

## گام اول: نصب geth

```
|sadeghmacbook@sadeghs-MacBook ~ % geth --help
NAME:
  geth - the go-ethereum command line interface

USAGE:
  geth [global options] command [command options] [arguments...]

VERSION:
  1.14.3-stable

COMMANDS:
  account          Manage accounts
  attach           Start an interactive JavaScript environment (connect to node)
  console          Start an interactive JavaScript environment
  db               Low level database operations
  dump             Dump a specific block from storage
  dumpconfig       Export configuration values in a TOML format
  dumpgenesis     Dumps genesis block JSON configuration to stdout
  export           Export blockchain into file
  export-history   Export blockchain history to Era archives
  import           Import a blockchain file
  import-history   Import an Era archive
  import-preimages Import the preimage database from an RLP stream
  init             Bootstrap and initialize a new genesis block
  js               (DEPRECATED) Execute the specified JavaScript files
  license          Display license information
  removedb         Remove blockchain and state databases
  show-deprecated-flags Show flags that have been deprecated
  snapshot         A set of commands based on the snapshot
  verkle           A set of experimental verkle tree management commands
  version          Print version numbers
  version-check    Checks (online) for known Geth security vulnerabilities
  wallet           Manage Ethereum presale wallets
  help, h          Shows a list of commands or help for one command
```

- **ACCOUNT COMMANDS:**

account new: Creates a new Ethereum account. •  
 account list: Lists all existing Ethereum accounts. •  
 account update: Updates an existing account, for example, changing the password. •  
 account import: Imports an account from a file. •

- **BLOCKCHAIN COMMANDS:**

import <file>: Imports blockchain data from a file. •  
 export <file>: Exports blockchain data to a file. •  
 copydb <source> <destination>: Creates a copy of the database from one location to another. •  
 removedb: Deletes the blockchain database. •

- **CONSOLE COMMANDS:**

console: Opens an interactive JavaScript console (connects to the geth instance). •  
attach <endpoint>: Opens an interactive JavaScript console, attaching to the specified endpoint. •

- **MINER COMMANDS:**

miner start <threads>: Starts the mining process with the specified number of threads. •  
miner stop: Stops the mining process. •  
setEtherbase <address>: Sets the etherbase (mining rewards address) for the node. •

- **NETWORKING COMMANDS:**

addnode <node> <add|remove|connect>: Manages peer connections. •  
peers: Lists all connected peers. •  
admin.nodeInfo: Displays information about the node. •

- **DEV MODE COMMANDS:**

init <genesis.json>: Initializes a new genesis block and definition for a private chain. •  
test: Runs node in test mode. •

- **CONFIGURATION COMMANDS:**

--datadir <directory>: Specifies the data directory for the databases and keystore. •  
--networkid <id>: Network identifier (integer) for the private network. •  
--nodiscover: Disables the peer discovery mechanism. •  
--maxpeers <number>: Limits the number of network peers. •  
--syncmode <mode>: Specifies the blockchain sync mode (full, fast, or light). •  
--ipcpath <path>: Sets the file path for IPC RPC server endpoint. •  
--rpc, --rpcaddr <address>, --rpcport <port>: Enable and configure the HTTP-RPC server. •

- **CHAIN COMMANDS:**

--genesis <file>: Specifies a custom genesis block. •  
--networkid <id>: Specifies the network ID. •  
--syncmode <mode>: Specifies the sync mode (full, fast, light). •  
--verbosity <level>: Sets the logging verbosity level. •

- **ACCOUNT MANAGEMENT:**

account new: Create a new account. •

account update: Update an existing account. •  
account import: Import an account. •  
account export: Export an account. •

- **UTILITY COMMANDS:**

dump: Dumps the state of the blockchain. •  
monitor: Monitor blockchain statistics. •

- **MISCELLANEOUS:**

version: Displays the geth version. •  
help: Displays help information for commands. •

گام دوم: کانفیگ کردن مشخصات بلوک اولیه

```
1  {
2      "config": {
3          "chainId": 15,
4          "homesteadBlock": 0,
5          "eip155Block": 0,
6          "eip158Block": 0
7      },
8      "difficulty": "400000",
9      "gasLimit": "2100000",
10     "alloc": [
11         "[Account #1 Address)": { "balance": "1000000000810100274" },
12         "[Account #2 Address)": { "balance": "2000000000810100274" },
13         "[Account #3 Address)": { "balance": "1500000000810100274" }
14     ]
15 }
```

## گام سوم: ایجاد 3 نود لوکال اتریوم

```
[sadeghmacbook@sadeghs-MacBook ~ % mkdir node01 node02 node03]
```

In a cryptocurrency network, nodes are essential components that participate in maintaining and verifying the blockchain. Here's an explanation of the function of each node and the concepts of Full Node and Light Node:

### Function of Each Node in a Cryptocurrency Network

1. **Storing Blockchain Data:** Nodes store a copy of the blockchain, which includes the entire history of transactions. This ensures the data is distributed and not reliant on a single entity.
2. **Validating Transactions and Blocks:** Nodes validate transactions and new blocks according to the network's consensus rules. This prevents fraudulent transactions and maintains the integrity of the blockchain.
3. **Broadcasting Transactions and Blocks:** Nodes communicate with each other by broadcasting transactions and blocks across the network. This helps to ensure that all nodes have the latest information.
4. **Maintaining Consensus:** Nodes participate in the consensus mechanism (e.g., Proof of Work, Proof of Stake) that determines the state of the blockchain. They agree on the validity of transactions and the order in which they are added to the blockchain.

