# Department of Computer Engineering

**Academic Term: July-November 2023**

**Class :** B.E Computers-A Sem VII **Subject:** Blockchain Technology Lab **Subject Code:** CSDL7022

|  |  |
| --- | --- |
| **Practical No:** | 6 |
| **Title:** | Blockchain platform ethereum using Geth |
| **Date of Performance:** |  |
| **Date of Submission:** |  |
| **Roll No:** | 9427 |
| **Name of the Student:** | Atharva Prashant Pawar |

# Evaluation:

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Rubrics** | **Grades** |
| 1 | Time Line (2) |  |
| 2 | Output (3) |  |
| 3 | Code optimization (2) |  |
| 4 | Post lab (3) |  |

**Signature of the Teacher :**

# Experiment No. 6

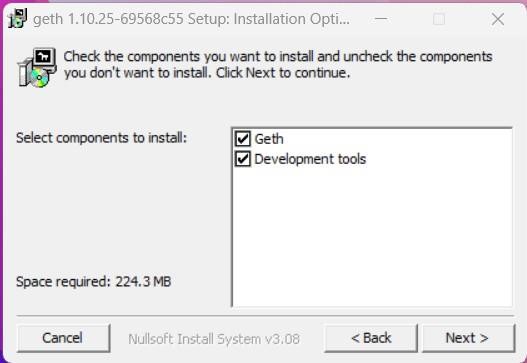
**Blockchain platform ethereum using Geth.**

**Aim:** Study of Blockchain platform ethereum using Geth.

# Theory:

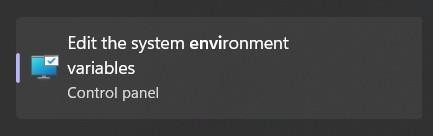
**Step:1**

Following link is used to download Geth 1.10.25 for Windows. <https://geth.ethereum.org/downloads/>

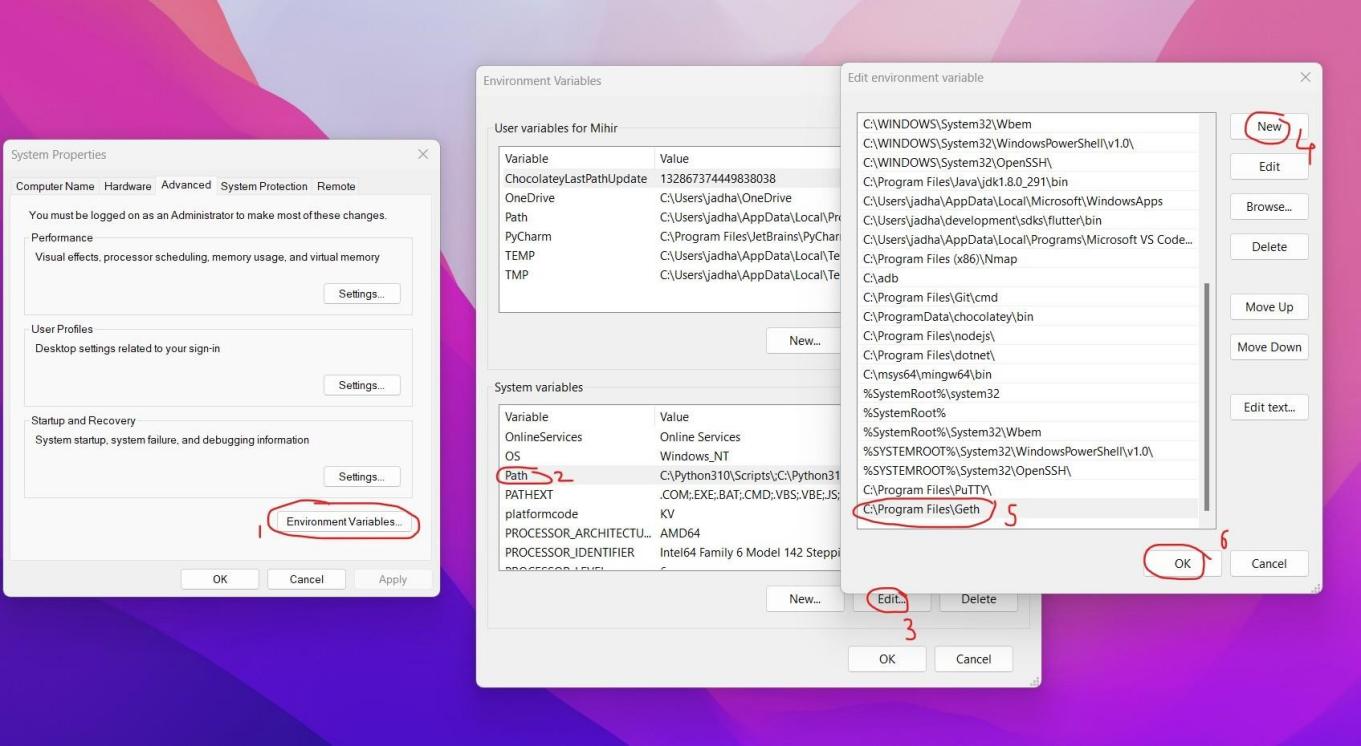


After installation add the Geth to the path

Search Environment Variables in Windows and Open it



After opening, add the Geth location i.e., C:\Program Files\Geth to the path in the Environment Variables



OR

Create folder in other drive say D drive and install Geth in that drive in folder Geth.

Following are the steps to be followed.

## Steps:

1. create a folder
2. create custom genesis file
3. create custom data directory
4. set custom networkID (ChainID)

# Step:2

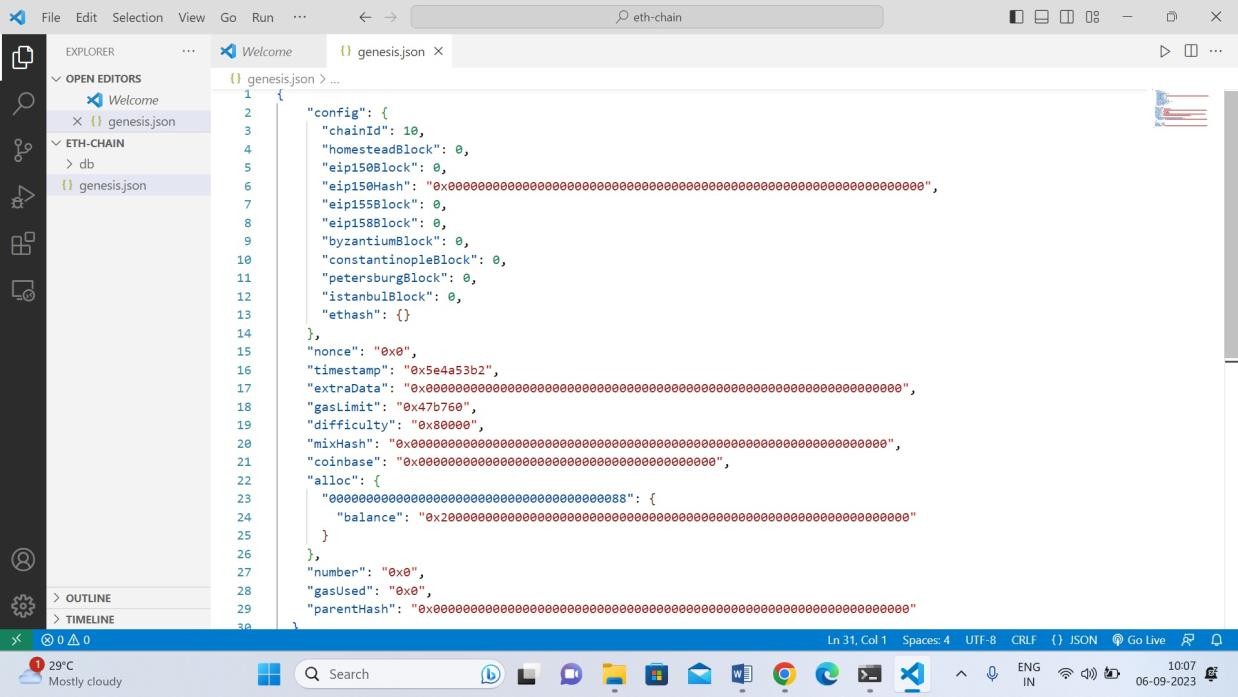
## Create Genesis File

The Genesis block is the start block of the Blockchain — the first block, block 0, and the only block that does not point to a predecessor block. the genesis block is hard coded into clients, but in Ethereum it can be whatever you like. The Genesis file is a JSON file that defines the characteristics of that initial block and subsequently the rest of the blockchain.

