# Cryptocurrency-CA2


## مبینا مهرآذر - 810100216


## بخش اول: تعامل به شبکه بیت کوین

### قسمت اول : Address Generation

#### سوال 1)

```python
private_key = generate_private_key()
public_key = private_key_to_public_key(private_key)
address = public_key_to_address(public_key)
wif_key = private_key_to_wif(private_key)
print(f"private key : {private_key.hex()}")
print(f"wif key     : {wif_key}")
print(f"public key  : {public_key.hex()}")
print(f"address     : {address}")
```

Python

```
private key : 9509ea9187b319b8fb662384e0f7596e0a5206b003db455ef6bd1f1ce59408e5
wif key     : 92iZA6f2LMJX9mJA5VkQHF5Bb94o2BusvowKpRFNv4pkD7SJcg6
public key  : 0462401c3047f9bd9dbc7fe44b05f4fde4f6e2f662685a50759c9e28f847294052f875c6e662d2
address     : msxYMc5ZYP3BzgavA46gASPDDssaF2aRGr
```

#### سوال 2)

آدرس ونیتی فوق با استفاده از قطعه کد زیر، تولید شده است:

```python
desired_prefix = "4cd"
private_key, wif_key, public_key, address = generate_vanity_address(desired_prefix)

print("Private Key (hex):", private_key.hex())
print("Private Key (WIF):", wif_key)
print("Public Key (hex):", public_key.hex())
print("Bitcoin Testnet Address:", address)

with open("./logs.txt", "w") as f:
    f.write("Private Key (hex): " + private_key.hex() + "\n")
    f.write("Private Key (WIF): " + wif_key + "\n")
    f.write("Public Key (hex): " + public_key.hex() + "\n")
    f.write("Bitcoin Testnet Address: " + address + "\n")
```

Python

```
Private Key (hex): 61678501b340a00c6d8d9356ca6abe3c9d037d207920e2ac309e7f67e9cccde2
Private Key (WIF): 92KpE8WpCaVY2gpv3wdNSBPNadB2fZXZz8ERMoj4TWuTiKp9AYe
Public Key (hex): 04c247f39dc0f2b9d270ecbc83dc5f031aac0d7278fa7dbc0ace7c3bc20824d45785c3c793
Bitcoin Testnet Address: n4cdFzyVNQXv9dwoZ8G9EsV18PS2y5C6TG
```

در تصویر بالا، کلید عمومی، کلید خصوصی، آدرس و کلید خصوصی WIF نشان داده شده است.

کلید خصوصی WIF فرمت خاصی است که برای نمایش کلید خصوصی بیت کوین استفاده می‌شود. این فرمت، کپی کردن، ذخیره و وارد کردن کلیدهای خصوصی در کیف پول و حتی خواندن آن را برای کاربران آسان‌تر می‌کند.

## قسمت دوم : ایجاد تراکنش

**تراکنش 1)** تراکنشی با یک ورودی و دو خروجی ایجاد کنید که خروجي اول آن توسط هیچکس قابل خرج شدن نباشد و خروجی دوم آن توسط هر شخصی قابل خرج شدن باشد. در تراکنشی دیگر خروجی قابل خرج این تراکنش را خرج کرده و به آدرس اصلي خود به صورت خروجی PKH2P بازگردانید.