### Full Node

A Full Node performs several critical functions in the cryptocurrency network:

1. **Complete Blockchain Storage:** Full Nodes store the entire history of the blockchain, from the genesis block to the most recent block.
2. **Transaction and Block Validation:** Full Nodes independently verify all transactions and blocks. They check that all transactions follow the network's rules, ensuring that the blockchain remains accurate and secure.
3. **Support for Light Nodes:** Full Nodes can provide data to Light Nodes. They serve as a source of verified information, allowing Light Nodes to operate without storing the entire blockchain.
4. **Consensus Participation:** Full Nodes participate in the consensus process, helping to confirm new blocks and reject invalid ones. This ensures the security and consistency of the blockchain.

#### Advantages:

- Higher security as they verify everything independently.
- Contribute to the network's decentralization and resilience.

### **Disadvantages:**

- Require significant storage and bandwidth.
- Higher computational requirements.

## **Light Node**

A Light Node (also known as SPV - Simplified Payment Verification node) operates with less resource intensity:

1. **Partial Blockchain Storage:** Light Nodes do not store the entire blockchain. Instead, they download only the block headers, which contain summary information about the blocks.
2. **Rely on Full Nodes:** Light Nodes rely on Full Nodes to retrieve the necessary data for transaction verification. They query Full Nodes for specific information as needed.
3. **Basic Validation:** Light Nodes perform basic transaction validation by checking if a transaction is included in a block and verifying the block headers. They do not verify the entire history of the blockchain.

### **Advantages:**

- Lower storage and bandwidth requirements.
- Can run on less powerful hardware, such as mobile devices.

### **Disadvantages:**

- Dependence on Full Nodes for data.
- Reduced security as they do not independently verify the entire blockchain.

## گام چهارم: ایجاد اکانت و شروع به کار گرہ ها

```
[sadeghmacbook@sadeghs-MacBook ~ % geth --datadir node01 account new
INFO [05-28|09:55:47.740] Maximum peer count           ETH=50 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: 0xbcB839BbA6909623957bc7eC97976822Af53Ccab
Path of the secret key file: node01/keystore/UTC--2024-05-28T06-26-00.240850000Z--bcb839bba6909623957bc7ec97976822af53ccab

- 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!

sadeghmacbook@sadeghs-MacBook ~ % geth --datadir node02 account new
INFO [05-28|09:56:49.747] Maximum peer count           ETH=50 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
[Your new key was generated

Public address of the key: 0xf06e22924382cb901186790B645B36f16d4DE456
Path of the secret key file: node02/keystore/UTC--2024-05-28T06-26-58.073860000Z--f06e22924382cb901186790b645b36f16d4de456

- 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!

sadeghmacbook@sadeghs-MacBook ~ % geth --datadir node03 account new
INFO [05-28|09:57:10.170] Maximum peer count           ETH=50 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: 0xFbC61fB665A71e0Bf667F9e748fBC75a1100eb3c
Path of the secret key file: node03/keystore/UTC--2024-05-28T06-27-18.003907000Z--fbc61fb665a71e0bf667f9e748fbc75a1100eb3c

- 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!
```

جایگزاری آدرس ها در فایل `genesis.json`

```
1  {
2      "config": {
3          "chainId": 15,
4          "homesteadBlock": 0,
5          "daoForkSupport": true,
6          "eip150Block": 0,
7          "eip155Block": 0,
8          "eip158Block": 0,
9          "terminalTotalDifficultyPassed": true
10     },
11     "difficulty": "400000",
12     "gasLimit": "2100000",
13     "alloc": {
14         "0xbcB839BbA6909623957bc7eC97976822Af53Ccab": { "balance": "10000000000810100274" },
15         "0xf06e22924382cb901186790B645B36f16d4DE456": { "balance": "20000000000810100274" },
16         "0xFbC61fB665A71e0Bf667F9e748fBC75a1100eb3c": { "balance": "15000000000810100274" }
17     }
18 }
```