* 1. Create a directory to hold your network files.

mkdir eth-chain cd eth-chain

* 1. Create your genesis file

Open notepad and save file in eth-chain folder with name genesis.json. Copy following code in this Genesis.json file.

## config

chainId — this is your chain’s identifier, and is used in replay protection.

homesteadBlock, eip155Block, eip158Block, byzantiumBlock — these relate to chain forking and versioning, so in our case lets leave them 0 since we’re starting a new blockchain.

## difficulty

This dictates how difficult it is to mine a block.

## gasLimit

This is the the total amount of gas that can be used in each block. With such a low mining difficulty, blocks will be moving pretty quick, but you should still set this value pretty high to avoid hitting the limit and slowing down your network.

## alloc

Here you can allocate ETH to specific addresses. This won’t create the account for you, so make sure its an account you already have control of. You will need to add the account to your private chain in order to use it, and to do that you need access to the keystore/utc file.

## Nonce

A scalar value equal to the number of transactions sent by the sender.

## mixhash

mixhash is an intermediary calculation to finding the nonce that is not as costly to determine.

## coinbase

The ether rewards gained from “mining” the genesis block go to the 160-bit coinbaseaddress. timestamp

The output of the Unix time() function when the block was created

## parentHash

The Keccak 256-bit hash of the previous block’s header. This is meaningless in the genesis block, since block 0 has no parent.

## extraData

An optional free, but max. 32 byte long space to conserve smart things for ethernity on the Blockchain.

# Step:3

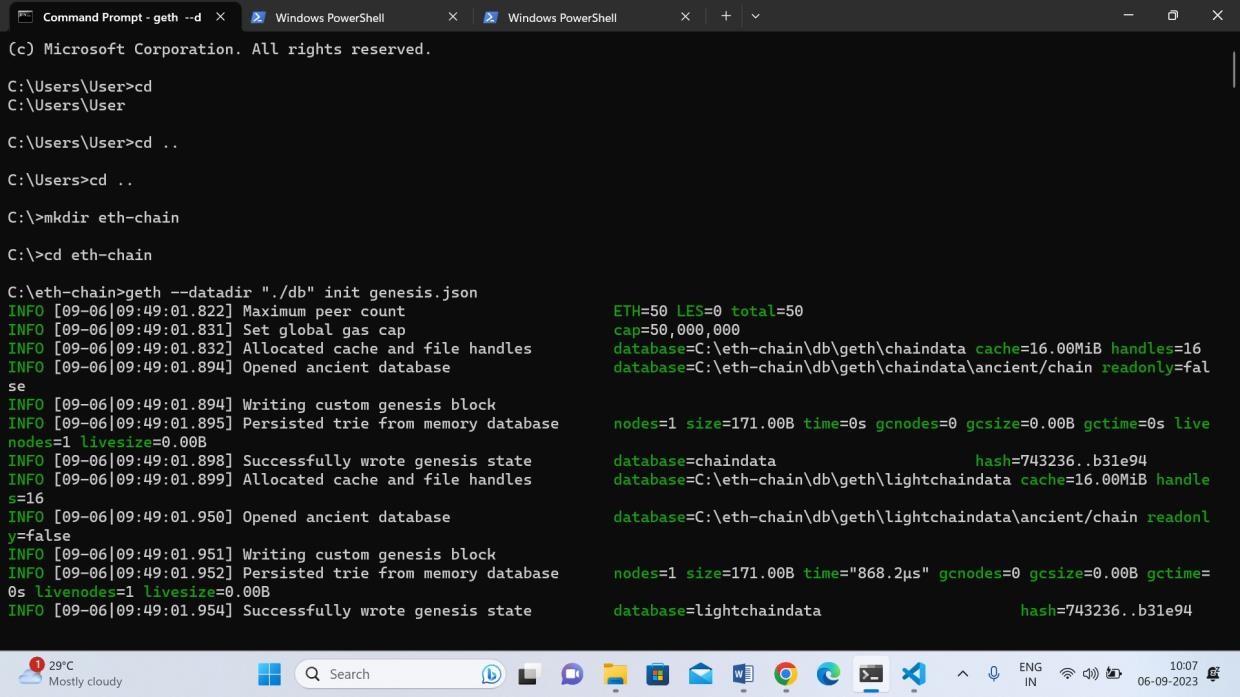
## Initial the genesis block

Init our blockchain with the settings in the genesis file and define a folder for storing chain data.

* geth --datadir "./db" init genesis.json

datadir : Data directory for the databases and keystore init： initialize a new genesis block

We get the following output’



## Data Directory

Everything geth persists gets written inside its data directory (except for the PoW Ethash DAG. The default data directory locations in Window platform is %APPDATA%\Ethereum.

