Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 5**	
Title of Lab Assignment: Implementation of block(Genesis Block) and private chain using geth.		
DOP: 20-08-2024	**DOS: 04-09-2024**	
CO Mapped: **CO3, CO4**	**PO Mapped:** **PO1, PO2, PO3, PO7, PO9, PSO1**	**Signature:**

Practical No. 5

Aim: Implementation of block(Genesis Block) and private chain using geth.

Description:

What is a node?

- ➢ A node is generally a point of intersection or connection in a telecommunications network. A node may also mean any system or physical device that is connected to a network and can execute certain functions like creating, receiving or sending information via a communication channel. The explanation of a node varies depending on the protocol layer being referred to.
- ➢ For example, a basic resident network may consist of a file server, two laptops and a fax machine. In this case, the network has four nodes, each equipped with a MAC address to uniquely identify them.
- ➢ The most popular usage of the term “node” is seen in the blockchain space. In this guide, we will explain what nodes are in more detail, including the different types of blockchain nodes being used today.
- ➢ In Ethereum, a user can run three different kinds of nodes: light, full and archive. Their differences lie in how fast they can synchronize with the entire network.
- ➢ There are many ways to run your own Ethereum node, but some popular hardware that can work on the network are DAppNode and Avado. Ethereum nodes have almost the same requirements as Bitcoin nodes, only that the former requires less computing power.
- ➢ Note that before you run an Ethereum node, it is advisable to check your bandwidth limitations first.
- ➢ Ethereum nodes are essential in keeping its blockchain network secure and reliable, as well as transparent. In fact, anyone can view the nodes and their performances on the network via Etherscan’s node tracker.
- ➢ In order to receive block rewards, you would have to run an Ethereum staking node.

Geth:

Geth (Go Ethereum) is a command line interface for running Ethereum node implemented in Go Language. Using Geth you can join Ethereum network, transfer ether between accounts or even mine ethers.

Genesis Block:

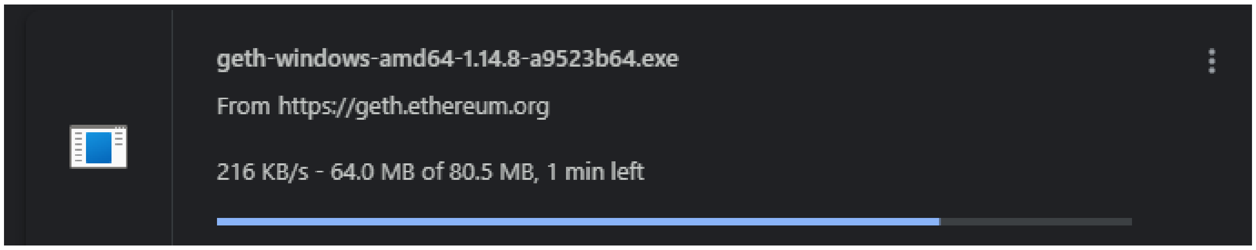
- A genesis block refers to the first block in a blockchain and is usually hardcoded into its application's software. A blockchain has multiple "blocks" (containing validated transactions and recorded activity data) linked together by a metaphorical chain.
- Each "block" of a crypto asset contains referential data for the previous one and derives its value/legitimacy from its predecessor.The genesis block, thus, refers to the first block (Block 0 or Block 1) of a new blockchain, to which all other subsequent blocks are attached.
- A genesis block is unique as it is the only block in a blockchain that does not reference a predecessor block, and in almost all cases, the first mining rewards it unlocks are unspendable.
- Genesis blocks have special significance, as they form the very foundation of a blockchain and often contain interesting stories or hidden meanings. For instance, Bitcoin's genesis block contains the now-famous message "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks" — a reference to the deteriorating financial conditions of that time and the rationale for creating cryptocurrencies like Bitcoin and Ethereum. Bitcoin's genesis block in 2009 contained 50 BTC.
- The Bitcoin genesis block is very intriguing not just for its included message, but also due to the fact that the next block was timestamped nearly six days later (the average time is 10 minutes).

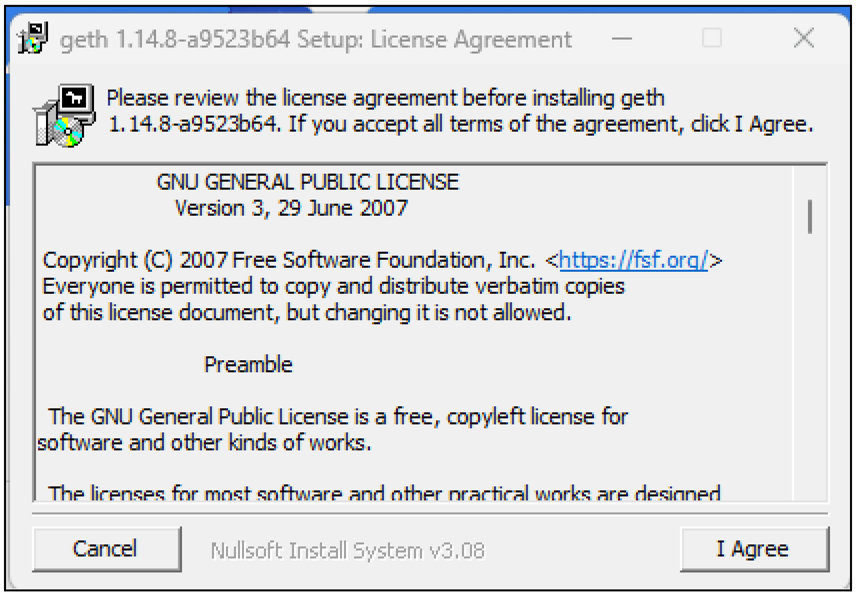
Commands used in this practical:

- **geth --datadir chaindata init genesis.json:** Initializes geth into chaindata.
- **geth --datadir=./chaindata/:** Used to run geth.
- **geth attach ipc:\\.\pipe\geth.ipc:** IPC to interact with geth.
- **personal.newAccount():** Create a new account.
- **eth.accounts:** Get information about all the accounts present.
- **eth.coinbase:** Get information about the coinbase account.
- **eth.getBalance(eth.accounts[0]):** Get the current balance of an account.
- **miner.start() / miner.stop():** Start/Stop the mining process.
- **eth.blockNumber:** Get the blockNumber.
- **personal.unlockAccount(eth.accounts[0]):** Unlock the coinbase account for transaction.