## گام چهارم: اینشیلایز کردن نودها و اجرای آن

```
sadeghmacbook@sadeghs-MacBook ~ % geth --datadir "node01" init ./genesis.json
[INFO] [05-28|11:07:08.978] Maximum peer count                                     ETH=50 total=50
[INFO] [05-28|11:07:08.989] Set global gas cap                                       cap=50,000,000
[INFO] [05-28|11:07:08.990] Initializing the KZG library                           backend=gokzg
[INFO] [05-28|11:07:09.071] Defaulting to pebble as the backing database          database=/Users/sadeghmacbook/node01/geth/chaindata cache=16.00MiB handles=16
[INFO] [05-28|11:07:09.071] Allocated cache and file handles                      database=/Users/sadeghmacbook/node01/geth/chaindata/ancient/chain readonly=false
[INFO] [05-28|11:07:09.179] Opened ancient database                                scheme=hash
[INFO] [05-28|11:07:09.181] State schema set to default
[INFO] [05-28|11:07:09.181] Writing custom genesis block
[INFO] [05-28|11:07:09.190] Persisted trie from memory database                  nodes=6 size=736.00B time=6.645584ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=0 livesize=0.00B
[INFO] [05-28|11:07:09.229] Successfully wrote genesis state                         database=chaindata hash=c413f1..705862
[INFO] [05-28|11:07:09.229] Defaulting to pebble as the backing database          database=/Users/sadeghmacbook/node01/geth/lightchaindata cache=16.00MiB handles=16
[INFO] [05-28|11:07:09.229] Allocated cache and file handles                      database=/Users/sadeghmacbook/node01/geth/lightchaindata/ancient/chain readonly=false
[INFO] [05-28|11:07:09.294] Opened ancient database                                scheme=hash
[INFO] [05-28|11:07:09.294] State schema set to default
[INFO] [05-28|11:07:09.294] Writing custom genesis block
[INFO] [05-28|11:07:09.298] Persisted trie from memory database                  nodes=6 size=736.00B time=3.689083ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=0 livesize=0.00B
[INFO] [05-28|11:07:09.341] Successfully wrote genesis state                         database=lightchaindata hash=c413f1..705862
sadeghmacbook@sadeghs-MacBook ~ % geth --datadir "node02" init ./genesis.json
[INFO] [05-28|11:07:19.698] Maximum peer count                                     ETH=50 total=50
[INFO] [05-28|11:07:19.704] Set global gas cap                                       cap=50,000,000
[INFO] [05-28|11:07:19.705] Initializing the KZG library                           backend=gokzg
[INFO] [05-28|11:07:19.775] Defaulting to pebble as the backing database          database=/Users/sadeghmacbook/node02/geth/chaindata cache=16.00MiB handles=16
[INFO] [05-28|11:07:19.775] Allocated cache and file handles                      database=/Users/sadeghmacbook/node02/geth/chaindata/ancient/chain readonly=false
[INFO] [05-28|11:07:19.856] Opened ancient database                                scheme=hash
[INFO] [05-28|11:07:19.856] State schema set to default
[INFO] [05-28|11:07:19.862] Writing custom genesis block
[INFO] [05-28|11:07:19.862] Persisted trie from memory database                  nodes=6 size=736.00B time=5.147667ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=0 livesize=0.00B
[INFO] [05-28|11:07:19.989] Successfully wrote genesis state                         database=chaindata hash=c413f1..705862
[INFO] [05-28|11:07:19.989] Defaulting to pebble as the backing database          database=/Users/sadeghmacbook/node02/geth/lightchaindata cache=16.00MiB handles=16
[INFO] [05-28|11:07:19.989] Allocated cache and file handles                      database=/Users/sadeghmacbook/node02/geth/lightchaindata/ancient/chain readonly=false
[INFO] [05-28|11:07:19.995] Opened ancient database                                scheme=hash
[INFO] [05-28|11:07:19.995] State schema set to default
[INFO] [05-28|11:07:19.996] Writing custom genesis block
[INFO] [05-28|11:07:20.008] Persisted trie from memory database                  nodes=6 size=736.00B time=4.523791ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=0 livesize=0.00B
[INFO] [05-28|11:07:20.040] Successfully wrote genesis state                         database=lightchaindata hash=c413f1..705862
sadeghmacbook@sadeghs-MacBook ~ % geth --datadir "node03" init ./genesis.json
[INFO] [05-28|11:07:25.898] Maximum peer count                                     ETH=50 total=50
[INFO] [05-28|11:07:25.984] Set global gas cap                                       cap=50,000,000
[INFO] [05-28|11:07:25.985] Initializing the KZG library                           backend=gokzg
[INFO] [05-28|11:07:25.979] Defaulting to pebble as the backing database          database=/Users/sadeghmacbook/node03/geth/chaindata cache=16.00MiB handles=16
[INFO] [05-28|11:07:25.979] Allocated cache and file handles                      database=/Users/sadeghmacbook/node03/geth/chaindata/ancient/chain readonly=false
[INFO] [05-28|11:07:26.061] Opened ancient database                                scheme=hash
[INFO] [05-28|11:07:26.061] State schema set to default
[INFO] [05-28|11:07:26.061] Writing custom genesis block
[INFO] [05-28|11:07:26.067] Persisted trie from memory database                  nodes=6 size=736.00B time=5.271583ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=0 livesize=0.00B
[INFO] [05-28|11:07:26.107] Successfully wrote genesis state                         database=chaindata hash=c413f1..705862
[INFO] [05-28|11:07:26.107] Defaulting to pebble as the backing database          database=/Users/sadeghmacbook/node03/geth/lightchaindata cache=16.00MiB handles=16
[INFO] [05-28|11:07:26.107] Allocated cache and file handles                      database=/Users/sadeghmacbook/node03/geth/lightchaindata/ancient/chain readonly=false
[INFO] [05-28|11:07:26.179] Opened ancient database                                scheme=hash
[INFO] [05-28|11:07:26.179] State schema set to default
[INFO] [05-28|11:07:26.179] Writing custom genesis block
[INFO] [05-28|11:07:26.184] Persisted trie from memory database                  nodes=6 size=736.00B time=4.350708ms gcnodes=0 gcsize=0.00B gctime=0s livenodes=0 livesize=0.00B
[INFO] [05-28|11:07:26.223] Successfully wrote genesis state                         database=lightchaindata hash=c413f1..705862
sadeghmacbook@sadeghs-MacBook ~ %
```