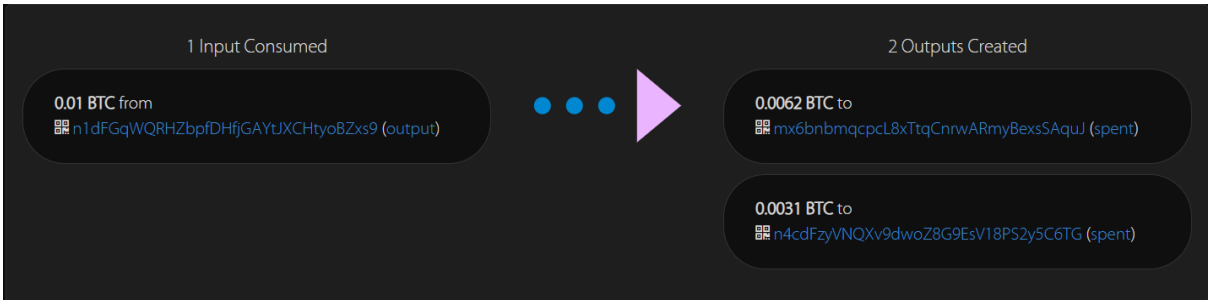
---

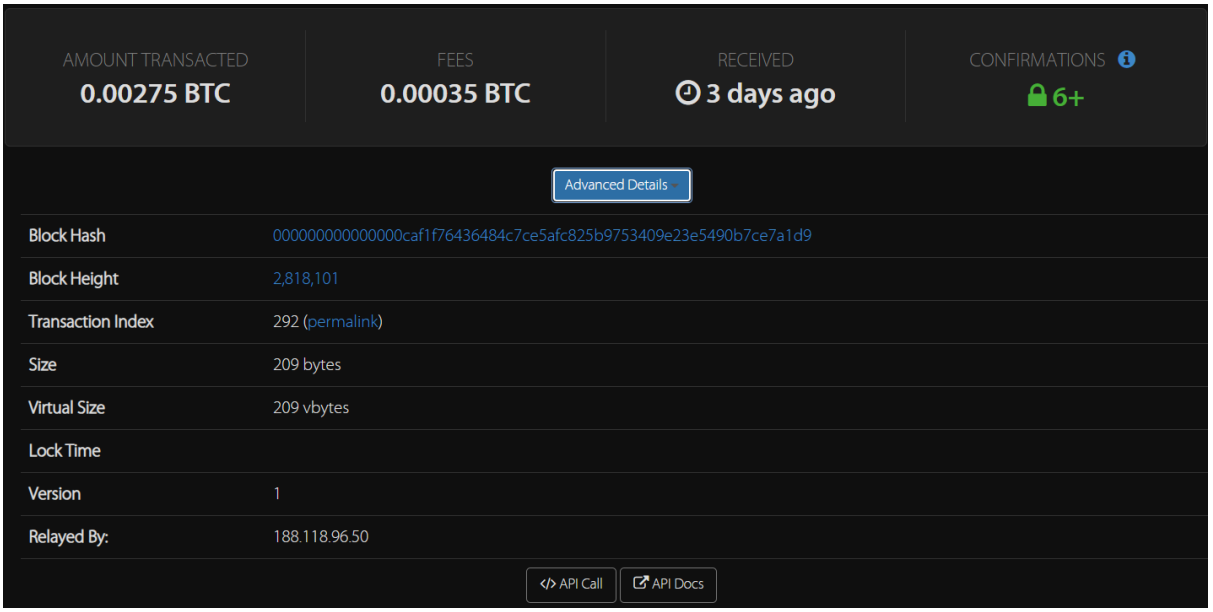
**ارسال**

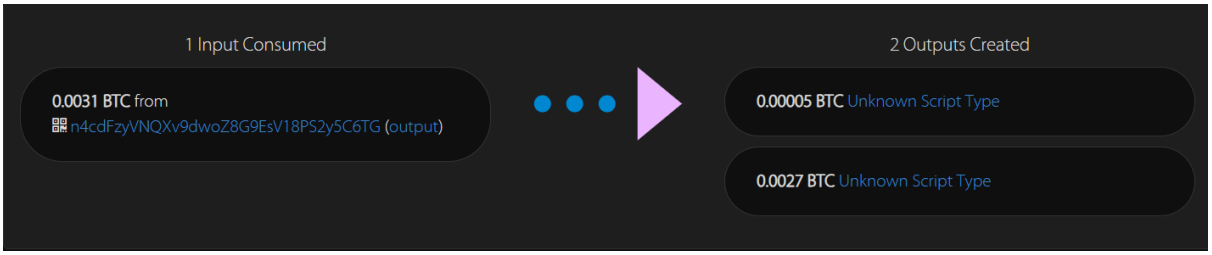
برای انجام این بخش، مقداری پول از اکانتی به آدرسی n1d..xs9 دریافت شده است. دومین مقصد این تراکنش، آدرس ونیتی n4c..C6TG، آدرسی است که در این پروژه از آن استفاده میکنیم.

همانطور که در کد نیز قابل مشاهده است، به مقدار 0.00005 کوین، در این تراکنش می‌سوزد.

**7990000f3f7276a3260d045b97d8c680f15e17d0e6f4f95e904986f82a8c55a6**



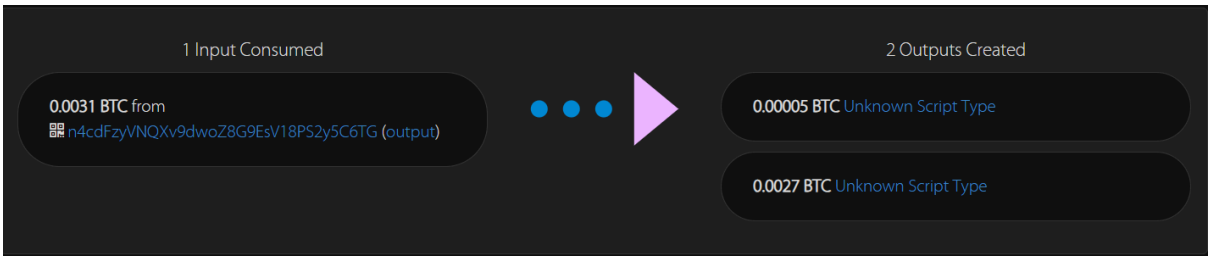


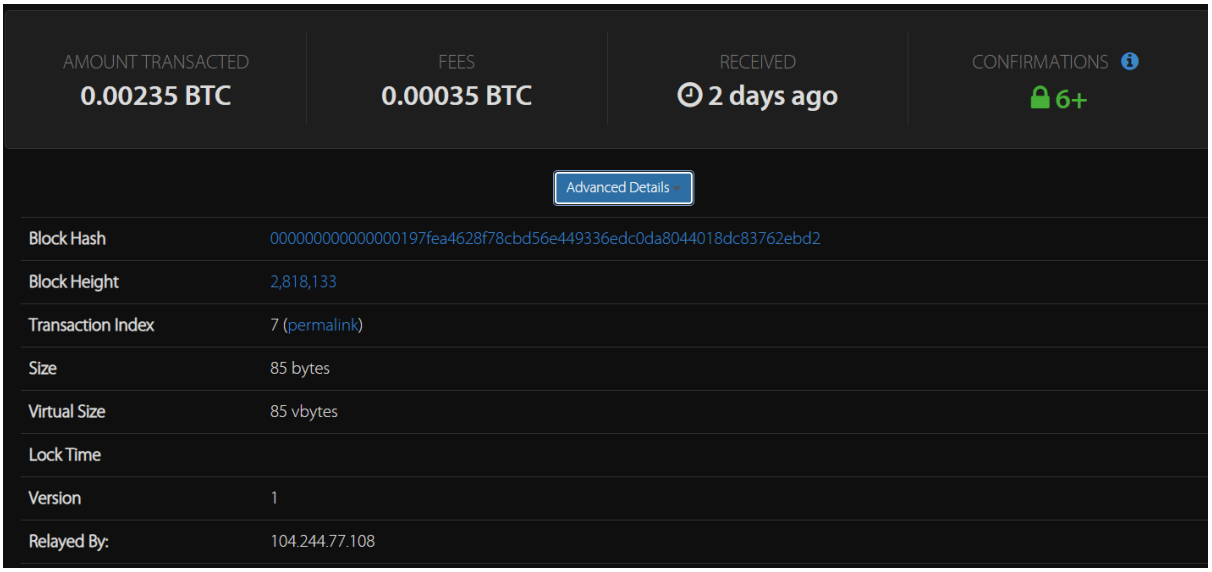

---

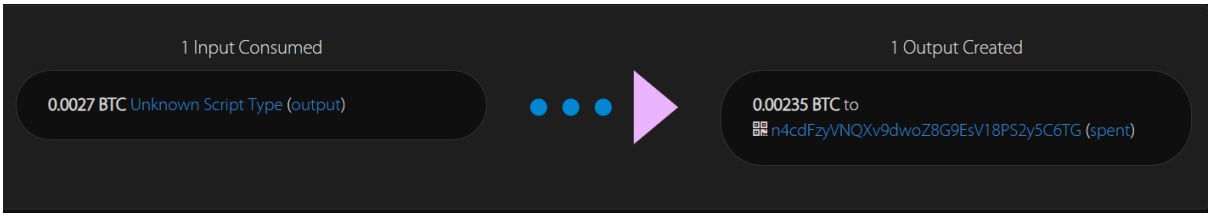
## دریافت

منبع پول برای انجام تراکنش

**47cf9cab1c18ca0883b7c851e8d7d8ced674fae365d86d83396fe9a6d370db1d**



**تراکنش 2)** سه آدرس جدید تولید کنید و مشخصات آن را در گزارش ذکر کنید. تراکنشی ایجاد کنید که یک ورودی و یک خروجی داشته باشد که خروجی آن از نوع MS2P یا Multisig بوده و توسط ۲ نفر از این ۳ آدرس آن قابل خرج شدن باشد. در تراکنشی دیگر این خروجی را خرج کرده و پول آن را به آدرس اصلی خود بازگردانید.