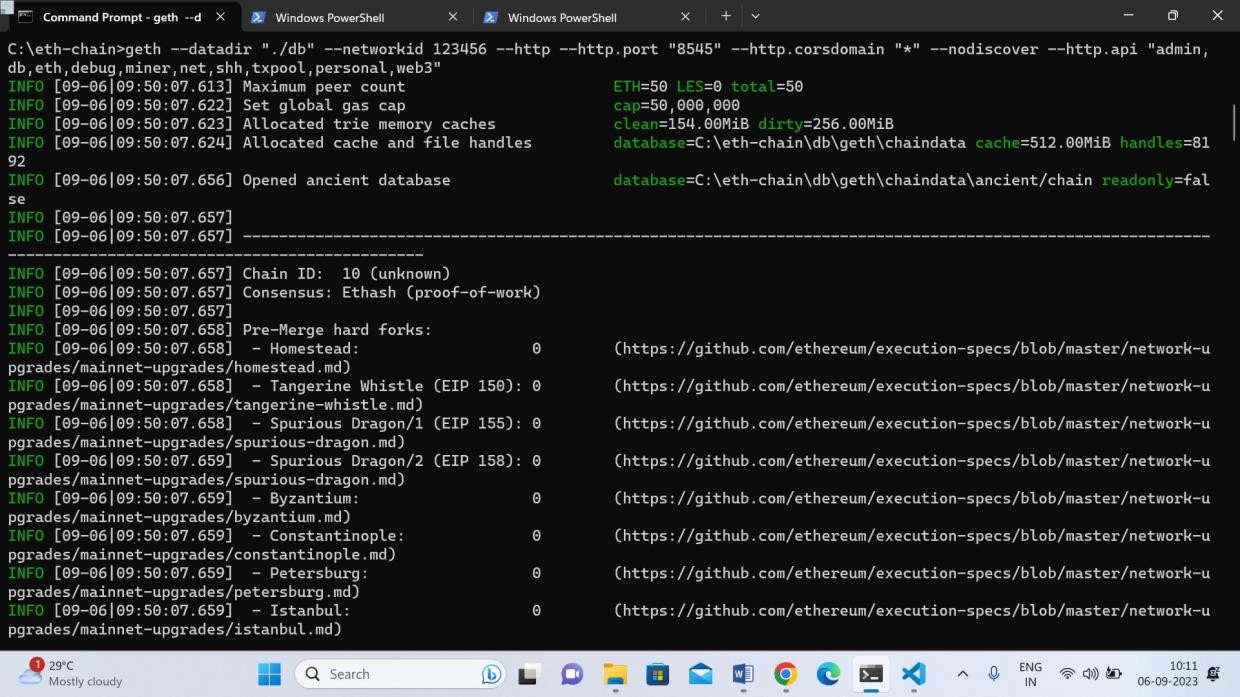
# Step:4

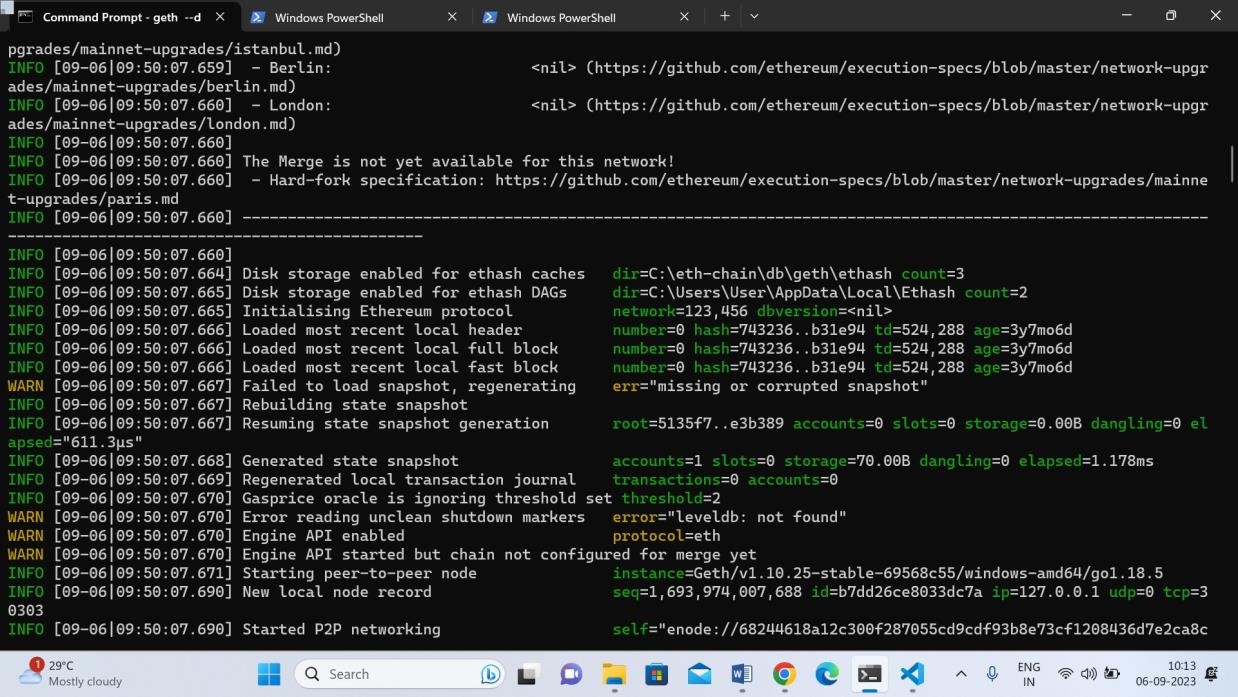
## Start your Ethereum peer node

Networkid helps ensure the privacy of your network. You can use any number here (where we used “ 123456”), but other peers joining your network must use the same one. Use following command.

geth --datadir "./db" --networkid 123456 --http –http.port "8545" –http.corsdomain "\*"

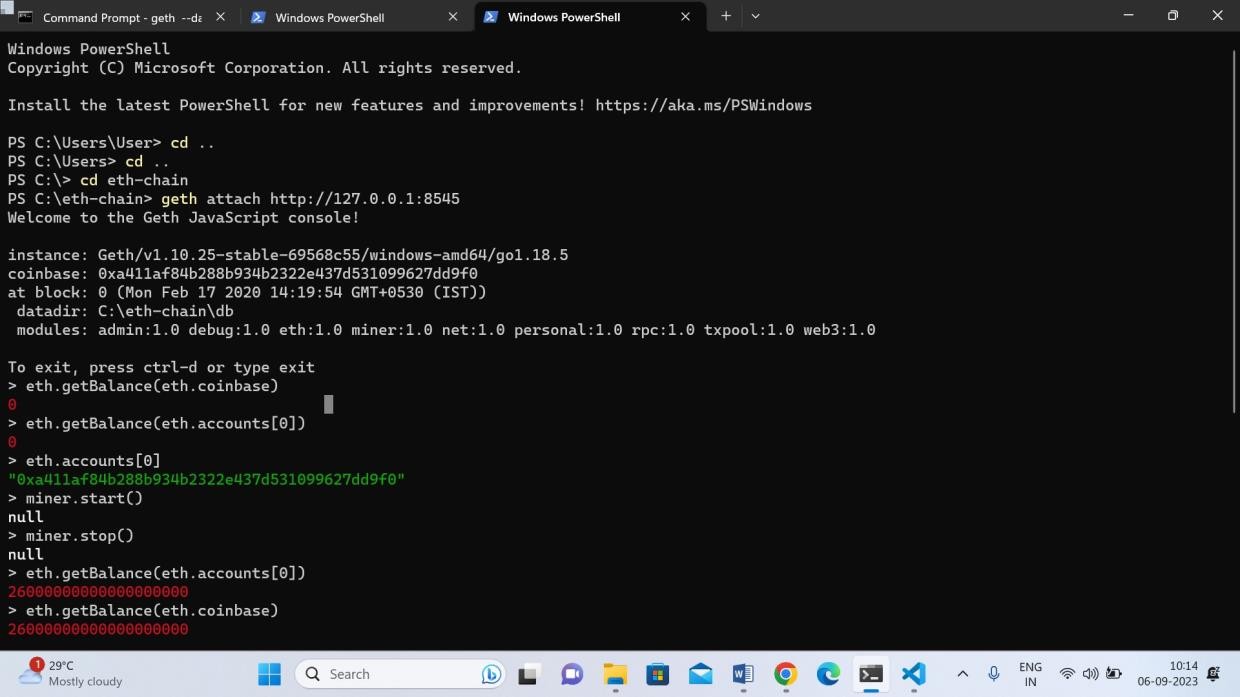
--nodiscover –http.api=”admin,db,eth,debug,miner,net,shh,txpool,personal,web3”





# Step:5

Open a new command prompt. Use following command. geth attach http://127.0.0.1:8545