حال اقدام به استارت زدن هر کدام از نودها در ترمینالی متفاوت با استفاده از دستور فوق مینماییم :

## Node1:

```
sadeghmacbook@Sadegh-MacBook ~ % geth --allow-insecure-unlock --identity "node01" --http --http.port "8000" --http.corsdomain "*" --datadir "node01" --port "30303" --nodiscover --http.api "db,eth,net,web3,personal,miner,admin" --networkid 15 -- "any" console
Ethereum peer count: 0
Set global gas cap: 65536
Allocated memory for XDG library: 128.00MiB
Allocated Trie memory caches: 154.00MiB dirty=256.00MiB
Using pebble as the backing database
Allocated cache and file handles database=/Users/sadeghmacbook/node01/geth/chaindata cache=512.00MiB handles=5120
Opened ancient database database=/Users/sadeghmacbook/node01/geth/chaindata/ancient/chain readonly=false
State scheme set to already existing scheme=hash
Initialising Ethereum protocol network=15 dbversion=<nil>
[...]
Chain ID: 15 (unknown)
Consensus: unknown
[...]
Pre-Merge hard forks (block based):
Homestead: #0 (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/homestead.md)
Tangerine Whistle (CIP 150): #0 (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/tangerine-whistle.md)
Spurious Dragon (CIP 151): #0 (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md)
Spurious Dragon2 (CIP 150): #0 (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon2.md)
Byzantium: #cnil (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/byzantium.md)
Constantinople: #cnil (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/constantinople.md)
Petersburg: #cnil (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/petersburg.md)
Istanbul: #cnil (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/istanbul.md)
Berlin: #cnil (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/berlin.md)
London: #cnil (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/london.md)
The Merge is not yet available for this network!
Hard fork specification: Hard-fork specification: https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md
Post-Merge hard forks (timestamp based):
[...]
Loaded most recent local block number=0 hash=3d0b8e..741cf7 tc=400,000 age=55y2mo2w
Failed to load snapshot err="missing or corrupted snapshot"
Rebuilding state snapshot range="last 2350000 blocks"
Initialized transaction indexer range="last 2350000 blocks" storage=0.000 dangling=0 elapsed=3.81ms
Generated state snapshot storage=0.000 dangling=0 elapsed=14.792ms
Enabled snap sync header=0 hash=3d0b8e..741cf7
Chain configured post-merge, but without TTD. Are you debugging sync?
Gasprie oracle in ignore threshold set threshold=2
WARN [06-01|13:23:40.961] Failed to load snapshot
WARN [06-01|13:23:40.968] Starting peer-to-peer node
INFO [06-01|13:23:41.113] Starting peer-to-peer node instance=Geth/node01/v1.13.15-stable-c5ba367edarwin-amd64/go1.21.6
INFO [06-01|13:23:41.139] IPC endpoint opened url=/Users/sadeghmacbook/node01/geth.ipc
ERROR [06-01|13:23:41.139] Unavailable modules in HTTP API list unavailable=[db] available=["admin debug web3 eth txpool personal miner net"]
INFO [06-01|13:23:41.143] HTTP server started endpoint=:127.0.0.1:8090 auth=false prefix= cors= vhost=localhost
INFO [06-01|13:23:41.148] New local node record seq=1,737,235,421,137 id=8d52f1ff47826e2 ip=127.0.0.1 udp=0 tcp=30303
INFO [06-01|13:23:41.148] Started P2P networking self=:enode://a915d77c53fb72bb16e593e24beef09843e48b462f9bc2073cf87084d98a357a286d5dc20dc7962d315f5fafd834c94c2738b569891447af72d2a89e51809726a0127.0.0.1:30303?discport=0
INFO [06-01|13:23:41.148] WebSocket enabled urlws://127.0.0.1:8551
INFO [06-01|13:23:41.211] Stopped HTTP server, started endpoint=:127.0.0.1:8551 auth=true prefix= cors=localhost vhost=localhost
WARN [06-01|13:23:41.211] Stopped database
Welcome to the Geth JavaScript console!
```