---

سه آدرس جدید با قطعه کد زیر تولید شده اند:

```
key_pairs = []

for _ in range(3):
    private_key = generate_private_key()
    public_key = private_key_to_public_key(private_key)
    address = public_key_to_address(public_key)
    wif_key = private_key_to_wif(private_key)

    key_pair = {
        "address": address,
        "wif_key": wif_key
    }
    key_pairs.append(key_pair)

for key_pair in key_pairs:
    print("Address:", key_pair["address"])
    print("WIF Key:", key_pair["wif_key"])
    print()
```

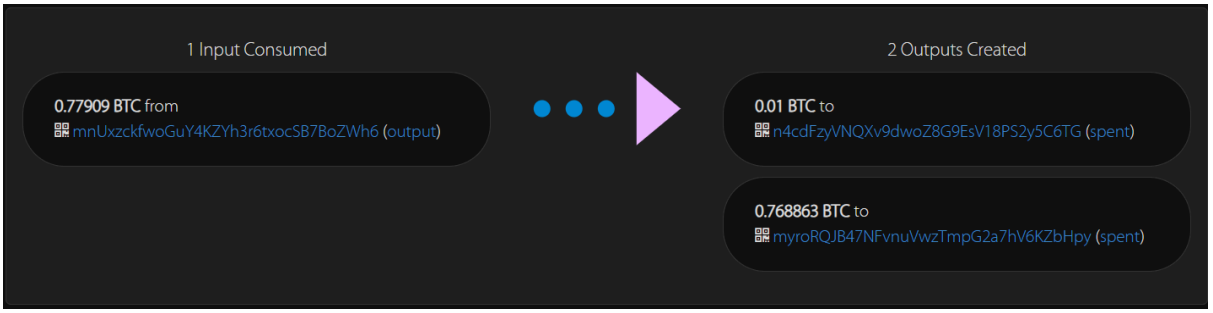
(variable) address: str

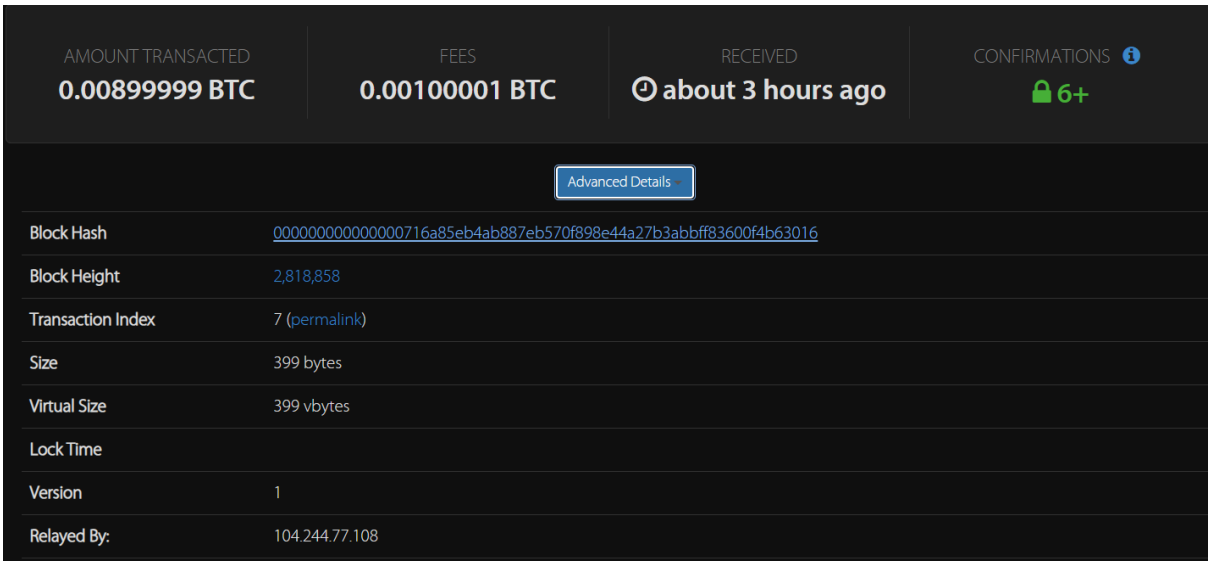
```
Address: mhAejg26Rvez6oQuJKr112w3vjrF97BvDX
WIF Key: 93Q1MV3TbAeBkWgDDwqkzewfiqtxH9vsguq7kjdDGqzeBLDXftE

Address: n2GWFEyXWVYvVWfTKKUPXLJVNHCFeMCLWo
WIF Key: 92EL9fuukGsyYFfyfbofWUXiiNW9h4Db13sS15RXiW4EPpRCzec

Address: mg6Hfu1yW9AXUz8KhxG2UJYiq3kK41NmvB
WIF Key: 92e5FncVApKtZHnPqBEtHkcU3LgW2DxfVTHZskCdfgcb86XubPo
```

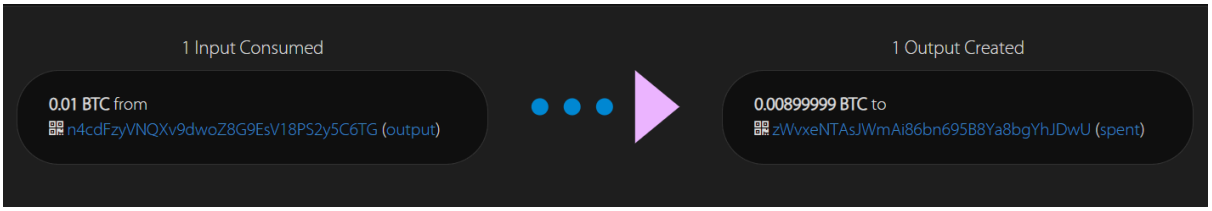
**ارسال**

برای انجام این بخش، به مقدار 0.009 کوین پول موجود در حساب آدرس که از تراکنش زیر حاصل شده، استفاده می‌کنیم.