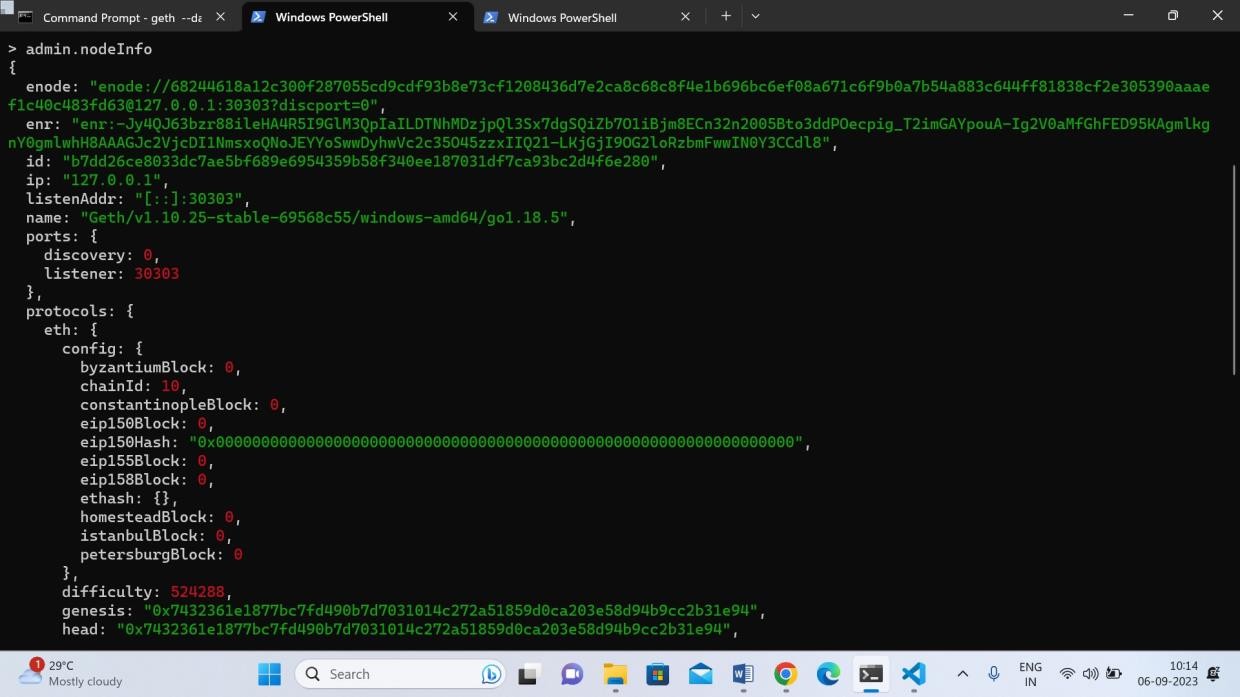
# Step:6

## Node Information

The nodeInfo administrative property can be queried for all the information known about the running Geth node at the networking granularity.

Type the below in the new terminal(command prompt)

* admin.nodeInfo



# Step:7

## Creating a new account

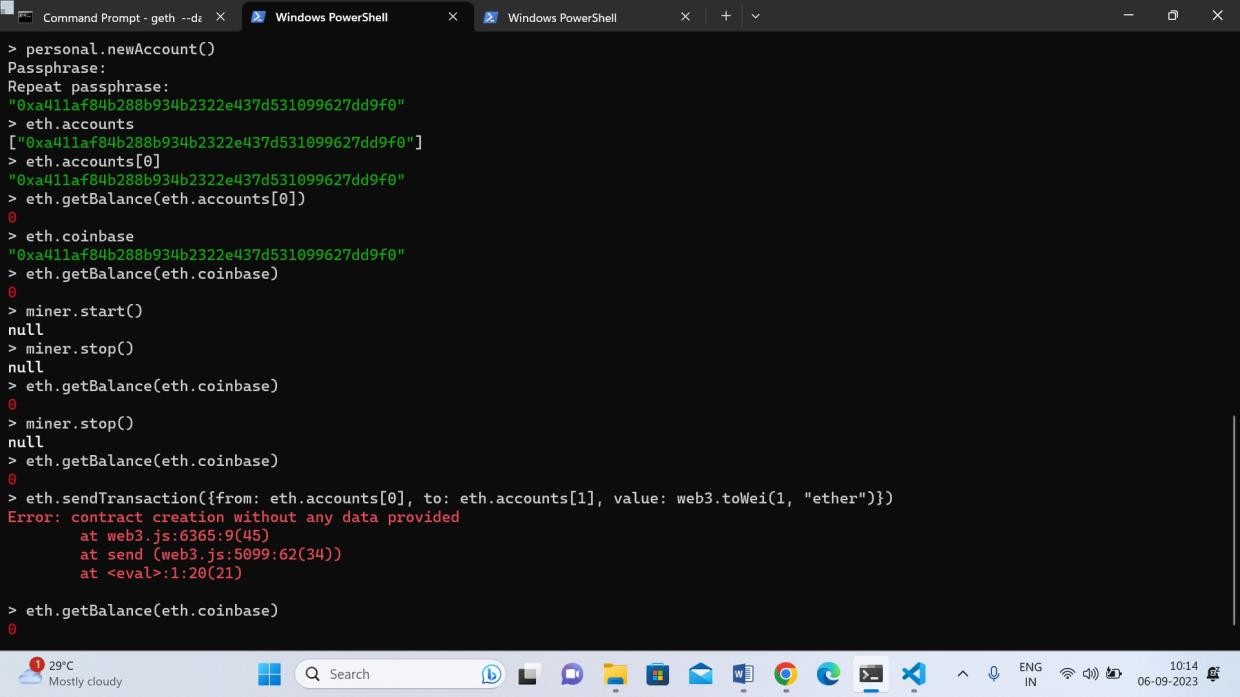
* personal.newAccount()

Creates a new account and prints the address. On the console, use:

Enter your password and then it will display account address. In my case, I set password to “123456”

Second way to create a new account

* personal.newAccount("123456")