- **eth.sendTransaction({from: eth.coinbase, to: eth.accounts[1], value: web3.toWei(10,"ether")}):** Command for transaction. Enter the account number to which we want to transfer the ethers to.
- **eth.getTransaction(txHash):** get the information about the transaction. Enter the hash instead of "txHash".
- **web3.fromWei(eth.getBalance(eth.accounts[1]), "ether"):** get balance of the second account in ethers.
- **eth.getBlock("latest"):** Get information about the latest block.
- **eth.getBlock(388):** Get information about a specific block.

Output:


geth-windows-amd64-1.14.8-a9523b64.exe
From https://geth.ethereum.org
216 KB/s - 64.0 MB of 80.5 MB, 1 min left


geth 1.14.8-a9523b64 Setup: License Agreement

Please review the license agreement before installing geth 1.14.8-a9523b64. If you accept all terms of the agreement, click I Agree.

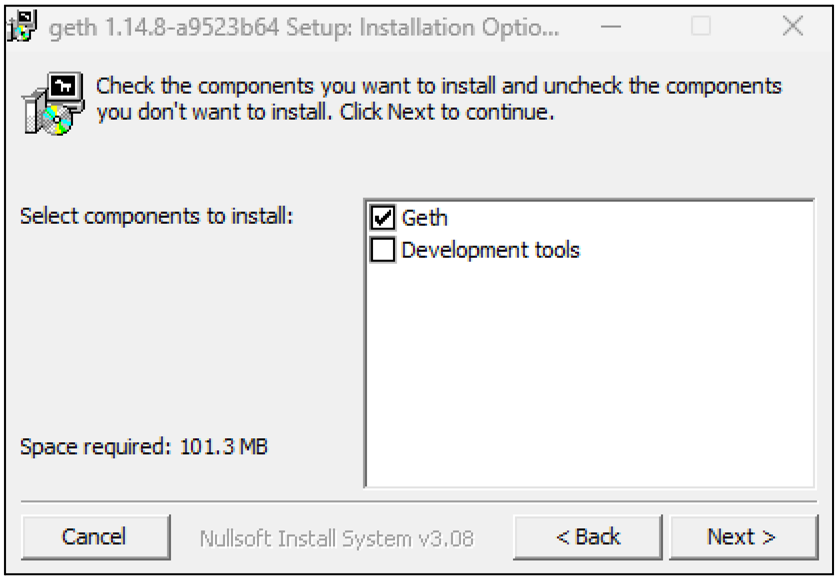
GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed



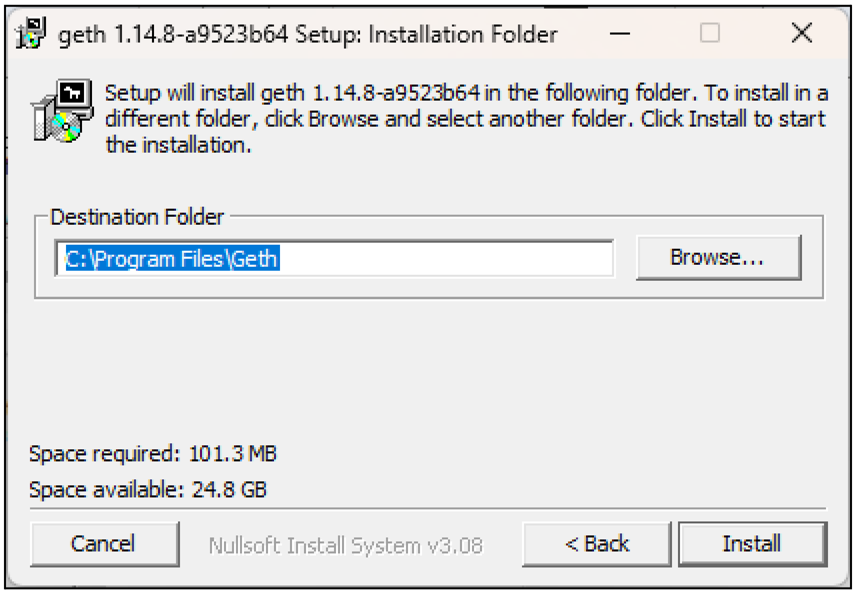
geth 1.14.8-a9523b64 Setup: Installation Optio...

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- [x] Geth
- [ ] Development tools

Space required: 101.3 MB



geth 1.14.8-a9523b64 Setup: Installation Folder

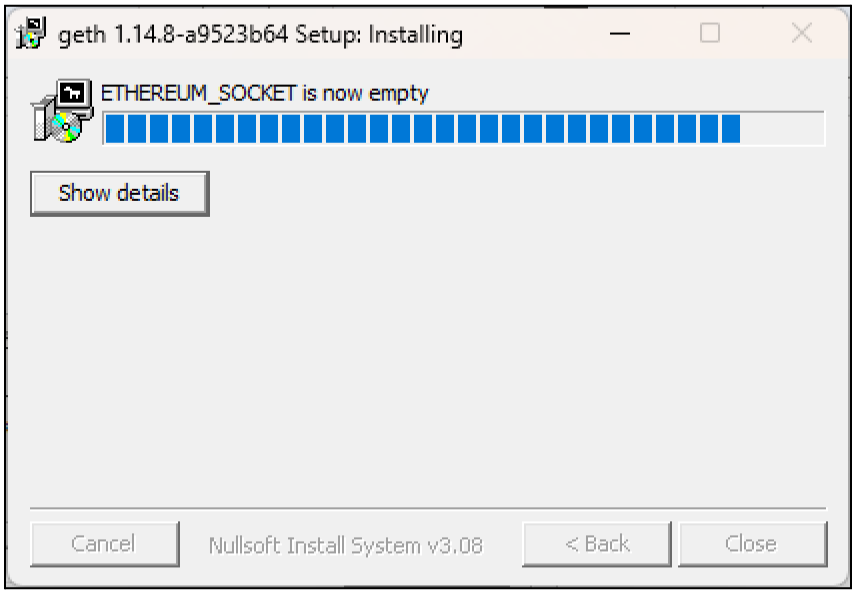
Setup will install geth 1.14.8-a9523b64 in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