**568356290e0c63de902fde8001ef92a01a3783b6faa3aeb17f401b1a8fce4925**

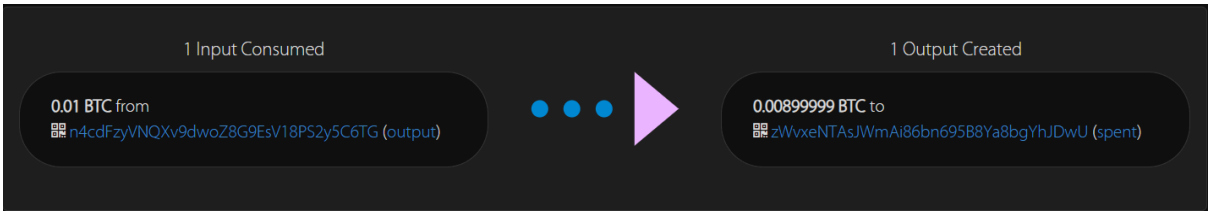



دریافت

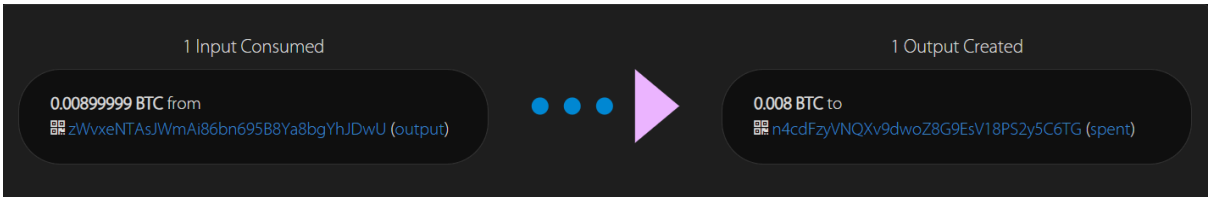
**2240fcdeae21a9aa593d0be9046470b254520bdfa2450733bb218a91ec12bf87**



| AMOUNT TRANSACTED | FEES | RECEIVED | CONFIRMATIONS |
| --- | --- | --- | --- |
| 0.008 BTC | 0.00099999 BTC | about 2 hours ago | 6+ |

Advanced Details

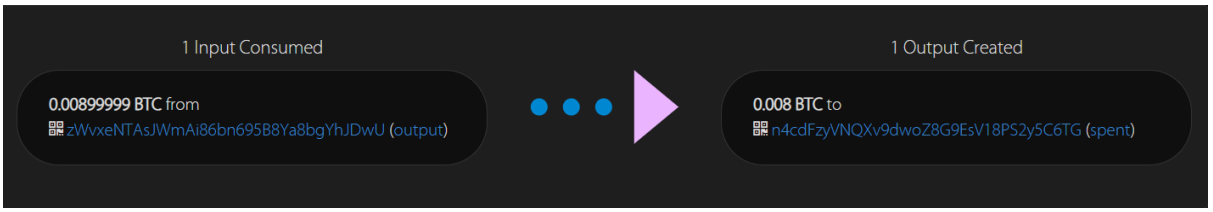
| | |
| --- | --- |
| Block Hash | 000000000000000b11c52eb5800a3b365f89ff61fd9f19599440cc1fffe1d5d8 |
| Block Height | 2,818,869 |
| Transaction Index | 22 (permalink) |
| Size | 230 bytes |
| Virtual Size | 230 vbytes |
| Lock Time | |
| Version | 1 |
| Relayed By: | 104.244.77.108 |

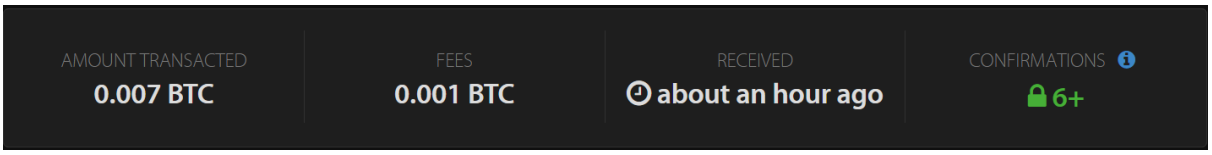


## تراکنش 3)

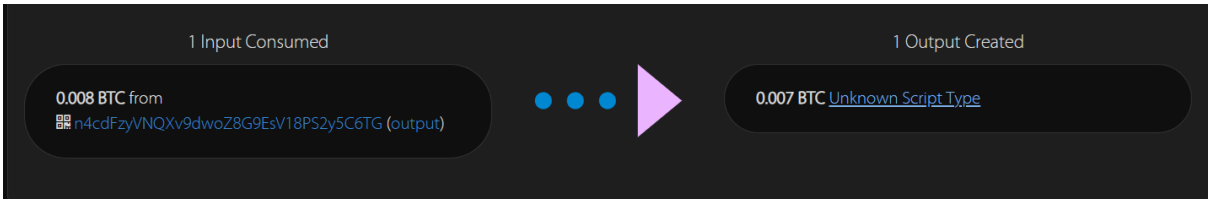
منبع پول برای انجام تراکنش اول

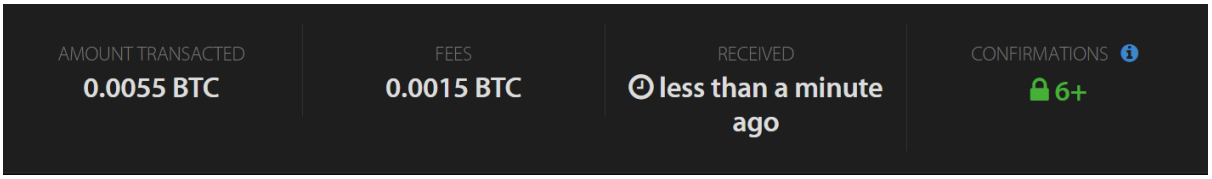
**5dea59163ecc7e809af114a541e2fe247985c110878345fe2c1c5d2a074f43d8**



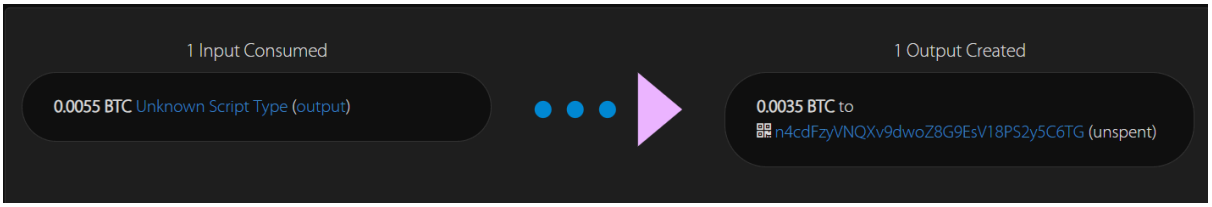




1. فراهم کردن سال تولد و سال فعلی و داشتن صلاحیت سن بزرگتر از 18 سال




2. فراهم کردن یک رمز عبور از قبل تعیین شده (شماره دانشجویی)