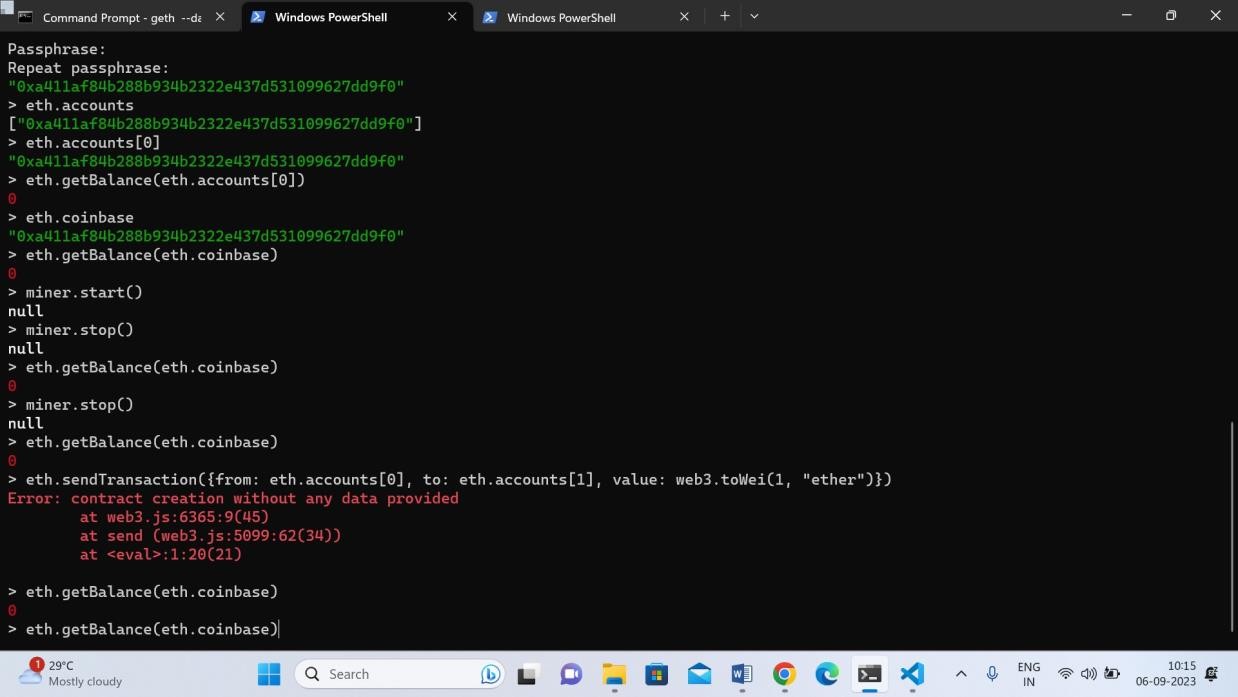


# Step:8

## Check Accounts

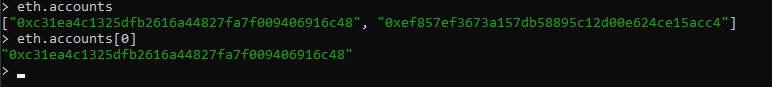
**Use following command to check account**

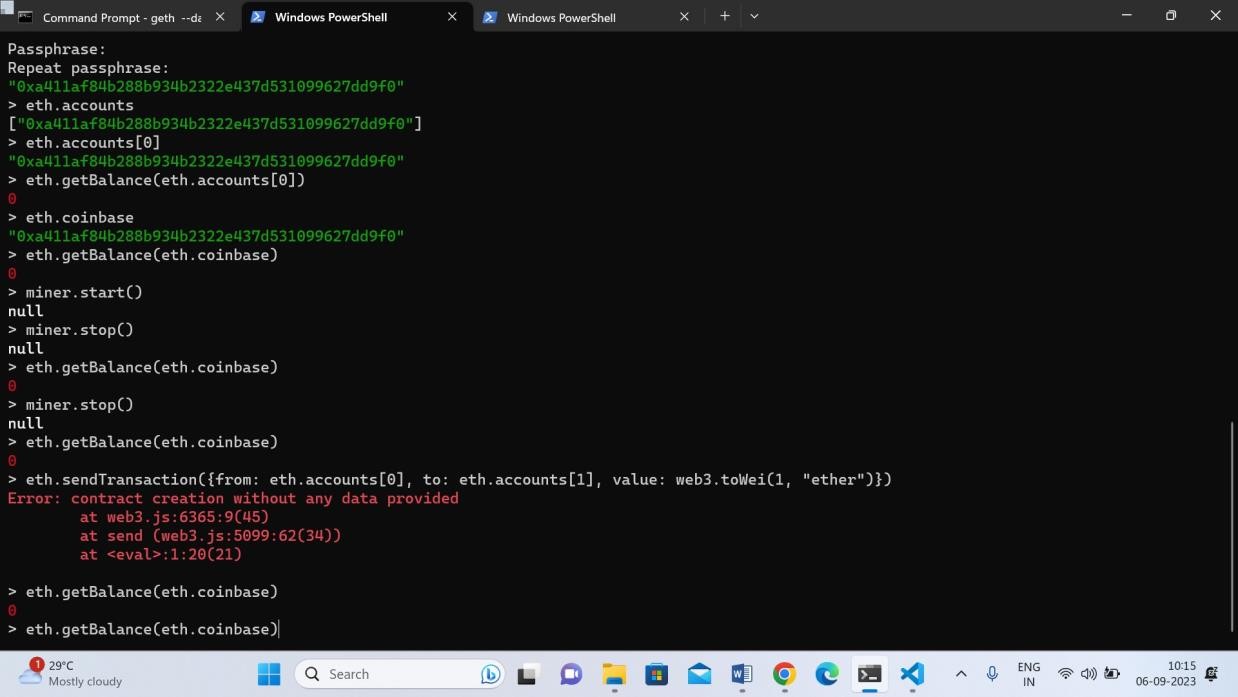
* eth.accounts



Accounts is an array so you can search account by index also.

* eth.accounts[0]



