**Practical No: 05**

**Aim: Install Geth Ethereum Node (Show installation steps also). Create a Genesis block and a private chain. Use geth commands to create user accounts, mine, transact etc.**

**Theory:**

* **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

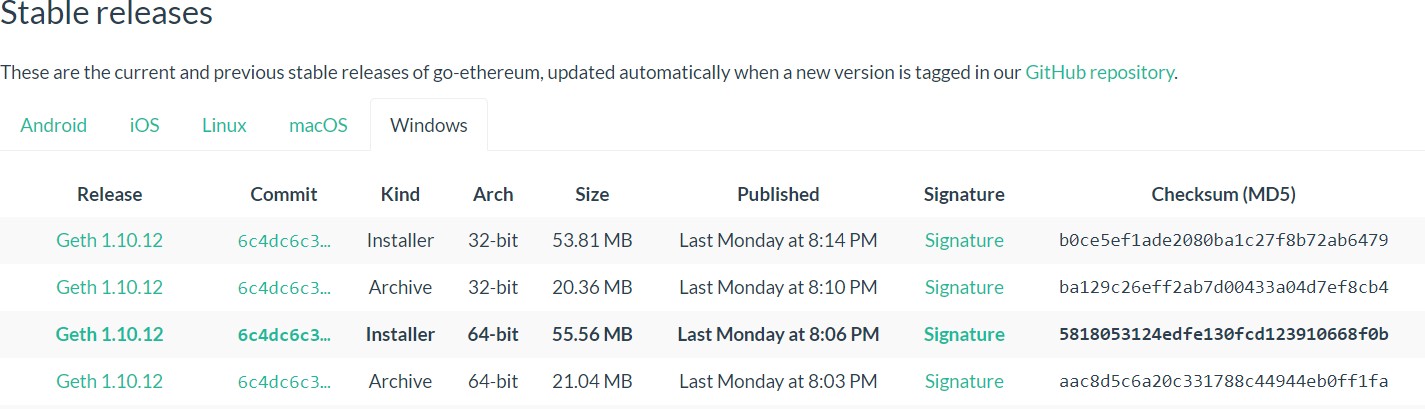
implemented in Go Language. Using Geth you can join Ethereum network, transfer ether between accounts or even mine ethers.

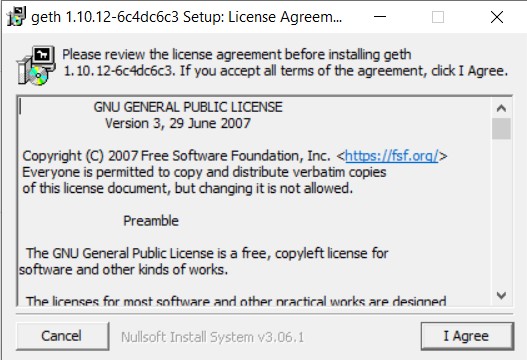
* **Genesis Block:**
* A genesis block refers to the first block in a blockchain and is usually hardcoded into its application’s software. A blockchain has multiple “blocks” (containing validated transactions and recorded activity data) linked together by a metaphorical chain.
* Each “block” of a crypto 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).

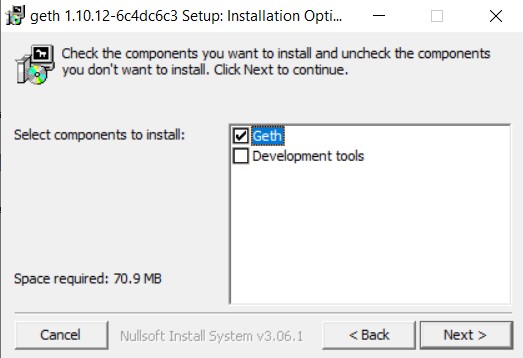
* **Commands used in this practical:**
* **geth --datadir chaindata init genesis.json:** Initializes geth into chaindata.
* **geth --datadir=./chaindata/:** Used to run geth.
* **geth attach ipc:\\.\pipe\geth.ipc**: IPC to interact with geth.
* **personal.newAccount():** Create a new account.
* **eth.accounts:** 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.
* **Steps to Install Go-Ethereum:**