**Click here**

# بخش دوم: راه اندازی نود لوکال اتریوم

## گام اول: نصب geth

در این بخش، به کمک دستورهای داده شده در صورت پروژه، به نصب کتابخانه‌های مربوطه می‌پردازیم:

```
mahdi@LAPTOP-TUFMM2MT:~/mhr$ sudo add-apt-repository -y ppa:ethereum/ethereu
m
Repository: 'deb https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu/ jammy main'
More info: https://launchpad.net/~ethereum/+archive/ubuntu/ethereum
Adding repository.
Found existing deb entry in /etc/apt/sources.list.d/ethereum-ubuntu-ethereum-jammy.list
Adding deb entry to /etc/apt/sources.list.d/ethereum-ubuntu-ethereum-jammy.list
Found existing deb-src entry in /etc/apt/sources.list.d/ethereum-ubuntu-ethereum-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/ethereum-ubuntu-ethereum-jammy.list
Adding key to /etc/apt/trusted.gpg.d/ethereum-ubuntu-ethereum.gpg with fingerprint 2A518C819BE37D2C2031944D1C52189C923F6CA9
Hit:1 https://baltocdn.com/helm/stable/debian all InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:6 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
mahdi@LAPTOP-TUFMM2MT:~/mhr$ sudo apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 https://baltocdn.com/helm/stable/debian all InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:4 https://ppa.launchpadcontent.net/ethereum/ethereum/ubuntu jammy InRelease
Hit:5 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:6 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
mahdi@LAPTOP-TUFMM2MT:~/mhr$ sudo apt-get install ethereum
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ethereum is already the newest version (1.14.3+build29843+jammy).
0 upgraded, 0 newly installed, 0 to remove and 81 not upgraded.
mahdi@LAPTOP-TUFMM2MT:~/mhr$ mkdir node01 node02 node03
mahdi@LAPTOP-TUFMM2MT:~/mhr$
```

## گام دوم: کانفیگ کردن مشخصات بلوک اولیه 1

فایلی به نام genesis.json با محتوای زیر می‌سازیم.

```
mahdi@LAPTOP-TUFMM2MT:~/mhr$ ls -la
total 20
drwxrwxr-x  5 mahdi mahdi 4096 May 28 16:56 .
drwxr-x--- 34 mahdi mahdi 4096 May 28 16:54 ..
drwxrwxr-x  2 mahdi mahdi 4096 May 28 16:56 node01
drwxrwxr-x  2 mahdi mahdi 4096 May 28 16:56 node02
drwxrwxr-x  2 mahdi mahdi 4096 May 28 16:56 node03
mahdi@LAPTOP-TUFMM2MT:~/mhr$ touch genesis.json
mahdi@LAPTOP-TUFMM2MT:~/mhr$ ls -la
total 20
drwxrwxr-x  5 mahdi mahdi 4096 May 28 16:56 .
drwxr-x--- 34 mahdi mahdi 4096 May 28 16:54 ..
-rw-rw-r--  1 mahdi mahdi    0 May 28 16:56 genesis.json
drwxrwxr-x  2 mahdi mahdi 4096 May 28 16:56 node01
drwxrwxr-x  2 mahdi mahdi 4096 May 28 16:56 node02
drwxrwxr-x  2 mahdi mahdi 4096 May 28 16:56 node03
mahdi@LAPTOP-TUFMM2MT:~/mhr$ nano genesis.json
mahdi@LAPTOP-TUFMM2MT:~/mhr$ cat genesis.json
{
    "config":{
        "chainId":15,
        "homesteadBlock":0,
        "eip155Block":0,
        "eip158Block":0,
        "eip150Block":0
    },
    "difficulty":"400000",
    "gasLimit":"2100000",
    "alloc":{
    "0xB3FCd35A4D2Dd4623C44a9a0ACd2dFa568703975":{"balance":"1000000000810100216"},
    "0xf7C534F04DBA15Fd847B4Bea2678CE9aAa741DD8":{"balance":"2000000000810100216"},
    "0xaD915803bBe396E3eF9B3dcf6b38383f0f397F24":{"balance":"1500000000810100216"}
    }
}
```

## گام سوم: ایجاد 3 نود لوکال اتریوم

با دستور json فوق، سه نود را initialize کردیم و در مراحل بعدی، آدرس مربوط به هرکدام را آپدیت می‌کنیم.

## گام چهارم: ایجاد اکانت و شروع به کار گره ها | Initialize کردن نودها و اجرای آن

ابتدا، اکانتی برای هر نود ساخته و برای هرکدام پسورد منحصر به فردی انتخاب می‌کنیم. در این مرحله به ازای هر نود یک آدرس منحصر به فرد به دست می‌آید که در فایل genesis اشاره‌شده، وارد می‌کنیم.
سپس نودها را به کمک دستور init، آماده اجرا می‌کنیم.

```
mahdi@LAPTOP-TUFMM2MT:~/mhr$ geth --datadir "./node01/" account new
INFO [05-28|16:57:15.522] Maximum peer count                       ETH=50 LES=0 total=50
INFO [05-28|16:57:15.522] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
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:   0xB3FCd35A4D2Dd4623C44a9a0ACd2dFa568703975
Path of the secret key file: node01/keystore/UTC--2024-05-28T13-27-20.709327610Z--b3fcd35a4d2dd4623c44a9a0acd2dfa568703975

- 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!

mahdi@LAPTOP-TUFMM2MT:~/mhr$ geth --datadir "./node01" init ./genesis.json
INFO [05-28|17:01:40.949] Maximum peer count                       ETH=50 LES=0 total=50
INFO [05-28|17:01:40.950] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [05-28|17:01:40.953] Set global gas cap                       cap=50,000,000
INFO [05-28|17:01:40.954] Using leveldb as the backing database
INFO [05-28|17:01:40.954] Allocated cache and file handles         database=/home/mahdi/mhr/node01/geth/chaindata cache=16.00MiB handles=16
INFO [05-28|17:01:40.964] Using LevelDB as the backing database
INFO [05-28|17:01:40.982] Opened ancient database                  database=/home/mahdi/mhr/node01/geth/chaindata/ancient/chain readonly=false
INFO [05-28|17:01:40.982] Writing custom genesis block
INFO [05-28|17:01:40.983] Persisted trie from memory database      nodes=4 size=585.00B time="33.538µs" gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesiz
e=0.00B
INFO [05-28|17:01:40.985] Successfully wrote genesis state         database=chaindata                          hash=f9f3c1..c7d824
INFO [05-28|17:01:40.985] Using leveldb as the backing database
INFO [05-28|17:01:40.985] Allocated cache and file handles         database=/home/mahdi/mhr/node01/geth/lightchaindata cache=16.00MiB handles=16
INFO [05-28|17:01:40.992] Using LevelDB as the backing database
INFO [05-28|17:01:41.014] Opened ancient database                  database=/home/mahdi/mhr/node01/geth/lightchaindata/ancient/chain readonly=false
INFO [05-28|17:01:41.014] Writing custom genesis block
INFO [05-28|17:01:41.015] Persisted trie from memory database      nodes=4 size=585.00B time="502.327µs" gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesi
ze=0.00B
INFO [05-28|17:01:41.016] Successfully wrote genesis state         database=lightchaindata                     hash=f9f3c1..c7d824
```