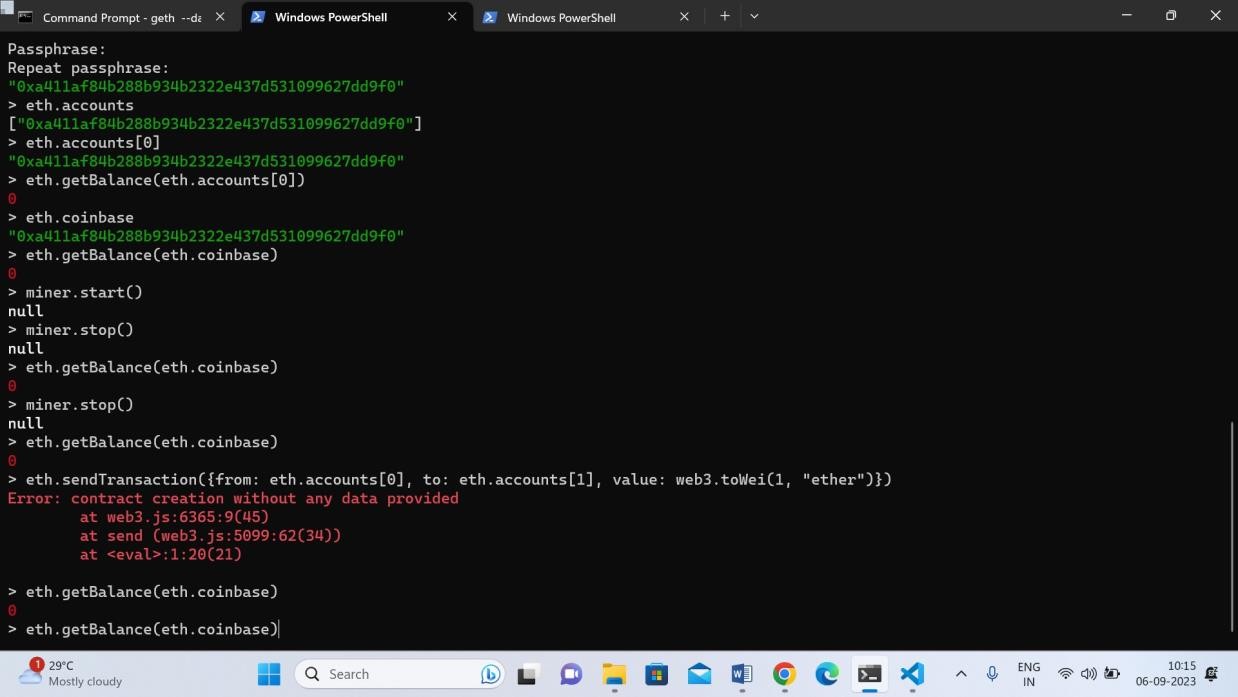


# Step:9

## Check balance of account

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

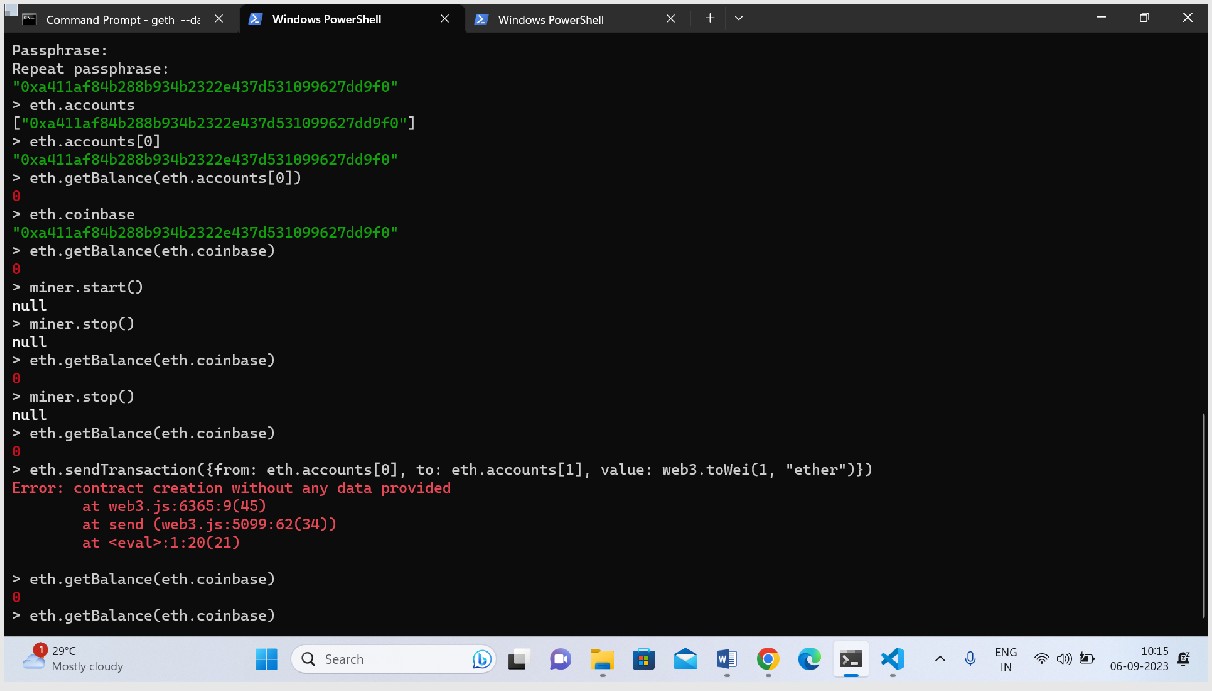




# Step:10 Mining

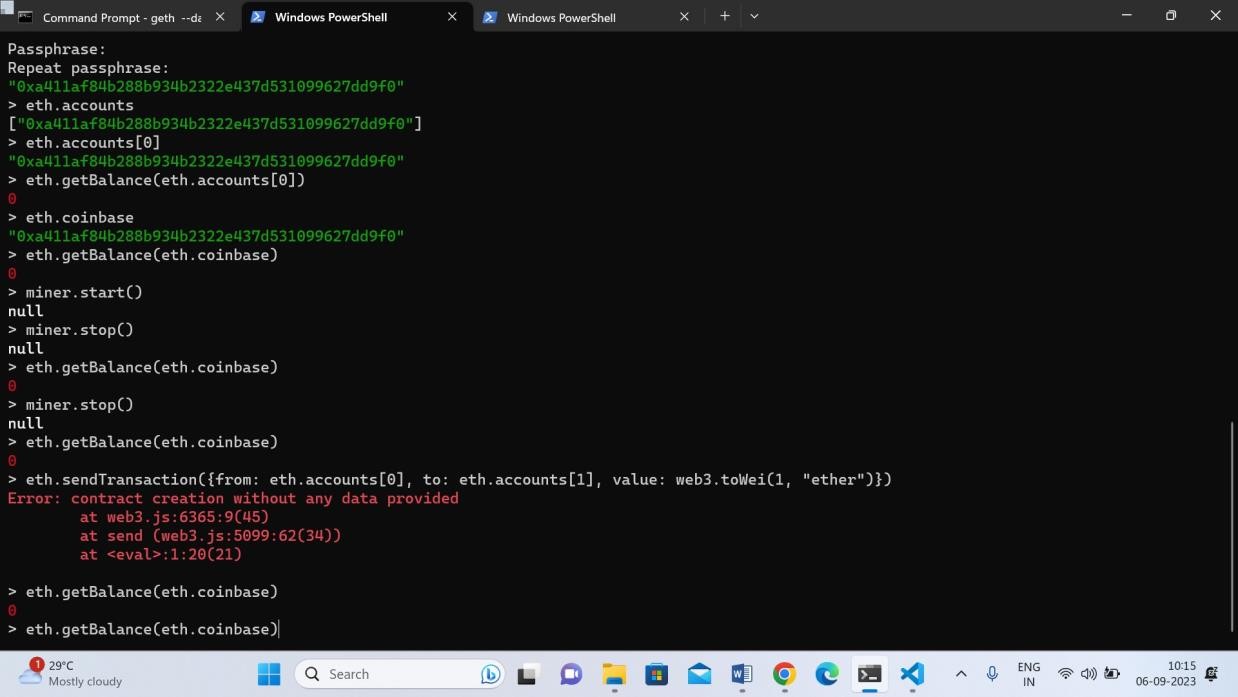
## Set Default Account

* + Check your default account, type
    - eth.coinbase



## Start Mining

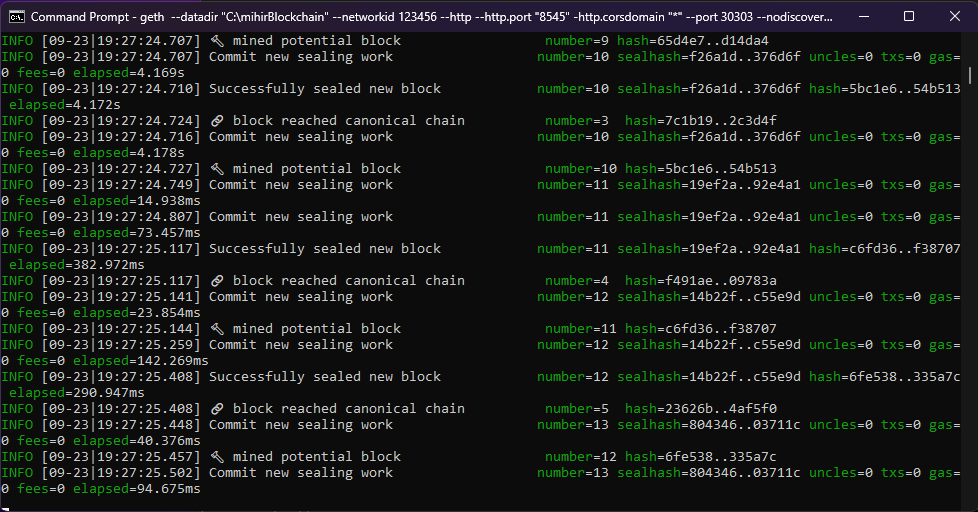
* + Check your balance with
    - eth.getBalance(eth.coinbase)