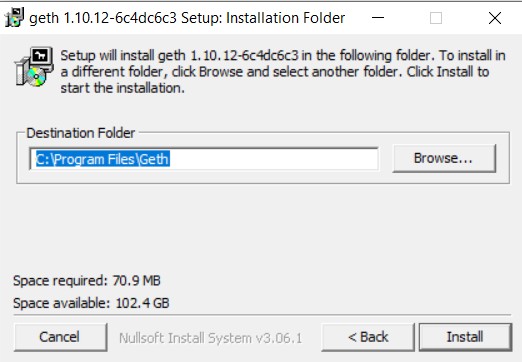
1. Visit the Go Ethereum website and install Geth.
2. Visit here: <https://geth.ethereum.org/downloads/>

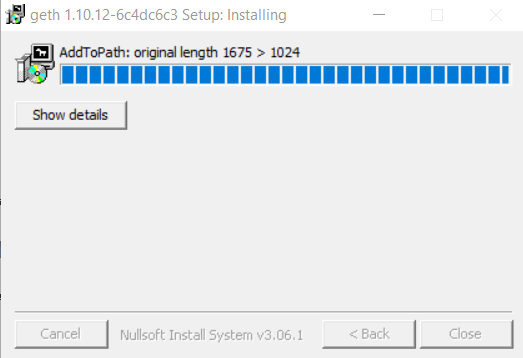
* **Download the latest release of Geth for Windows, make sure you download the 64-bit version.**
* **Once your download is complete, open the installer and click “I Agree”**



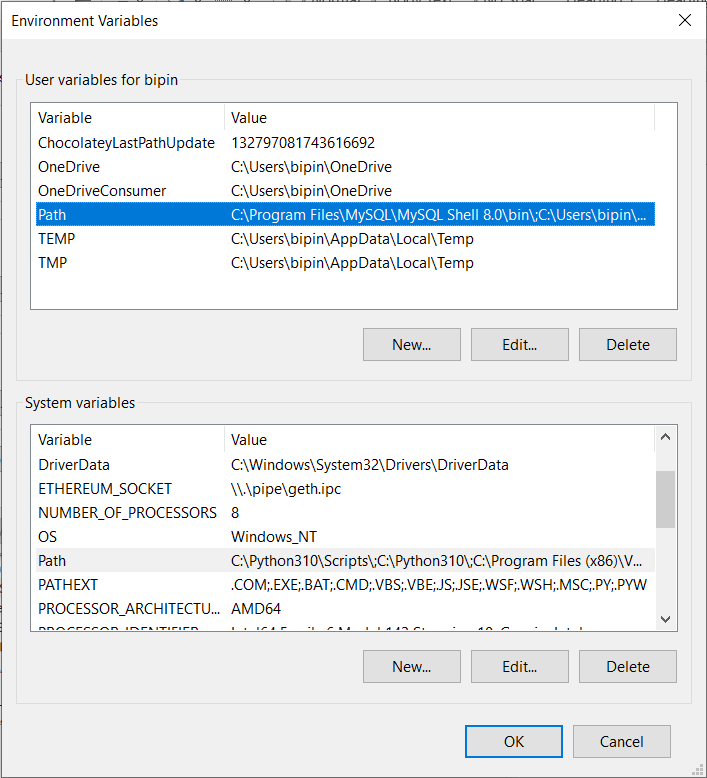
* **Follow all the installation steps.**

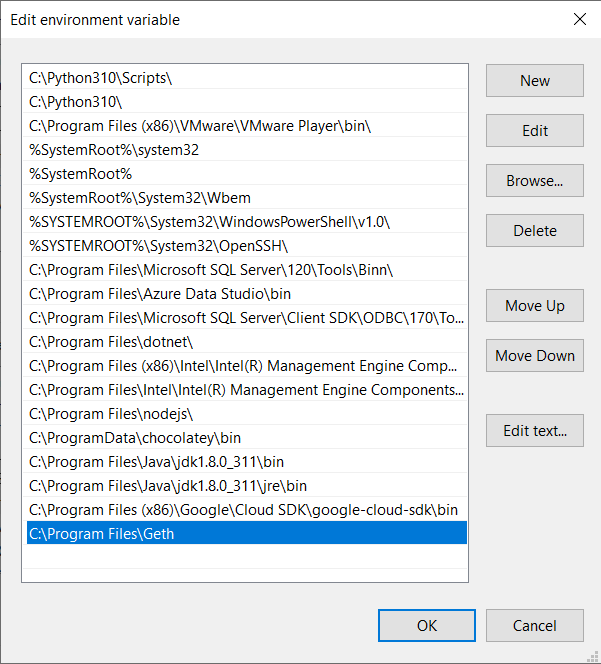






* **Check if you get this error after installation. (Path not updated, org length 1675>1024)**
* **If you do, then go to the installation path of geth and copy the path of the folder. Then open up environment variables and add the path to the “Path” section there.**