---

```
mahdi@LAPTOP-TUFMM2MT:~/mhr$ geth --datadir "./node03/" account new
INFO [05-28|17:00:04.168] Maximum peer count                       ETH=50 LES=0 total=50
INFO [05-28|17:00:04.168] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
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:   0xaD915803bBe396E3eF9B3dcf6b38383f0f397F24
Path of the secret key file: node03/keystore/UTC--2024-05-28T13-30-08.402182031Z--ad915803bbe396e3ef9b3dcf6b38383f0f397f24

- 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!

mahdi@LAPTOP-TUFMM2MT:~/mhr$ geth --datadir "./node03" init ./genesis.json
INFO [05-28|17:01:47.181] Maximum peer count                       ETH=50 LES=0 total=50
INFO [05-28|17:01:47.182] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [05-28|17:01:47.185] Set global gas cap                       cap=50,000,000
INFO [05-28|17:01:47.186] Using leveldb as the backing database
INFO [05-28|17:01:47.186] Allocated cache and file handles         database=/home/mahdi/mhr/node03/geth/chaindata cache=16.00MiB handles=16
INFO [05-28|17:01:47.194] Using LevelDB as the backing database
INFO [05-28|17:01:47.212] Opened ancient database                  database=/home/mahdi/mhr/node03/geth/chaindata/ancient/chain readonly=false
INFO [05-28|17:01:47.212] Writing custom genesis block
INFO [05-28|17:01:47.213] Persisted trie from memory database      nodes=4 size=585.00B time="56.158µs" gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesiz
e=0.00B
INFO [05-28|17:01:47.215] Successfully wrote genesis state         database=chaindata                          hash=f9f3c1..c7d824
INFO [05-28|17:01:47.215] Using leveldb as the backing database
INFO [05-28|17:01:47.215] Allocated cache and file handles         database=/home/mahdi/mhr/node03/geth/lightchaindata cache=16.00MiB handles=16
INFO [05-28|17:01:47.223] Using LevelDB as the backing database
INFO [05-28|17:01:47.241] Opened ancient database                  database=/home/mahdi/mhr/node03/geth/lightchaindata/ancient/chain readonly=false
INFO [05-28|17:01:47.241] Writing custom genesis block
INFO [05-28|17:01:47.242] Persisted trie from memory database      nodes=4 size=585.00B time="456.405µs" gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesi
ze=0.00B
INFO [05-28|17:01:47.244] Successfully wrote genesis state         database=lightchaindata                     hash=f9f3c1..c7d824
mahdi@LAPTOP-TUFMM2MT:~/mhr$
```

---

```
mahdi@LAPTOP-TUFMM2MT:~/mhr$ geth --datadir "./node02/" account new
INFO [05-28|16:59:40.208] Maximum peer count                       ETH=50 LES=0 total=50
INFO [05-28|16:59:40.208] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
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:   0xf7C534F04DBA15Fd847B4Bea2678CE9aAa741DD8
Path of the secret key file: node02/keystore/UTC--2024-05-28T13-29-44.730237281Z--f7c534f04dba15fd847b4bea2678ce9aaa741dd8

- 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!

mahdi@LAPTOP-TUFMM2MT:~/mhr$ geth --datadir "./node02" init ./genesis.json
INFO [05-28|17:01:43.890] Maximum peer count                       ETH=50 LES=0 total=50
INFO [05-28|17:01:43.891] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [05-28|17:01:43.896] Set global gas cap                       cap=50,000,000
INFO [05-28|17:01:43.897] Using leveldb as the backing database
INFO [05-28|17:01:43.897] Allocated cache and file handles         database=/home/mahdi/mhr/node02/geth/chaindata cache=16.00MiB handles=16
INFO [05-28|17:01:43.906] Using LevelDB as the backing database
INFO [05-28|17:01:43.926] Opened ancient database                  database=/home/mahdi/mhr/node02/geth/chaindata/ancient/chain readonly=false
INFO [05-28|17:01:43.926] Writing custom genesis block
INFO [05-28|17:01:43.926] Persisted trie from memory database      nodes=4 size=585.00B time="43.54µs" gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize
=0.00B
INFO [05-28|17:01:43.928] Successfully wrote genesis state         database=chaindata                              hash=f9f3c1..c7d824
INFO [05-28|17:01:43.928] Using leveldb as the backing database
INFO [05-28|17:01:43.928] Allocated cache and file handles         database=/home/mahdi/mhr/node02/geth/lightchaindata cache=16.00MiB handles=16
INFO [05-28|17:01:43.936] Using LevelDB as the backing database
INFO [05-28|17:01:43.954] Opened ancient database                  database=/home/mahdi/mhr/node02/geth/lightchaindata/ancient/chain readonly=false
INFO [05-28|17:01:43.954] Writing custom genesis block
INFO [05-28|17:01:43.955] Persisted trie from memory database      nodes=4 size=585.00B time="623.166µs" gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesi
ze=0.00B
INFO [05-28|17:01:43.957] Successfully wrote genesis state         database=lightchaindata                         hash=f9f3c1..c7d824
mahdi@LAPTOP-TUFMM2MT:~/mhr$
```