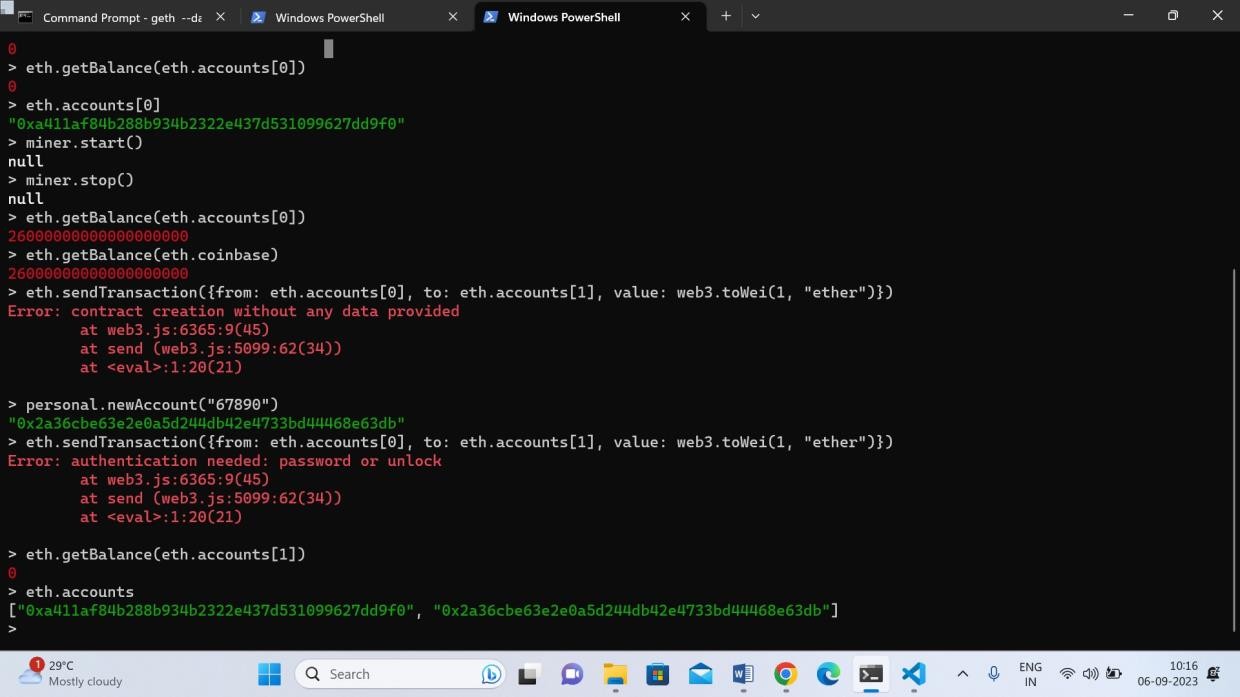


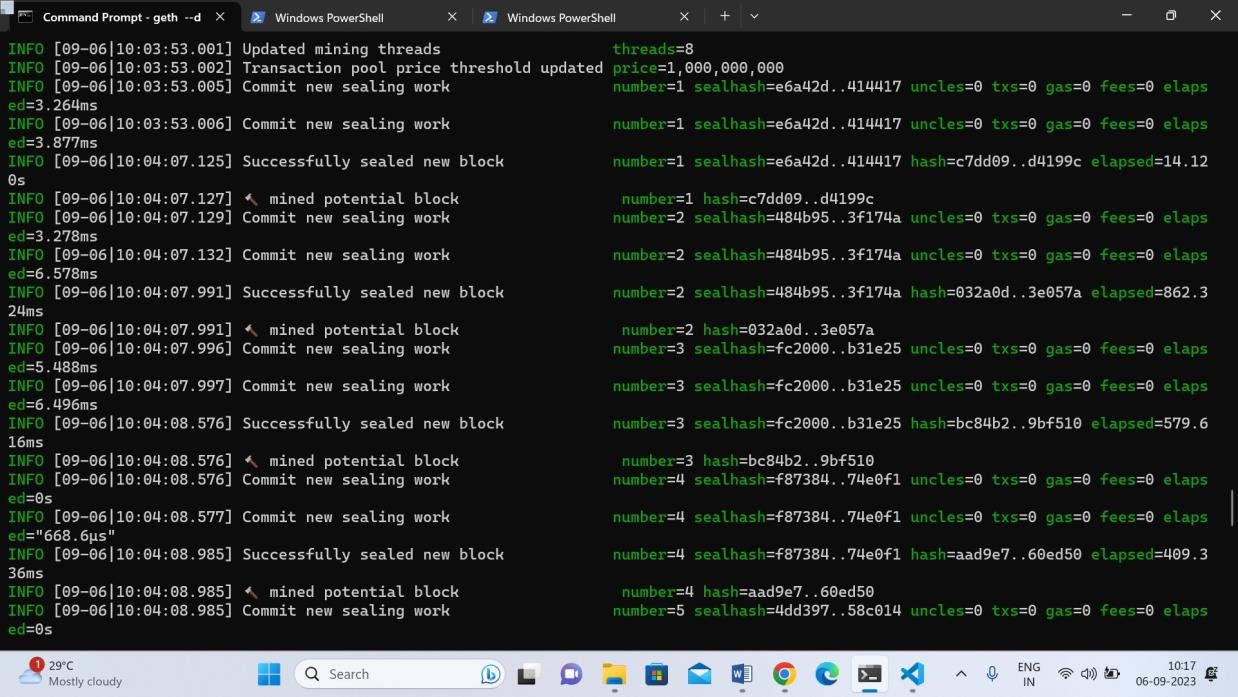
* + Run
    - miner.start()



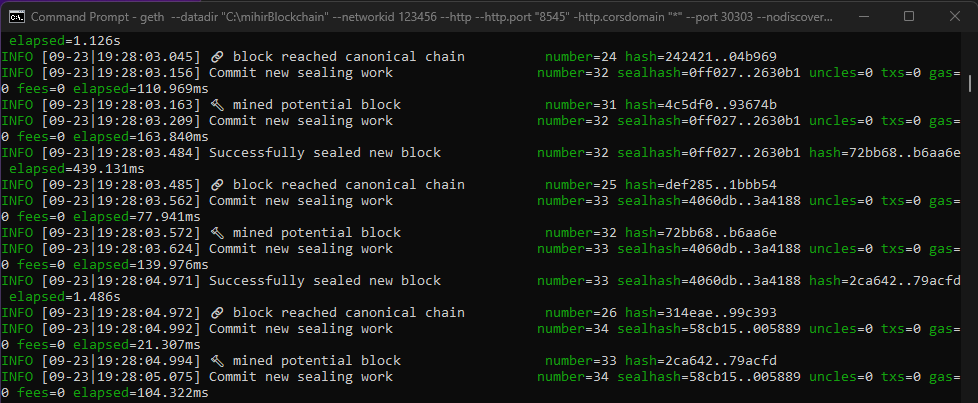
* + Now, Mining will get started on the Blockchain network i.e., on the first Command Prompt
  + Look at your other terminal window, you should see some mining action in the logs.

