     err="sealing paused while waiting for transactions"
INFO [12-07|22:40:21.473] Commit new sealing work                  number=4 sealhash=c4274e..cce40f txs=1 gas=21000 fees=8.4e-05 elapsed="272.76_s"
INFO [12-07|22:40:21.813] Successfully sealed new block            number=4 sealhash=c4274e..cce40f hash=76f7f9..1c1e35 elapsed=340.073ms
INFO [12-07|22:40:21.814] Commit new sealing work                  number=5 sealhash=db12dc..d32c77 txs=0 gas=0     fees=0       elapsed="421.036_s"
WARN [12-07|22:40:21.814] Block sealing failed                     err="sealing paused while waiting for transactions"
```

Each node import the new block which is created in network



```
 INFO [12-07|23:40:22.397] Rebuilding state snapshot
 INFO [12-07|23:40:22.455] Committed new head block                 number=1 hash=f9b350..13a3c7
 INFO [12-07|23:40:22.469] Resuming state snapshot generation       root=d621b8..f087eb accounts=0         slots=0       storage=0.00B dangling=0 elapsed=65.807ms
 INFO [12-07|23:40:22.536] Generated state snapshot                 accounts=4         slots=0       storage=188.00B dangling=0 elapsed=133.372ms
 INFO [12-07|23:40:22.657] Imported new chain segment               number=3 hash=3235b8..a6ca3a blocks=2 txs=6 mgas=0.126 elapsed=188.462ms mgasps=0.669 age=17m59s   snapdiffs=
 180.00B triedirty=2.42KiB
 INFO [12-07|23:40:22.718] Imported new chain segment               number=4 hash=76f7f9..1c1e35 blocks=1 txs=1 mgas=0.021 elapsed=57.791ms  mgasps=0.363 snapdiffs=315.00B tried
 irty=3.61KiB
 INFO [12-07|23:40:24.157] Looking for peers                        peercount=3 tried=0 static=0
 INFO [12-07|23:40:25.189] Syncing: chain download in progress      synced=400.00% chain=18.00B headers=4@6.00B bodies=4@6.00B receipts=4@6.00B eta=-1h18m26.171s
 INFO [12-07|23:40:25.190] Snap sync complete, auto disabling
 INFO [12-07|23:40:34.518] Looking for peers                        peercount=3 tried=0 static=0
 INFO [12-07|23:40:44.948] Looking for peers                        peercount=3 tried=0 static=0
```

Check balance:



```
To exit, press ctrl-d or type exit
> eth.getBalance(eth.coinbase)
8999994000000000000
> eth.getBalance("0x60f9C7346406b293a6a780663cd9A6D4e0583633")
9000007000000000000
> eth.sendTransaction({from:eth.coinbase, to:"0x60f9C7346406b293a6a780663cd9A6D4e0583633", value:web3.toWei(0.000001,"ether"),gasPrice:web3.toWei(4, 'gwei')});
"0x631e39387e09c501818ceb57751e83f7d75dae2366b260b122945e0aad32f4fb"
> eth.getBalance(eth.coinbase)
8999909000000000000
> eth.getBalance("0x60f9C7346406b293a6a780663cd9A6D4e0583633")
9000008000000000000
>
```

Node1Account was able to send ether to node 4 account successfully