* Click OK. The installation of Geth is now done.
  + Steps to create a Genesis block and a private chain:
  + Create a new folder on your desktop called “Private Chain”.
  + Open command prompt in this folder and create a data directory folder for our chaindata by typing “mkdir chaindata”.
  + Next, we need to create and save our genesis.json block in our Private Chain folder, as the genesis block will be used to initialize our private network and store data in the data directory folder “chaindata”.
  + Open up notepad, copy & paste the code below into a new file called “genesis.json” and save this file in our Private Chain folder.

{

"config": { "chainId": 4777,

"homesteadBlock": 0,

"eip150Block": 0,

"eip155Block": 0, "eip158Block": 0

},

"alloc" : {}, "difficulty" : "0x400", "extraData" : "",

"gasLimit" : "0x7A1200",

"parentHash" : "0x0000000000000000000000000000000000000000000000000000000

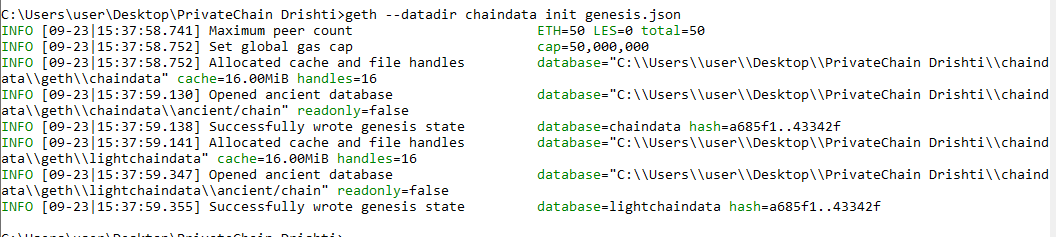
000000000", "timestamp" : "0x00"

}

**Now, we have to initialize our Ethereum Node. For that, open the command prompt and change your current directory to the Private Chain folder.**