## Node2:

```
sadeghmacbook:Sadeghs-MacBook ~ % geth --allow-insecure-unlock --identity "node02" --http --http.port "8001" --http.corsdomain "*" --datadir "node02" --port "39394" --nodiscover --http.api "db,eth,net,web3,personal,miner,admin" --networkid 15 --name "any" --authrpc.port 8852 console

INFO [06-01|13:33:16.366] Maximum peer count                                     ETHER9 total=69
INFO [06-01|13:33:16.373] Set global gas cap                                     caps=50,000,000
INFO [06-01|13:33:16.373] Initializing the KZG library                           backlog=qokz
INFO [06-01|13:33:16.421] Allocated trie memory caches                         cleanup154.00MiB dirty=266.00MiB
INFO [06-01|13:33:16.443] Using pebble as the backing database                   database=/Users/sadeghmacbook/node02/geth/chaindata cache=612.00MiB handle=6120
INFO [06-01|13:33:16.443] Allocated cache and file handles                      database=/Users/sadeghmacbook/node02/geth/chaindata/ancient/chain readonly=false
INFO [06-01|13:33:16.451] State scheme set to already existing                     schemaHash
INFO [06-01|13:33:16.451] Initializing Ethereum protocol                         network=15 dbversion=8

INFO [06-01|13:33:16.451] Chain ID: 15 (unknown)
INFO [06-01|13:33:16.451] Consensus: unknown

INFO [06-01|13:33:16.515] Pre-Merge hard forks (block based):
INFO [06-01|13:33:16.515]   - Homestead: #0                                         (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/homestead.md)
INFO [06-01|13:33:16.515]   - Spurious Dragon/0 (IP 150): #0                               (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/tangerine-whistle.md)
INFO [06-01|13:33:16.515]   - Spurious Dragon/1 (IP 155): #0                               (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md)
INFO [06-01|13:33:16.515]   - Spurious Dragon/2 (IP 158): #0                               (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md)
INFO [06-01|13:33:16.515]   - Byzantium: #cn1                                           (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/byzantium.md)
INFO [06-01|13:33:16.515]   - Constantinople: #cn2                                         (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/byzantium.md)
INFO [06-01|13:33:16.515]   - Constantinople: #cn3                                         (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/byzantium.md)
INFO [06-01|13:33:16.515]   - Istanbul: #cn1                                           (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/istanbul.md)
INFO [06-01|13:33:16.515]   - Berlin: #cn1                                           (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/berlin.md)
INFO [06-01|13:33:16.515]   - London: #cn1                                           (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/london.md)

INFO [06-01|13:33:16.515] Hard fork specification: https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md
INFO [06-01|13:33:16.515] Post-Merge hard forks (timestamp based):
INFO [06-01|13:33:16.515]   - Paris: #0                                         (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md)
INFO [06-01|13:33:16.515]   - London: #0                                         (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/london.md)

INFO [06-01|13:33:16.515] The Merge is not yet available for this network!
INFO [06-01|13:33:16.515] Hard-fork specification: https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md

INFO [06-01|13:33:16.515] Post-Merge hard forks (timestamp based):
INFO [06-01|13:33:16.515]   - Paris: #0                                         (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md)
INFO [06-01|13:33:16.515]   - London: #0                                         (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/london.md)

INFO [06-01|13:33:16.515] Loaded most recent local block
INFO [06-01|13:33:16.515]   number=0 hash=3d0b8e..741cf7 to=400,000 age=55y2mo2w
INFO [06-01|13:33:16.515] Loaded snapshot journal
INFO [06-01|13:33:16.515]   initialized transaction indexer
INFO [06-01|13:33:16.515]   last block=0 transaction journal
INFO [06-01|13:33:16.533] Enabled snap sync
INFO [06-01|13:33:16.533] Chain configured post-merge, but without TTD. Are you debugging sync?
ERROR [06-01|13:33:16.533] Gasprie oracle in ignoring threshold set threshold=2
INFO [06-01|13:33:16.533] Unclean shutdown detected
WARN [06-01|13:33:16.538] Engine API enabled
WARN [06-01|13:33:16.538] Peer-to-peer node configuration detected but chain not configured yet
INFO [06-01|13:33:16.538] Starting peer-to-peer node
INFO [06-01|13:33:16.571] IPC endpoint opened
ERROR [06-01|13:33:16.571]Unavailable module in HTTP API list
INFO [06-01|13:33:16.571] New local node record
INFO [06-01|13:33:16.571] Websocket enabled
INFO [06-01|13:33:16.571] Started P2P networking
INFO [06-01|13:33:16.577] WebSocket enabled
INFO [06-01|13:33:16.577] HTTP server started
INFO [06-01|13:33:16.577] Geth Java client interface available
Welcome to the Geth JavaScript console!

instance: Geth/node02/v1.13.15-stable-c5ba367e/darwin-amd64/go1.21.6
at block: 0 (Thu Jan 01 1970 03:38:00 GMT+0330 (+0330))
datadir: /Users/sadeghmacbook/node02
modules: admin:1.0 debug:1.0 engine:1.0 eth:1.0 miner:1.0 net:1.0 rpc:1.0 txpool:1.0 web3:1.0
```