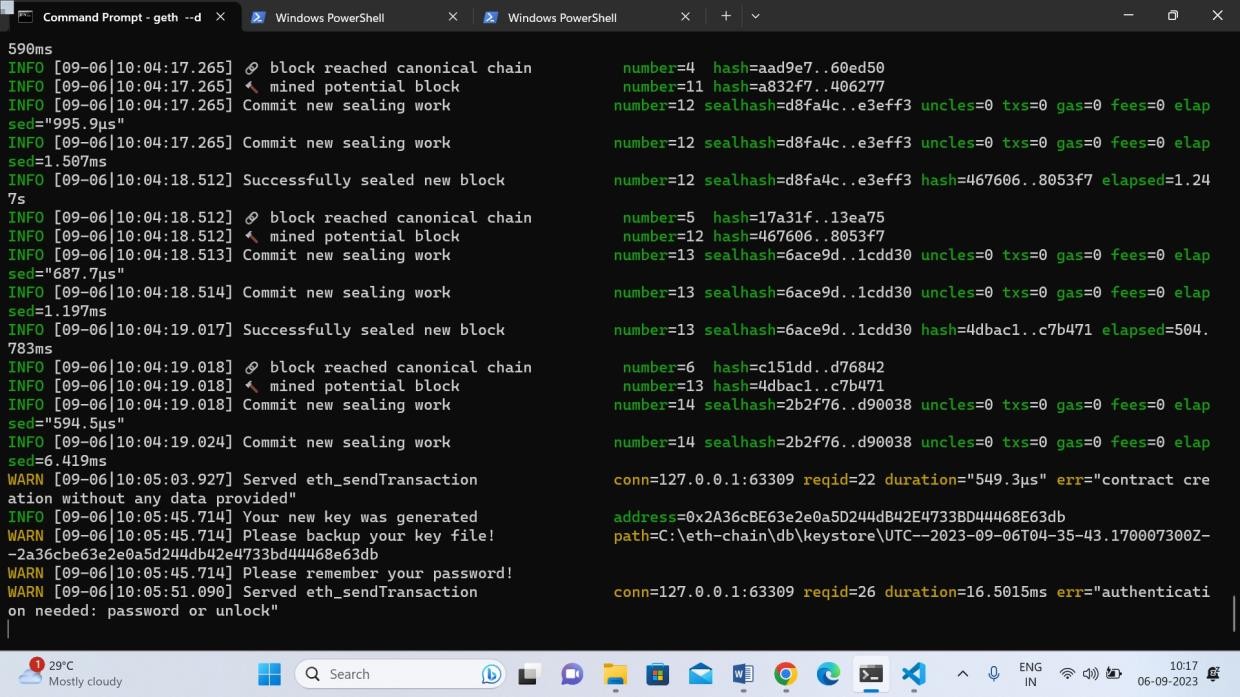






* + To end mining, type
    - miner.stop()

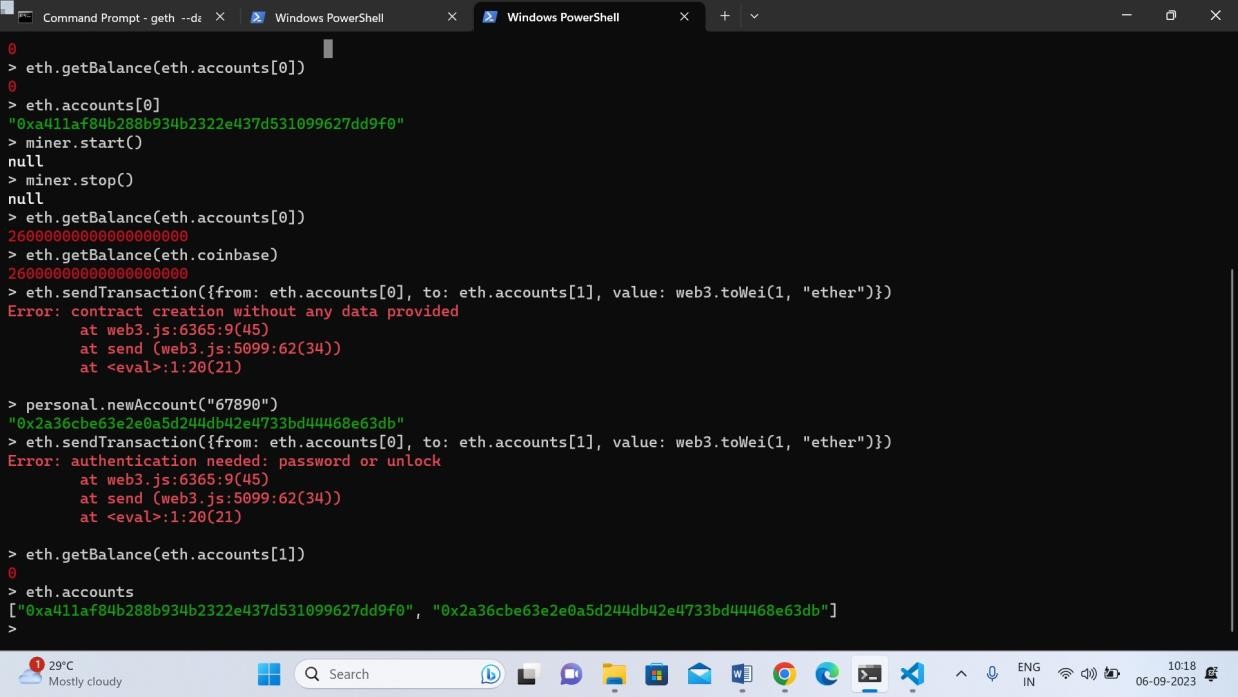




**Step 11: Transfer**

## Check your balance

* eth.getBalance(eth.coinbase)



## Transfer Ether, type:

* eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value:

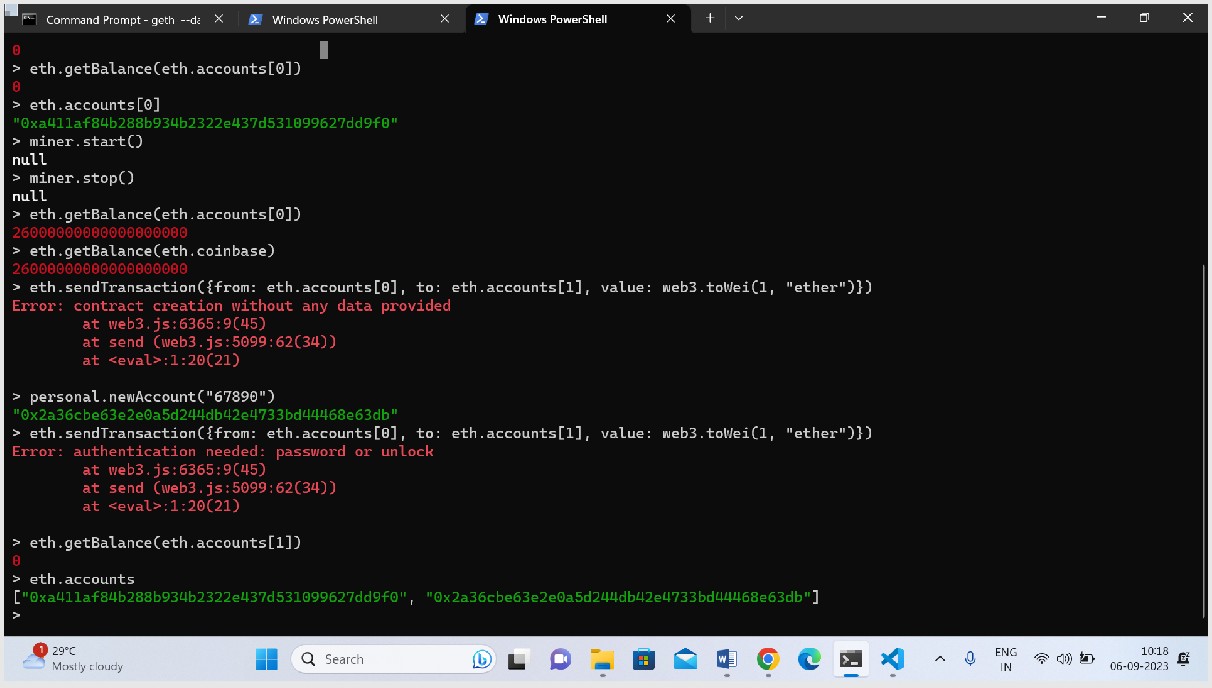
web3.toWei(1, "ether")})

## OR

* eth.sendTransaction({from: 'your\_first\_account\_address', to:

'your\_second\_account\_address', value: web3.toWei(1, "ether")})

But you should get error like this.



Actually, User have to unlock account first before transfer.

**Conclusion:** We have succesfully installed Geth and performed all the commands for various operations.