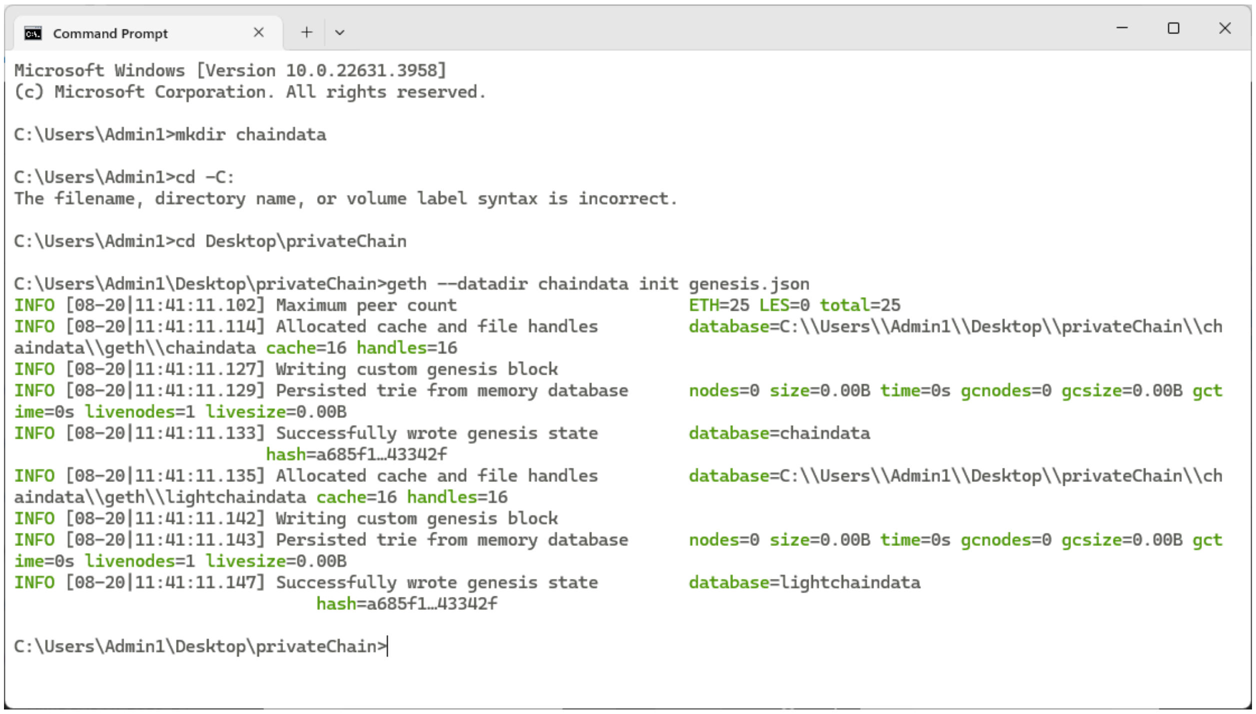
Destination Folder

C:\Program Files\Geth

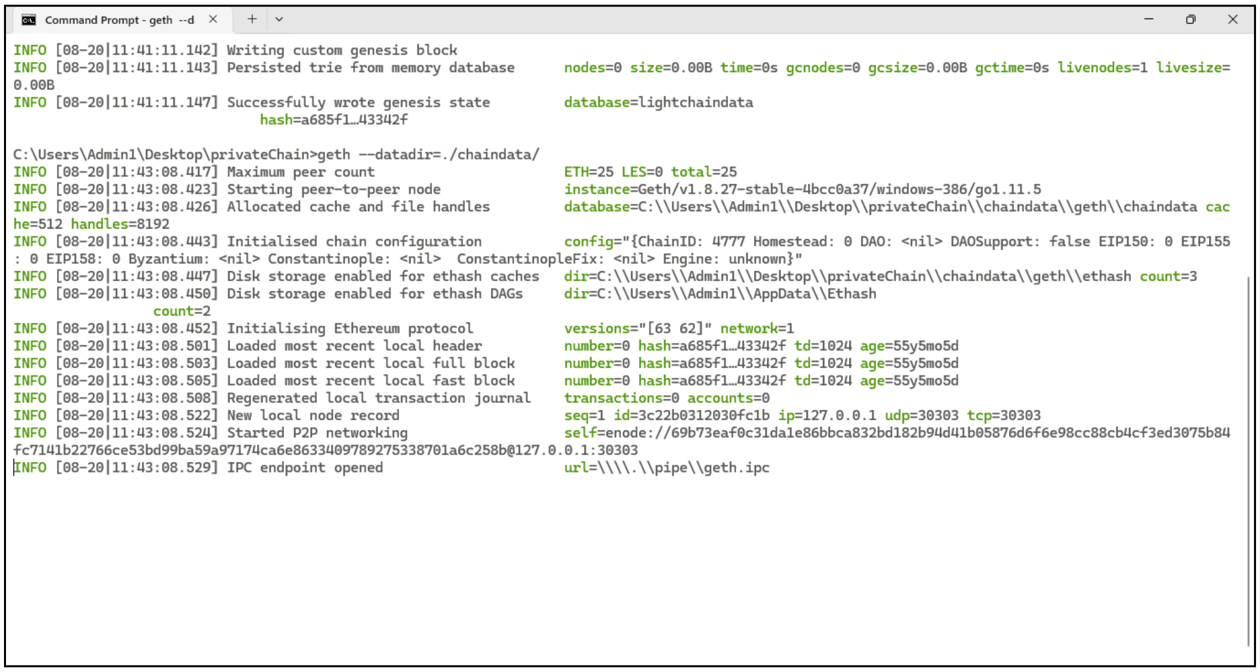
Space required: 101.3 MB

Space available: 24.8 GB


geth 1.14.8-a9523b64 Setup: Installing
ETHEREUM_SOCKET is now empty
Show details
Cancel
Nullsoft Install System v3.08
< Back
Close