## Node3:

```
sadeghmacbook@Sadegh-MacBook ~ % path --allow-insecure-unlock --identity "node03" --http --http.port "8002" --http.corsdomain "*" --datadir "node03" --port "30305" --nodiscover --http.api "db,eth,net,web3,personal,miner,admin" --networks "15" --nat "any" --authrpc.port 8653 console

INFO [06-01|13:36:36.888] Maximum peer count                                     ETHER=50 total=50
INFO [06-01|13:36:36.894] Set global gas cap                                      cap=50,000,000
INFO [06-01|13:36:36.894] Set global txpool size                                txpool=8192
INFO [06-01|13:36:36.194] Allocated trie memory caches                           backend=pbx2g
INFO [06-01|13:36:36.197] Using pebble as the backing database                   clear=164.00MiB dirty=256.00MiB
INFO [06-01|13:36:36.197] Allocated cache and file handles                      database=/Users/sadeghmacbook/node03/geth/chaindata cache=512.00MiB handles=5120
INFO [06-01|13:36:36.271] Opened ancient database                                 database=/Users/sadeghmacbook/node03/geth/chaindata/ancient chain readonly=false
INFO [06-01|13:36:36.271] State scheme set to already existing                  schema=hash
INFO [06-01|13:36:36.271] Initialising Ethereum protocol                         network=15 dbversion=8
INFO [06-01|13:36:36.274]
INFO [06-01|13:36:36.274] Pre-Merge hard forks (block based):
INFO [06-01|13:36:36.274] - Homestead: #0                                         (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/homestead.md)
INFO [06-01|13:36:36.274] - Tangerine Whistle (EIP 150): #0                     (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/tangerine-whistle.md)
INFO [06-01|13:36:36.274] - Spurious Dragon1 (EIP 158): #0                   (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon1.md)
INFO [06-01|13:36:36.274] - Spurious Dragon2 (EIP 160): #0                   (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon2.md)
INFO [06-01|13:36:36.274] - Byzantium: #cn1                                         (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/byzantium.md)
INFO [06-01|13:36:36.274] - Constantinople: #cn1                         (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/constantinople.md)
INFO [06-01|13:36:36.274] - Petersburg: #cn1                           (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/petersburg.md)
INFO [06-01|13:36:36.274] - London: #cn1                            (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/london.md)
INFO [06-01|13:36:36.274] - Berlin: #cn1                           (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/berlin.md)
INFO [06-01|13:36:36.274] - Paris: #cn1                           (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md)
INFO [06-01|13:36:36.274] The Merge is not yet available for this network!
INFO [06-01|13:36:36.274] Hard-fork specification: https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md
INFO [06-01|13:36:36.274] Post-Merge hard forks (timestamp based):
INFO [06-01|13:36:36.274]
INFO [06-01|13:36:36.274] Pre-Merge hard forks (timestamp based):
INFO [06-01|13:36:36.274] Loaded most recent local block                      number=0 hash=3d0b8e..741cf7 td=400,000 age=55y2m02w
WARN [06-01|13:36:36.274] Loaded snapshot journal                           diff=missing
INFO [06-01|13:36:36.274] Initialized transaction indexer                      range=last 2350000 blocks*
INFO [06-01|13:36:36.275] Loaded local transaction journal                   transactions=0 dropped=0
INFO [06-01|13:36:36.275] Started sync with system                               peer=0 hash=3d0b8e..741cf7
ERROR [06-01|13:36:36.284] Chain configured post-merge, but without the merge debugging sync?
INFO [06-01|13:36:36.287] Gasprice oracle is ignoring threshold set threshold=>
WARN [06-01|13:36:36.293] Unclean shutdown detected                          booted=2024-06-01T13:34:27+0330 age=1m9s
WARN [06-01|13:36:36.294] Ethereum API enabled                                protocol=eth
WARN [06-01|13:36:36.294] Ethereum API enabled but chain not configured for merge!
INFO [06-01|13:36:36.294] Starting peer-to-peer node                         url=/Users/sadeghmacbook/node03/geth.ipc
INFO [06-01|13:36:36.326] P2P endpoint opened                                unavailable[db] available=[admin debug web3 eth txpool personal miner net]
ERROR [06-01|13:36:36.326] Unavailable module in HTTP API list                pat=/Users/sadeghmacbook/node03/geth/jwtsecret
INFO [06-01|13:36:36.326] Generated JWT secret                                endpoint=127.0.0.1:8080 auth=false prefix=corss vhost=localhost
INFO [06-01|13:36:36.326] HTTP server started                                req=1,199,9564761d1e8a ->127.0.0.1 ip=0.0.0.0 port=30305
INFO [06-01|13:36:36.328] Started P2P networking                           sel=endee://15faa585a2933edc754e0c444a82995764a5bbcb1b797ca2c61763d7bc8fab507c5759290263f5986eb8d13fb058d999fbadca691c66b3799e5fd5ad8420127.0.0.1:30305?discport=0*
INFO [06-01|13:36:36.330] WebSocket enabled                                urls=ws://127.0.0.1:8553 auth=true prefix=corss vhost=localhost
INFO [06-01|13:36:36.330] HTTP server started                                endpoint=127.0.0.1:8553 auth=true prefix=corss vhost=localhost
WARN [06-01|13:36:36.381] Served eth_coinbase                                req=3 duration=49.157us err="etherbase must be explicitly specified"
Welcome to the Geth JavaScript console!
```