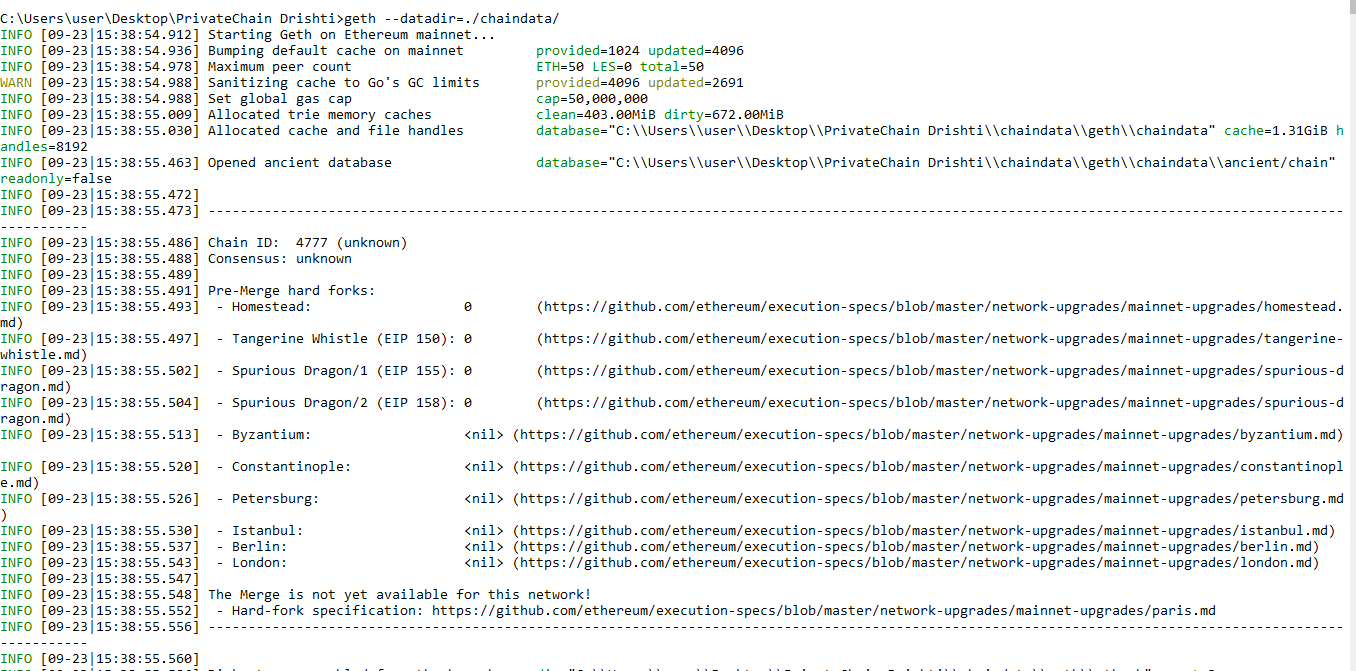
1. **Intialize geth into chaindata**

**Command: geth --datadir chaindata init genesis.json**



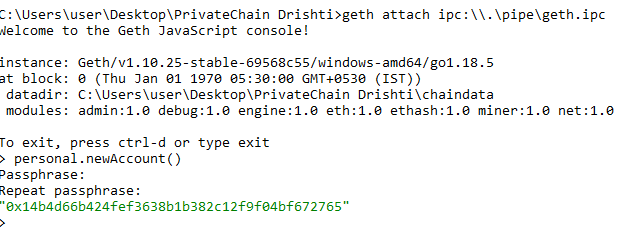
1. **run geth:**

**Command:** geth --datadir=./chaindata/

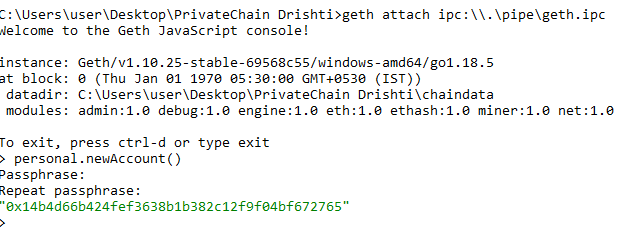


1. **IPC to interact with Geth:**

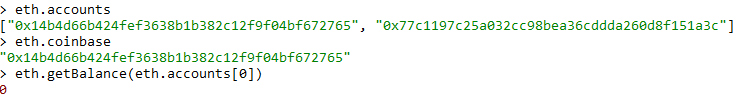
**Command: geth attach ipc:\\.\pipe\geth.ipc**



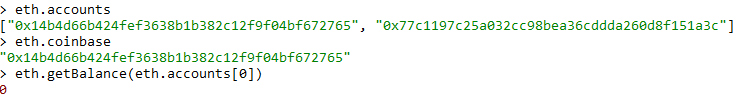
**Command: personal.newAccount()**



**Command:** **Eth.accounts //hexadecimal address, will show the accounts**



**Command:** **Eth.coinbase //first account is called as the coinbase account**



**Command:**

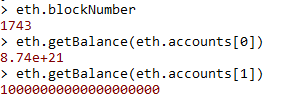
**eth.getBalance(eth.accounts[0]) //2nd account will have 0 ethers, first account will have some amount which we shall use to give to other accounts**