نودها را یکی پس از دیگری استارت می‌کنیم.

```
mahdi@LAPTOP-TUFMM2MT:~/mhr$ geth --identity "node01" --http --http.port "8004" --authrpc.port "8556" --http.corsdomain "*" --datadir "./node01/" --port "30
304" --nodiscover --http.api "db,eth,net,web3,personal,miner,admin" --networkid 1900 --nat "any" --allow-insecure-unlock --ipcdisable
INFO [05-28|17:09:56.095] Maximum peer count                       ETH=50 LES=0 total=50
INFO [05-28|17:09:56.096] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [05-28|17:09:56.098] Set global gas cap                       cap=50,000,000
INFO [05-28|17:09:56.099] Allocated trie memory caches             clean=154.00MiB dirty=256.00MiB
INFO [05-28|17:09:56.099] Using leveldb as the backing database
INFO [05-28|17:09:56.099] Allocated cache and file handles         database=/home/mahdi/mhr/node01/geth/chaindata cache=512.00MiB handles=524,288
INFO [05-28|17:09:56.116] Using LevelDB as the backing database
INFO [05-28|17:09:56.119] Opened ancient database                  database=/home/mahdi/mhr/node01/geth/chaindata/ancient/chain readonly=false
INFO [05-28|17:09:56.120] Disk storage enabled for ethash caches   dir=/home/mahdi/mhr/node01/geth/ethash count=3
INFO [05-28|17:09:56.120] Disk storage enabled for ethash DAGs     dir=/home/mahdi/.ethash                count=2
INFO [05-28|17:09:56.120] Initialising Ethereum protocol           network=1900 dbversion=8
INFO [05-28|17:09:56.120]
INFO [05-28|17:09:56.120] ---------------------------------------------------------------------------------------------------------------------------------
----------------------
INFO [05-28|17:09:56.120] Chain ID:  15 (unknown)
INFO [05-28|17:09:56.120] Consensus: unknown
INFO [05-28|17:09:56.120]
INFO [05-28|17:09:56.120] Pre-Merge hard forks (block based):
INFO [05-28|17:09:56.120]  - Homestead:                   #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrad
es/homestead.md)
INFO [05-28|17:09:56.120]  - Tangerine Whistle (EIP 150): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrad
es/tangerine-whistle.md)
INFO [05-28|17:09:56.121]  - Spurious Dragon/1 (EIP 155): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrad
es/spurious-dragon.md)
INFO [05-28|17:09:56.121]  - Spurious Dragon/2 (EIP 158): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrad
es/spurious-dragon.md)
INFO [05-28|17:09:56.121]  - Byzantium:                   #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
byzantium.md)
INFO [05-28|17:09:56.121]  - Constantinople:              #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
constantinople.md)
INFO [05-28|17:09:56.121]  - Petersburg:                  #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
petersburg.md)
INFO [05-28|17:09:56.121]  - Istanbul:                    #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
istanbul.md)
INFO [05-28|17:09:56.121]  - Berlin:                      #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
berlin.md)
INFO [05-28|17:09:56.121]  - London:                      #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
london.md)
INFO [05-28|17:09:56.121]
INFO [05-28|17:09:56.121] The Merge is not yet available for this network!
```

```
mahdi@LAPTOP-TUFMM2MT:~/mhr$ geth --identity "node02" --http --http.port "8005" --authrpc.port "8557" --http.corsdomain "*" --datadir "./node02/" --port "30
305" --nodiscover --http.api "db,eth,net,web3,personal,miner,admin" --networkid 1900 --nat "any" --allow-insecure-unlock --ipcdisable
INFO [05-28|17:10:07.641] Maximum peer count                       ETH=50 LES=0 total=50
INFO [05-28|17:10:07.643] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [05-28|17:10:07.647] Set global gas cap                       cap=50,000,000
INFO [05-28|17:10:07.648] Allocated trie memory caches             clean=154.00MiB dirty=256.00MiB
INFO [05-28|17:10:07.648] Using leveldb as the backing database
INFO [05-28|17:10:07.648] Allocated cache and file handles         database=/home/mahdi/mhr/node02/geth/chaindata cache=512.00MiB handles=524,288
INFO [05-28|17:10:07.671] Using LevelDB as the backing database
INFO [05-28|17:10:07.682] Opened ancient database                  database=/home/mahdi/mhr/node02/geth/chaindata/ancient/chain readonly=false
INFO [05-28|17:10:07.682] Disk storage enabled for ethash caches   dir=/home/mahdi/mhr/node02/geth/ethash count=3
INFO [05-28|17:10:07.682] Disk storage enabled for ethash DAGs     dir=/home/mahdi/.ethash                count=2
INFO [05-28|17:10:07.682] Initialising Ethereum protocol           network=1900 dbversion=<nil>
INFO [05-28|17:10:07.683]
INFO [05-28|17:10:07.684] ------------------------------------------------------------------------------------------------------------------------------------
----------------------
INFO [05-28|17:10:07.684] Chain ID:  15 (unknown)
INFO [05-28|17:10:07.684] Consensus: unknown
INFO [05-28|17:10:07.684]
INFO [05-28|17:10:07.684] Pre-Merge hard forks (block based):
INFO [05-28|17:10:07.684]  - Homestead:                   #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrad
es/homestead.md)
INFO [05-28|17:10:07.684]  - Tangerine Whistle (EIP 150): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrad
es/tangerine-whistle.md)
INFO [05-28|17:10:07.684]  - Spurious Dragon/1 (EIP 155): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrad
es/spurious-dragon.md)
INFO [05-28|17:10:07.684]  - Spurious Dragon/2 (EIP 158): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrad
es/spurious-dragon.md)
INFO [05-28|17:10:07.684]  - Byzantium:                   #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
byzantium.md)
INFO [05-28|17:10:07.684]  - Constantinople:              #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
constantinople.md)
INFO [05-28|17:10:07.684]  - Petersburg:                  #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
petersburg.md)
INFO [05-28|17:10:07.684]  - Istanbul:                    #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
istanbul.md)
INFO [05-28|17:10:07.684]  - Berlin:                      #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
berlin.md)
INFO [05-28|17:10:07.684]  - London:                      #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
london.md)
INFO [05-28|17:10:07.684]
INFO [05-28|17:10:07.684] The Merge is not yet available for this network!
```

---

```
mahdi@LAPTOP-TUFMM2MT:~/mhr$ geth --identity "node03" --http --http.port "8006" --authrpc.port "8558" --http.corsdomain "*" --datadir "./node03/" --port "30
36" --nodiscover --http.api "db,eth,net,web3,personal,miner,admin" --networkid 1900 --nat "any" --allow-insecure-unlock --ipcdisable
INFO [05-28|17:10:17.838] Maximum peer count                       ETH=50 LES=0 total=50
INFO [05-28|17:10:17.839] Smartcard socket not found, disabling    err="stat /run/pcscd/pcscd.comm: no such file or directory"
INFO [05-28|17:10:17.842] Set global gas cap                       cap=50,000,000
INFO [05-28|17:10:17.843] Allocated trie memory caches             clean=154.00MiB dirty=256.00MiB
INFO [05-28|17:10:17.843] Using leveldb as the backing database
INFO [05-28|17:10:17.843] Allocated cache and file handles         database=/home/mahdi/mhr/node03/geth/chaindata cache=512.00MiB handles=524,288
INFO [05-28|17:10:17.862] Using LevelDB as the backing database
INFO [05-28|17:10:17.871] Opened ancient database                  database=/home/mahdi/mhr/node03/geth/chaindata/ancient/chain readonly=false
INFO [05-28|17:10:17.873] Disk storage enabled for ethash caches   dir=/home/mahdi/mhr/node03/geth/ethash count=3
INFO [05-28|17:10:17.873] Disk storage enabled for ethash DAGs     dir=/home/mahdi/.ethash                count=2
INFO [05-28|17:10:17.873] Initialising Ethereum protocol           network=1900 dbversion=<nil>
INFO [05-28|17:10:17.874]
INFO [05-28|17:10:17.874] ------------------------------------------------------------------------------------------------------------------------------------
----------------------
INFO [05-28|17:10:17.874] Chain ID:  15 (unknown)
INFO [05-28|17:10:17.874] Consensus: unknown
INFO [05-28|17:10:17.874]
INFO [05-28|17:10:17.874] Pre-Merge hard forks (block based):
INFO [05-28|17:10:17.874]  - Homestead:                   #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrad
es/homestead.md)
INFO [05-28|17:10:17.874]  - Tangerine Whistle (EIP 150): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrad
es/tangerine-whistle.md)
INFO [05-28|17:10:17.874]  - Spurious Dragon/1 (EIP 155): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrad
es/spurious-dragon.md)
INFO [05-28|17:10:17.874]  - Spurious Dragon/2 (EIP 158): #0        (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrad
es/spurious-dragon.md)
INFO [05-28|17:10:17.874]  - Byzantium:                   #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
byzantium.md)
INFO [05-28|17:10:17.874]  - Constantinople:              #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
constantinople.md)
INFO [05-28|17:10:17.874]  - Petersburg:                  #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
petersburg.md)
INFO [05-28|17:10:17.874]  - Istanbul:                    #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
istanbul.md)
INFO [05-28|17:10:17.874]  - Berlin:                      #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
berlin.md)
INFO [05-28|17:10:17.874]  - London:                      #<nil> (https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/
london.md)
INFO [05-28|17:10:17.874]
INFO [05-28|17:10:17.874] The Merge is not yet available for this network!
```