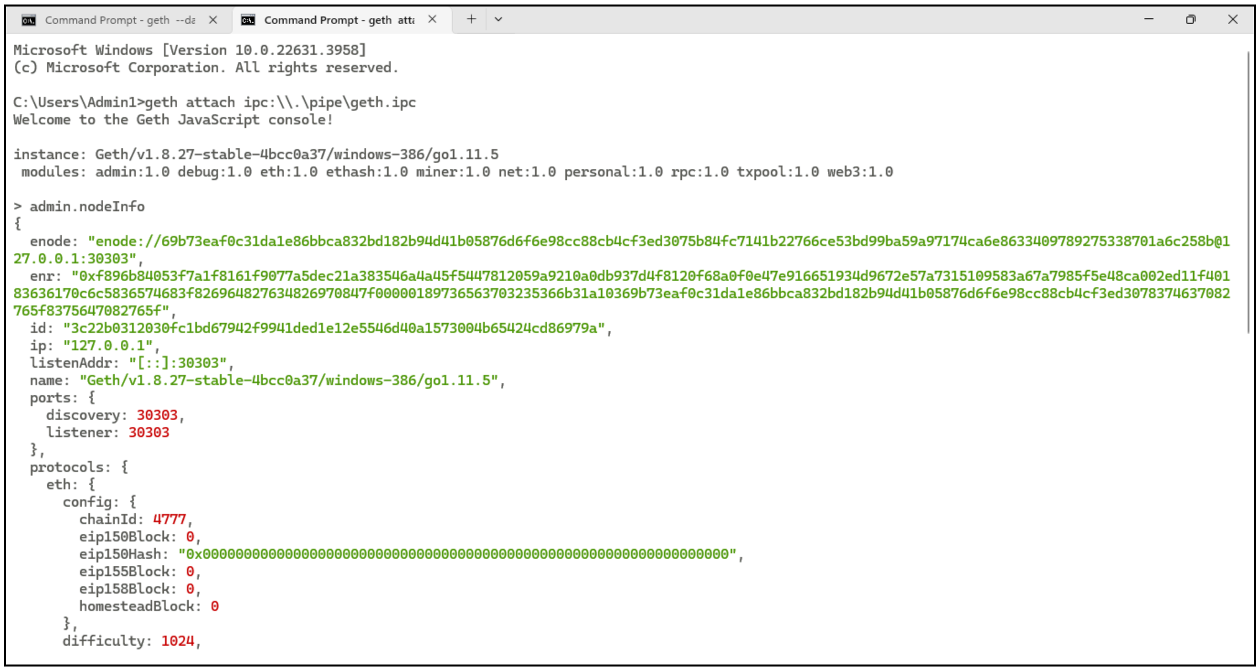
```
Microsoft Windows [Version 10.0.22631.3958]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin1>mkdir chaindata

C:\Users\Admin1>cd -C:
The filename, directory name, or volume label syntax is incorrect.

C:\Users\Admin1>cd Desktop\privateChain

C:\Users\Admin1\Desktop\privateChain>geth --datadir chaindata init genesis.json
INFO [08-20|11:41:11.102] Maximum peer count                       ETH=25 LES=0 total=25
INFO [08-20|11:41:11.114] Allocated cache and file handles         database=C:\\Users\\Admin1\\Desktop\\privateChain\\ch
aindata\\geth\\chaindata cache=16 handles=16
INFO [08-20|11:41:11.127] Writing custom genesis block
INFO [08-20|11:41:11.129] Persisted trie from memory database      nodes=0 size=0.00B time=0s gcnodes=0 gcsize=0.00B gct
ime=0s livenodes=1 livesize=0.00B
INFO [08-20|11:41:11.133] Successfully wrote genesis state         database=chaindata
                         hash=a685f1…43342f
INFO [08-20|11:41:11.135] Allocated cache and file handles         database=C:\\Users\\Admin1\\Desktop\\privateChain\\ch
aindata\\geth\\lightchaindata cache=16 handles=16
INFO [08-20|11:41:11.142] Writing custom genesis block
INFO [08-20|11:41:11.143] Persisted trie from memory database      nodes=0 size=0.00B time=0s gcnodes=0 gcsize=0.00B gct
ime=0s livenodes=1 livesize=0.00B
INFO [08-20|11:41:11.147] Successfully wrote genesis state         database=lightchaindata
                              hash=a685f1…43342f

C:\Users\Admin1\Desktop\privateChain>
```