**miner.start()**

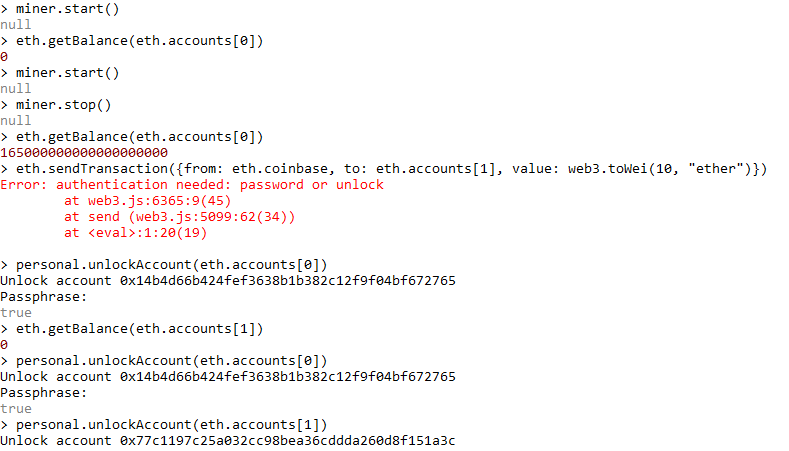
**miner.stop()**

**//After a transaction, stop to get some amount in other accounts**

**eth.blockNumber**



**Command:** **personal.unlockAccount(eth.accounts[0])**



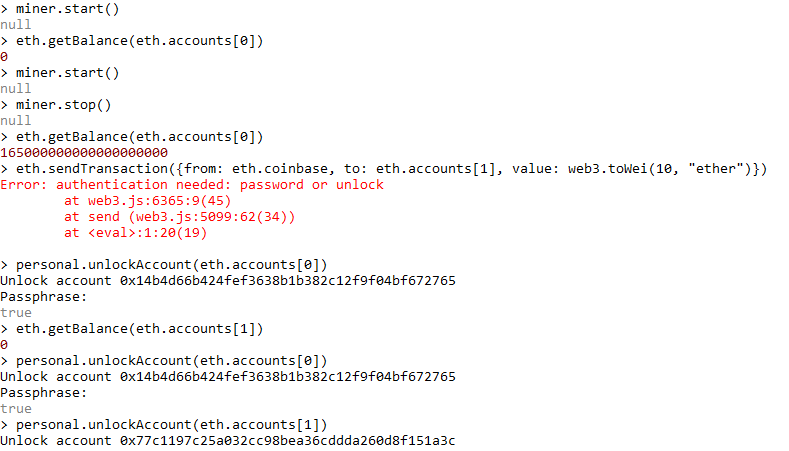
1. **Transaction:**

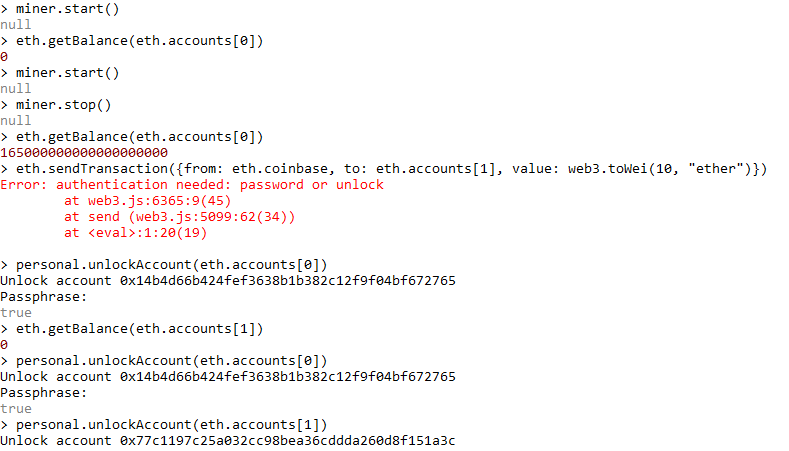
**Command:**

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

**miner.start()**

**Miner.stop()**





1. **Copy transaction hash and paste it in txHash**

**Transaction ID: 0x590216e5dd096293b866cb6c67d3be709ad814fad104792c4186c21a03ca29cb"**

**Command:**

**eth.getTransaction("0x590216e5dd096293b866cb6c67d3be709ad814fad104792c4186c21a03ca29cb")**



1. **Balance of second account**

**Command:**

**web3.fromWei(eth.getBalance(eth.accounts[1]), "ether")**

**web3.fromWei(eth.getBalance(eth.accounts[0]), "ether")**

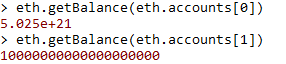




**Command:**

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

**eth.getBalance(eth.accounts[1])**



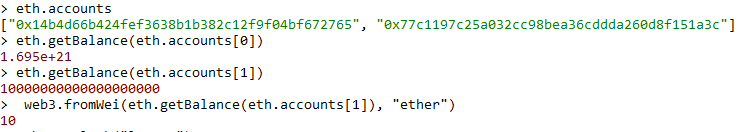
**Command: eth.getBlock("latest")**



**Command: eth.getBlock(410)**



**Command: personal.unlockAccount(eth.accounts[1])**



**Conclusion:** We have successfully learnt how to install Geth, create a genesis block, private chain and commands to create user accounts, mine and transact ethers among accounts.