## گام پنجم: وصل شدن به نودها از طریق یک کلاینت

هم اکنون با استفاده از دستور فوق در یک ترمینال مجزا از ترمینال های قبلی، به عنوان یک کلاینت به یکی از نودها به صورت لوکال متصل شوید.

```
[sadeghmacbook@sadeghs-MacBook ~ % geth attach http://127.0.0.1:8000
WARN [06-01|18:40:55.767] Enabling deprecated personal namespace
Welcome to the Geth JavaScript console!

instance: Geth/node01/v1.13.15-stable-c5ba367e/darwin-amd64/go1.21.6
at block: 0 (Thu Jan 01 1970 03:30:00 GMT+0330 (+0330))
datadir: /Users/sadeghmacbook/node01
modules: admin:1.0 eth:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 web3:1.0

To exit, press ctrl-d or type exit
```

و مشاهده اطلاعات مربوط به گر:

```
> admin.nodeInfo
{
  enode: "enode://e915d77c5c38f872bb61e693e24bee09043e408462f8c2073cf870b4d90a357a286d5dc20dc7962d315f5fafd834c94c2730b569091467af72d2a89e5180926a@127.0.0.1:30303?discport=0",
  enr: "enr:-y4QBFTj99DRTcx0RK1zQgwN-y_icftSyaQJLSXTI-BSR_Kmh91Pr1tBOMuPj1hipjJ_wfhm52wZriALF70wC186GAY_TQ80Rg2V0aMfGhLzdhmAgm1kgny0gm1whH8AAAGJc2VjCD1NmsoQlPfd8X0j4crthSpPiS-4JB05AhGL4wgC8-HC02Qo1eoRzimfwuIN0Y3CC18",
  id: "0d521ff417828e2d06147d10ce6dd5923ae0e96e8b881dcadcb1a0cef7dcf008",
  ip: "127.0.0.1",
  listenAddr: "{::}:30303",
  name: "Geth/node01/v1.13.15-stable-c5ba367e/darwin-amd64/go1.21.6",
  port: {
    discovery: 0,
    listener: 30303
  },
  protocols: {
    eth: {
      config: {
        difficultyCap: 15,
        doForkSupport: true,
        eip158Block: 0,
        eip158BlockLock: 0,
        eip158BlockLockLock: 0,
        homesteadBlock: 0,
        terminalTotalDifficultyPassed: true
      },
      difficulty: 400000,
      genesis: "0x3d0bb826b136fc7edcb9cb66daef22d92f876a3615d50a38ebf56710533741cf7",
      head: "0x3d0bb826b136fc7edcb9cb66daef22d92f876a3615d50a38ebf56710533741cf7",
      network: 15
    },
    snap: {}
  }
}
```

## گام ششم: اتصال نود ها به یکدیگر

```
> admin.addPeer("enode://e915d77c5c38f872bb61e693e24bee09043e408462f8c2073cf870b4d90a357a286d5dc20dc7962d315f5fafd834c94c2730b569091467af72d2a89e5180926a@127.0.0.1:30303?discport=0")
true
```

```
> net.peerCount
2
```

## گام هفتم: ایجاد تراکنش

مشاهده اکانت های موجود و بالанс هر اکانت:

```
[> eth.accounts  
[ "0xbcb839bba6909623957bc7ec97976822af53ccab"]  
[> eth.getBalance(eth.accounts[0])  
1000000000810100274
```

برای شروع تراکنش با استفاده از دستور زیر و دادن رمز عبور اکانت مورد نظر که میخواهیم تراکنش انجام دهد را باز میکنیم:

```
[> personal.unlockAccount(eth.accounts[0])  
Unlock account 0xbcb839bba6909623957bc7ec97976822af53ccab  
[Passphrase:  
true
```