Command Prompt - geth --d

```
INFO [08-20|11:41:11.142] Writing custom genesis block
INFO [08-20|11:41:11.143] Persisted trie from memory database      nodes=0 size=0.00B time=0s gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=
0.00B
INFO [08-20|11:41:11.147] Successfully wrote genesis state         database=lightchaindata
                              hash=a685f1…43342f

C:\Users\Admin1\Desktop\privateChain>geth --datadir=./chaindata/
INFO [08-20|11:43:08.417] Maximum peer count                       ETH=25 LES=0 total=25
INFO [08-20|11:43:08.423] Starting peer-to-peer node               instance=Geth/v1.8.27-stable-4bcc0a37/windows-386/go1.11.5
INFO [08-20|11:43:08.426] Allocated cache and file handles         database=C:\\Users\\Admin1\\Desktop\\privateChain\\chaindata\\geth\\chaindata cac
he=512 handles=8192
INFO [08-20|11:43:08.443] Initialised chain configuration          config="{ChainID: 4777 Homestead: 0 DAO: <nil> DAOSupport: false EIP150: 0 EIP155
: 0 EIP158: 0 Byzantium: <nil> Constantinople: <nil>  ConstantinopleFix: <nil> Engine: unknown}"
INFO [08-20|11:43:08.447] Disk storage enabled for ethash caches   dir=C:\\Users\\Admin1\\Desktop\\privateChain\\chaindata\\geth\\ethash count=3
INFO [08-20|11:43:08.450] Disk storage enabled for ethash DAGs     dir=C:\\Users\\Admin1\\AppData\\Ethash
                 count=2
INFO [08-20|11:43:08.452] Initialising Ethereum protocol           versions="[63 62]" network=1
INFO [08-20|11:43:08.501] Loaded most recent local header          number=0 hash=a685f1…43342f td=1024 age=55y5mo5d
INFO [08-20|11:43:08.503] Loaded most recent local full block      number=0 hash=a685f1…43342f td=1024 age=55y5mo5d
INFO [08-20|11:43:08.505] Loaded most recent local fast block      number=0 hash=a685f1…43342f td=1024 age=55y5mo5d
INFO [08-20|11:43:08.508] Regenerated local transaction journal    transactions=0 accounts=0
INFO [08-20|11:43:08.522] New local node record                    seq=1 id=3c22b0312030fc1b ip=127.0.0.1 udp=30303 tcp=30303
INFO [08-20|11:43:08.524] Started P2P networking                   self=enode://69b73eaf0c31da1e86bbca832bd182b94d41b05876d6f6e98cc88cb4cf3ed3075b84
fc7141b22766ce53bd99ba59a97174ca6e8633409789275338701a6c258b@127.0.0.1:30303
INFO [08-20|11:43:08.529] IPC endpoint opened                      url=\\\\.\\pipe\\geth.ipc
```

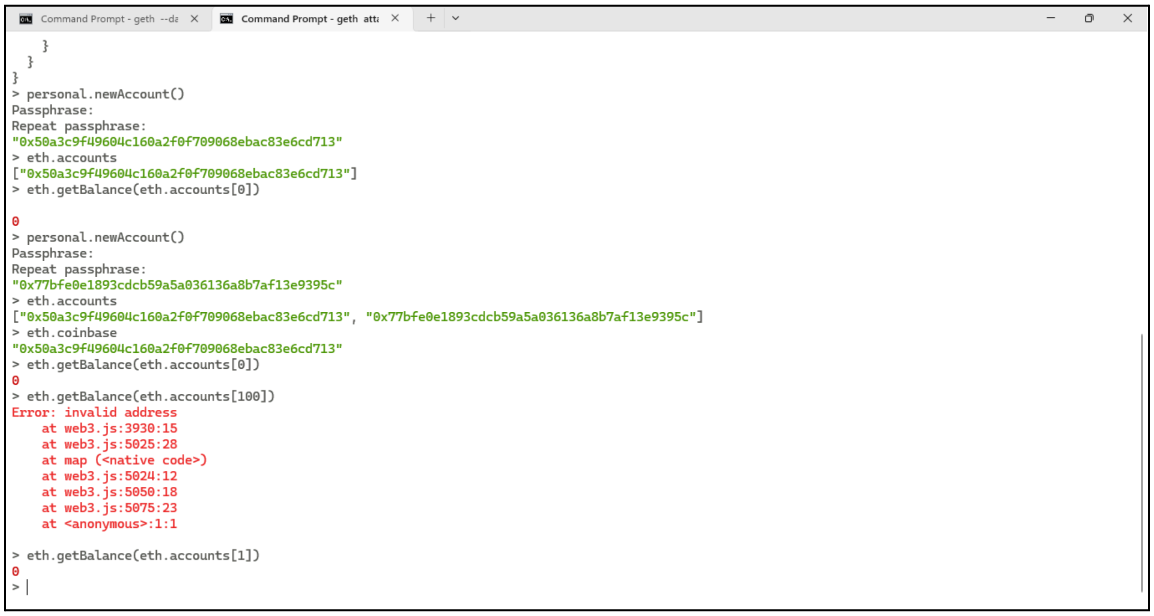
- **geth attach ipc:\\.\pipe\geth.ipc**



Command Prompt - geth --da | Command Prompt - geth atta

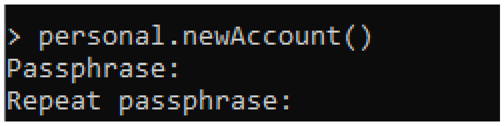
```
Microsoft Windows [Version 10.0.22631.3958]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin1>geth attach ipc:\\.\pipe\geth.ipc
Welcome to the Geth JavaScript console!

instance: Geth/v1.8.27-stable-4bcc0a37/windows-386/go1.11.5
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

> admin.nodeInfo
{
  enode: "enode://69b73eaf0c31da1e86bbca832bd182b94d41b05876d6f6e98cc88cb4cf3ed3075b84fc7141b22766ce53bd99ba59a97174ca6e8633409789275338701a6c258b@1
27.0.0.1:30303",
  enr: "0xf896b84053f7a1f8161f9077a5dec21a383546a4a45f5447812059a9210a0db937d4f8120f68a0f0e47e916651934d9672e57a7315109583a67a7985f5e48ca002ed11f401
83636170c6c5836574683f826964827634826970847f000001897365637032353666b31a10369b73eaf0c31da1e86bbca832bd182b94d41b05876d6f6e98cc88cb4cf3ed3078374637082
765f8375647082765f",
  id: "3c22b0312030fc1bd67942f9941ded1e12e5546d40a1573004b65424cd86979a",
  ip: "127.0.0.1",
  listenAddr: "[::]:30303",
  name: "Geth/v1.8.27-stable-4bcc0a37/windows-386/go1.11.5",
  ports: {
    discovery: 30303,
    listener: 30303
  },
  protocols: {
    eth: {
      config: {
        chainId: 4777,
        eip150Block: 0,
        eip150Hash: "0x0000000000000000000000000000000000000000000000000000000000000000",
        eip155Block: 0,
        eip158Block: 0,
        homesteadBlock: 0
      },
      difficulty: 1024,
```

- **personal.newAccount()**
- **eth.getBalance(eth.accounts[0])**



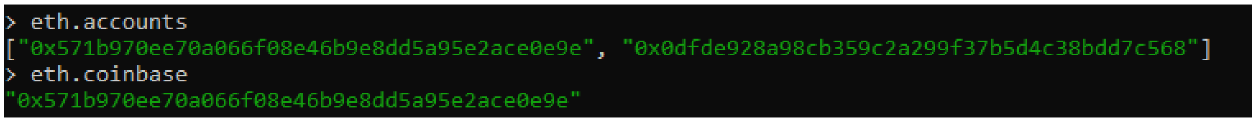
```
    }
  }
}
> personal.newAccount()
Passphrase:
Repeat passphrase:
"0x50a3c9f49604c160a2f0f709068ebac83e6cd713"
> eth.accounts
["0x50a3c9f49604c160a2f0f709068ebac83e6cd713"]
> eth.getBalance(eth.accounts[0])

0
> personal.newAccount()
Passphrase:
Repeat passphrase:
"0x77bfe0e1893cdcb59a5a036136a8b7af13e9395c"
> eth.accounts
["0x50a3c9f49604c160a2f0f709068ebac83e6cd713", "0x77bfe0e1893cdcb59a5a036136a8b7af13e9395c"]
> eth.coinbase
"0x50a3c9f49604c160a2f0f709068ebac83e6cd713"
> eth.getBalance(eth.accounts[0])
0
> eth.getBalance(eth.accounts[100])
Error: invalid address
    at web3.js:3930:15
    at web3.js:5025:28
    at map (<native code>)
    at web3.js:5024:12
    at web3.js:5050:18
    at web3.js:5075:23
    at <anonymous>:1:1

> eth.getBalance(eth.accounts[1])
0
> |
```

- **personal.newAccount():**



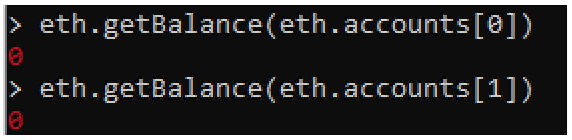
```
> personal.newAccount()
Passphrase:
Repeat passphrase:
```

- **eth.accounts**
- **eth.coinbase**



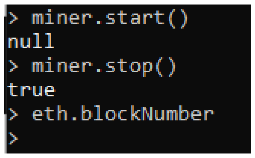
```
> eth.accounts
["0x571b970ee70a066f08e46b9e8dd5a95e2ace0e9e", "0x0dfde928a98cb359c2a299f37b5d4c38bdd7c568"]
> eth.coinbase
"0x571b970ee70a066f08e46b9e8dd5a95e2ace0e9e"
```

- **eth.getBalance(eth.accounts[0])**



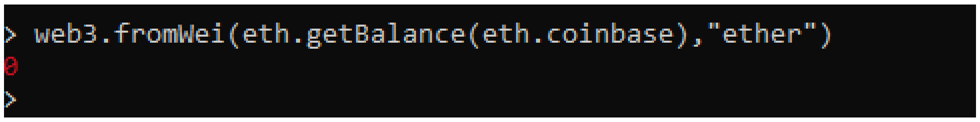
```
> eth.getBalance(eth.accounts[0])
0
> eth.getBalance(eth.accounts[1])
0
```

- **miner.start() / miner.stop()**



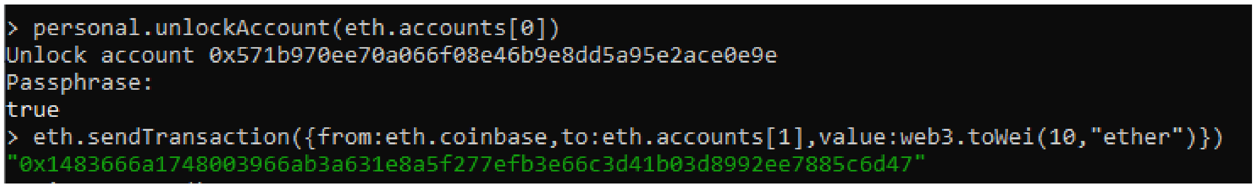
```
> miner.start()
null
> miner.stop()
true
> eth.blockNumber
>
```

- **web3.fromWei(eth.getBalance(eth.accounts[1]), “ether”)**

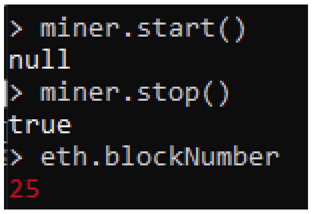


```
> web3.fromWei(eth.getBalance(eth.coinbase),"ether")
0
>
```

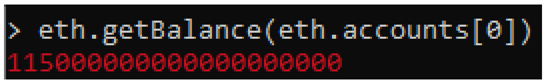
- **personal.unlockAccount(eth.accounts[0])**



```
> personal.unlockAccount(eth.accounts[0])
Unlock account 0x571b970ee70a066f08e46b9e8dd5a95e2ace0e9e
Passphrase:
true
> eth.sendTransaction({from:eth.coinbase,to:eth.accounts[1],value:web3.toWei(10,"ether")})
"0x1483666a1748003966ab3a631e8a5f277efb3e66c3d41b03d8992ee7885c6d47"
```