حال تراکنش مذکور را به نودها ارسال میکنیم:

```
[> eth.sendTransaction({from:eth.accounts[0], to:"0xf06e22924382cb901186790B645B36f16d4DE456", value:1000})  
"0x492242c7721d4b81929d238591f3d9926fffbfb7d3dbb9a146343ea26985bf27d"  
> █
```

## گام هشتم: ماین کردن یک بلاک جهت قرارگیری تراکنش در بلاک چین

```
[> miner.setEtherbase(eth.accounts[0])
true
> INFO [06-01|19:09:43.561] Legacy pool tip threshold updated           tip=1,000,000,000
INFO [06-01|19:09:43.567] Snap sync complete, auto disabling
INFO [06-01|19:09:43.571] Commit new sealing work           number=1 sealhash=81cbe...1d9a89 txs=1 gas=21000 fees=2.1e-05 elapsed=2.042ms
```

استارت عملیات ماینینگ:

```
[> miner.start()
INFO [06-01|19:09:56.863] Legacy pool tip threshold updated           tip=1,000,000,000
```

سوال 2 : خروجی مشاهده شده در عملیات ماینینگ را بررسی کنید و هر کدام از لაگ های بدست آمده را به اختصار توضیح دهید.

این لاگ ها شامل اطلاعات مختلفی هستند که به ما کمک می کنند تا وضعیت فرآیند ماینینگ را درک کنیم.

INFO [XX-XX|XX:XX:XX] Starting mining operation

این لاگ نشان می دهد که عملیات ماینینگ آغاز شده است.

INFO [XX-XX|XX:XX:XX] Commit new mining work number=19722001 sealhash...=

این لاگ نشان می دهد که یک بلاک جدید برای استخراج آماده شده است `number` شماره بلاک و `sealhash` هش بلاک جدید است.

INFO [XX-XX|XX:XX:XX] Successfully sealed new block number=19722001 hash...=

این لاگ نشان می دهد که یک بلاک جدید با موفقیت استخراج شده است `number` شماره بلاک و `hash` هش بلاک استخراج شده است.

INFO [XX-XX|XX:XX:XX] Commit new seal: number=19722001 sealhash=... uncles=0 txs=0 gas=0 fees=0

این لاگ نشان می‌دهد که پروسه تولید بلاک جدید شروع شده است `uncles`. تعداد بلاک‌های uncle (بلاک‌های استخراج نشده در فورک‌های کوتاه)، `txs` تعداد تراکنش‌ها در بلاک، `gas` میزان گاز مصرف شده و `fees` کارمزدهای تراکنش‌ها است.

INFO [XX-XX|XX:XX:XX] mined potential block number=19722001 hash...=

این لاگ نشان می‌دهد که بلاک استخراج شده به شبکه ارسال شده است `number`. شماره بلاک و `hash` هش بلاک استخراج شده است.

INFO [XX-XX|XX:XX:XX] Imported new chain segment blocks=1 txs=0 mgas=0 elapsed=XXXms number=19722001 hash...=

این لاگ نشان می‌دهد که بلاک جدیدی به زنجیره بلاک‌چین اضافه شده است `blocks`. تعداد بلاک‌های وارد شده، `txs` تعداد تراکنش‌ها، `mgas` میزان گاز مصرف شده، `elapsed` مدت زمان اضافه شدن بلاک و `number` شماره بلاک و `hash` هش بلاک اضافه شده است.

سوال 3: از سال 2015 اکنون، 19722000 بلاک در زنجیره اصلی شبکه اتریوم ماین شده است. فرض کنید ما در این تمرین به عنوان یک مهاجم، اقدام به ماین کردن سریع در بلاک‌چین لوکال خود نماییم و سعی کنیم تا 19722000 در بلاک‌چین لوکال خود ایجاد کنیم و سپس با `broadcast` کردن آن در شبکه اینترنت، ادعای کنیم که بلاک‌چین ما بلاک‌چین اصلی اتریوم است! آیا قادر به انجام چنین کاری می‌باشیم؟ اگر بله، علت اتفاق را توضیح دهید و اگر خیر، به چه دلیلی قادر به انجام این کار نمی‌باشیم؟

خیر، ما قادر به انجام چنین کاری نخواهیم بود. به این دلیل که بلاک‌چین اتریوم دارای مکانیزم‌های امنیتی مانند اثبات کار (Proof of Work) و اجماع عمومی است. تمام گره‌های شبکه باید تراکنش‌ها و بلاک‌های جدید را تایید کنند. همچنین، طول زنجیره و میزان قدرت هش استفاده شده برای استخراج بلاک‌ها توسط شبکه بررسی می‌شود. بنابراین، بلاک‌چین محلی ما بدون داشتن قدرت هش کافی و تاییدیه از سوی دیگر گره‌ها در شبکه اصلی قابل قبول نخواهد بود.