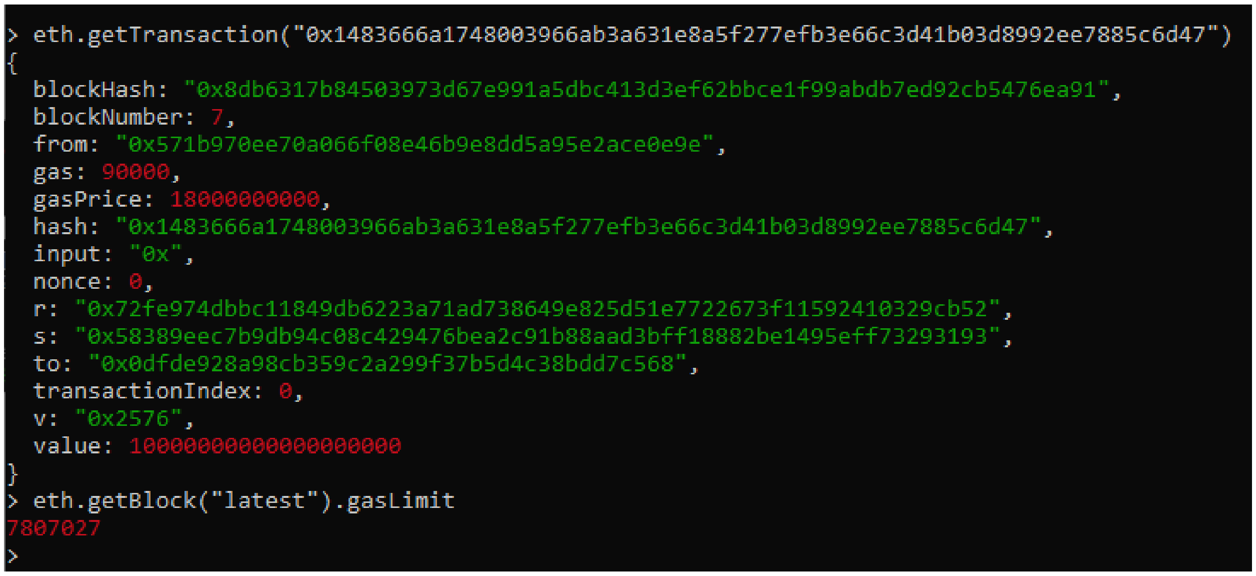


```
> miner.start()
null
> miner.stop()
true
> eth.blockNumber
25
```

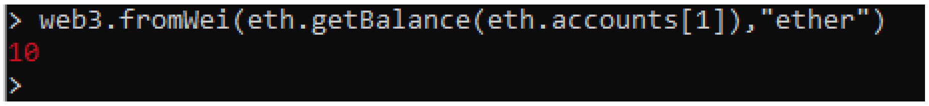


```
> eth.getBalance(eth.accounts[0])
115000000000000000000
```

- **eth.sendTransaction({from: eth.coinbase, to: eth.accounts[1], value: web3.toWei(10,“ether”)})**



```
> eth.getTransaction("0x1483666a1748003966ab3a631e8a5f277efb3e66c3d41b03d8992ee7885c6d47")
{
  blockHash: "0x8db6317b84503973d67e991a5dbc413d3ef62bbce1f99abdb7ed92cb5476ea91",
  blockNumber: 7,
  from: "0x571b970ee70a066f08e46b9e8dd5a95e2ace0e9e",
  gas: 90000,
  gasPrice: 18000000000,
  hash: "0x1483666a1748003966ab3a631e8a5f277efb3e66c3d41b03d8992ee7885c6d47",
  input: "0x",
  nonce: 0,
  r: "0x72fe974dbbc11849db6223a71ad738649e825d51e7722673f11592410329cb52",
  s: "0x58389eec7b9db94c08c429476bea2c91b88aad3bff18882be1495eff73293193",
  to: "0x0dfde928a98cb359c2a299f37b5d4c38bdd7c568",
  transactionIndex: 0,
  v: "0x2576",
  value: 10000000000000000000
}
> eth.getBlock("latest").gasLimit
7807027
>
```

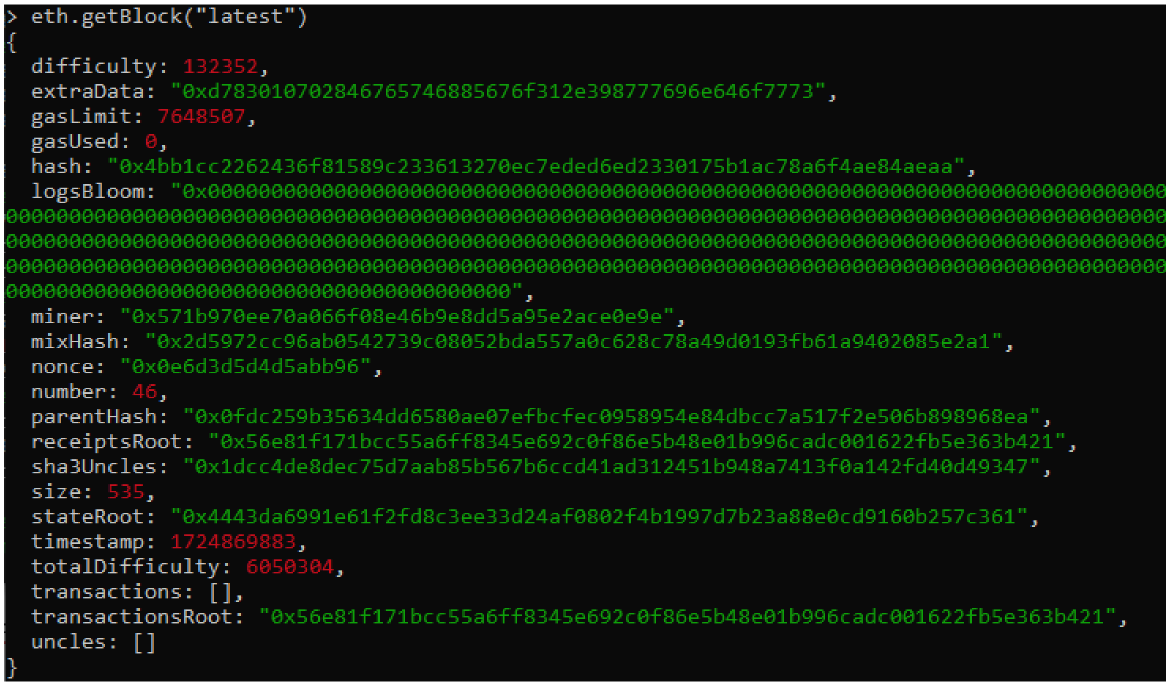


```
> web3.fromWei(eth.getBalance(eth.accounts[1]),"ether")
10
>
```

- **GET LATEST BLOCK:**

  Command:

  Eth.getBlock("latest")

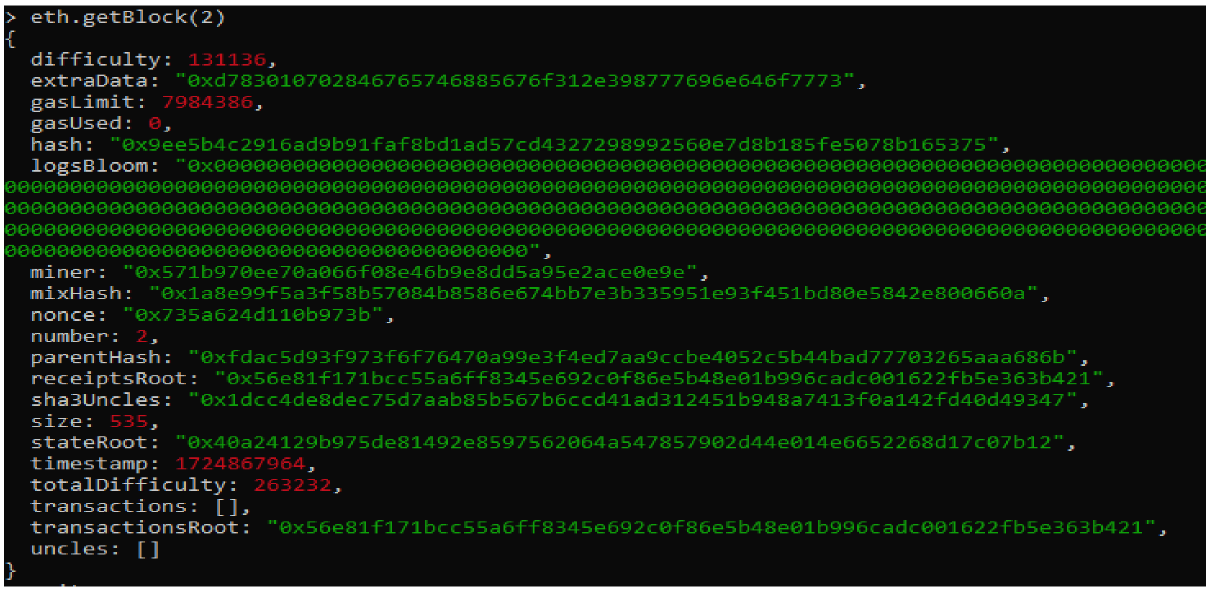


```
> eth.getBlock("latest")
{
  difficulty: 132352,
  extraData: "0xd783010702846765746885676f312e398777696e646f7773",
  gasLimit: 7648507,
  gasUsed: 0,
  hash: "0x4bb1cc2262436f81589c233613270ec7eded6ed2330175b1ac78a6f4ae84aeaa",
  logsBloom: "0x0000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000",
  miner: "0x571b970ee70a066f08e46b9e8dd5a95e2ace0e9e",
  mixHash: "0x2d5972cc96ab0542739c08052bda557a0c628c78a49d0193fb61a9402085e2a1",
  nonce: "0x0e6d3d5d4d5abb96",
  number: 46,
  parentHash: "0x0fdc259b35634dd6580ae07efbcfec0958954e84dbcc7a517f2e506b898968ea",
  receiptsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",
  sha3Uncles: "0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
  size: 535,
  stateRoot: "0x4443da6991e61f2fd8c3ee33d24af0802f4b1997d7b23a88e0cd9160b257c361",
  timestamp: 1724869883,
  totalDifficulty: 6050304,
  transactions: [],
  transactionsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",
  uncles: []
}
```

- **GET SPECFIC BLOCK INFO:**

  Command:

  Eth.getBlock(2)



```
> eth.getBlock(2)
{
  difficulty: 131136,
  extraData: "0xd783010702846765746885676f312e398777696e646f7773",
  gasLimit: 7984386,
  gasUsed: 0,
  hash: "0x9ee5b4c2916ad9b91faf8bd1ad57cd4327298992560e7d8b185fe5078b165375",
  logsBloom: "0x0000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000",
  miner: "0x571b970ee70a066f08e46b9e8dd5a95e2ace0e9e",
  mixHash: "0x1a8e99f5a3f58b57084b8586e674bb7e3b335951e93f451bd80e5842e800660a",
  nonce: "0x735a624d110b973b",
  number: 2,
  parentHash: "0xfdac5d93f973f6f76470a99e3f4ed7aa9ccbe4052c5b44bad77703265aaa686b",
  receiptsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",
  sha3Uncles: "0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
  size: 535,
  stateRoot: "0x40a24129b975de81492e8597562064a547857902d44e014e6652268d17c07b12",
  timestamp: 1724867964,
  totalDifficulty: 263232,
  transactions: [],
  transactionsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",
  uncles: []
}
```

Conclusion:

Setting up a private Ethereum network involves installing Geth, creating a genesis block, and initializing the blockchain. This setup allows for mining, sending transactions, and deploying smart contracts in a controlled environment, offering a hands-on way to explore Ethereum's features without using the main network.