دقت کنید که پورت و آی‌پی مختص هر نود به شرح زیر می‌باشد:

| node | http port | auth rpc port | port |
|---|---|---|---|
| **node01** | 8004 | 8556 | 30304 |
| **node02** | 8005 | 8557 | 30305 |
| **node03** | 8006 | 8558 | 3036 |

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

به کمک دستور زیر، با سه کلاینت به هرکدام از نودها به صورت لوکال (127.0.0.1) متصل می‌شویم.

```
mahdi@LAPTOP-TUFMM2MT:~/mhr$ geth attach http://127.0.0.1:8005

WARN [05-28|17:11:55.628] Enabling deprecated personal namespace
Welcome to the Geth JavaScript console!

instance: Geth/node02/v1.11.6-stable-ea9e62ca/linux-amd64/go1.20.3
at block: 0 (Thu Jan 01 1970 03:30:00 GMT+0330 (+0330))
 datadir: /home/mahdi/mhr/node02
 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
>
> admin.nodeInfo
{
  enode: "enode://491d1d6375623d9855962da315a2617af14d8f377de38e9244fb88926889ffe26ec01acee4268b2767090f5e24d1269d08a32636a692ec434cb74b6d1744d2c3@127.0.0.1
:30305?discport=0",
  enr: "enr:-Jy4QC_M5KuDTfN0khA5OFmutJJIe8X-C4CxonsYnH2z8oL8W5g6RDmus3KvnfDAstXLWDqqjpNfUwRdTK8brKjmP_OGAY-_bf2Sg2V0aMfGhIbrunKAgmlkgnY0gmlwhH8AAAGJc2VjcDI1
NmsxoQNJHR1jdWI9mFWWLaMVomF68U2PN33jjpJE-4iSaIn_4oRzbmFwwIN0Y3CCdmE",
  id: "d39661ed6ff6fa689c1695cbffe18c51e578799a67624d777a50300fa6b197d2",
  ip: "127.0.0.1",
  listenAddr: "[::]:30305",
  name: "Geth/node02/v1.11.6-stable-ea9e62ca/linux-amd64/go1.20.3",
  ports: {
    discovery: 0,
    listener: 30305
  },
  protocols: {
    eth: {
      config: {
        chainId: 15,
        eip150Block: 0,
        eip155Block: 0,
        eip158Block: 0,
        homesteadBlock: 0
      },
      difficulty: 400000,
      genesis: "0xf9f3c16b8db05ff882256bc730f3ad8d370fce8e2a8774f38e6bc0dd05c7d824",
      head: "0xf9f3c16b8db05ff882256bc730f3ad8d370fce8e2a8774f38e6bc0dd05c7d824",
      network: 1900
    },
    snap: {}
  }
```

---

```
mahdi@LAPTOP-TUFMM2MT:~/mhr$ geth attach http://127.0.0.1:8004
WARN [05-28|17:11:20.257] Enabling deprecated personal namespace
Welcome to the Geth JavaScript console!

instance: Geth/node01/v1.11.6-stable-ea9e62ca/linux-amd64/go1.20.3
at block: 0 (Thu Jan 01 1970 03:30:00 GMT+0330 (+0330))
 datadir: /home/mahdi/mhr/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
> admin.nodeInfo
{
  enode: "enode://0c3190751f846ce21e1221786e750fc7ea73768c376a23d10e87310952170b9421f2c47e43c22d66b00a18c113ad9037227befba197423a08b73798717b648f5@127.0.0.1
:30304?discport=0",
  enr: "enr:-Jy4QO9Mp_G5aAIwipoZJMmoznl9p6N7HTe18_cpgft-sM7EKSiYY_ChE5QPyHTrkmaTi3mxoJgzK_fESuzZIldSN6mGAY-_bdBog2V0aMfGhIbrunKAgmlkgnY0gmlwhH8AAAGJc2VjcDI1
NmsxoQMMMZB1H4Rs4h4SIXhudQ_H6nN2jDdqI9EOhzEJUhcLlIRzbmFwwIN0Y3CCdmA",
  id: "4635fd91e34c47bbe7d055158f2e2ae3174bcf1175a703dc26e7302f474d37ee",
  ip: "127.0.0.1",
  listenAddr: "[::]:30304",
  name: "Geth/node01/v1.11.6-stable-ea9e62ca/linux-amd64/go1.20.3",
  ports: {
    discovery: 0,
    listener: 30304
  },
  protocols: {
    eth: {
      config: {
        chainId: 15,
        eip150Block: 0,
        eip155Block: 0,
        eip158Block: 0,
        homesteadBlock: 0
      },
      difficulty: 400000,
      genesis: "0xf9f3c16b8db05ff882256bc730f3ad8d370fce8e2a8774f38e6bc0dd05c7d824",
      head: "0xf9f3c16b8db05ff882256bc730f3ad8d370fce8e2a8774f38e6bc0dd05c7d824",
      network: 1900
    },
    snap: {}
  }
}
```

---

```
mahdi@LAPTOP-TUFMM2MT:~/mhr$ geth attach http://127.0.0.1:8006
WARN [05-28|17:12:02.879] Enabling deprecated personal namespace
Welcome to the Geth JavaScript console!

instance: Geth/node03/v1.11.6-stable-ea9e62ca/linux-amd64/go1.20.3
at block: 0 (Thu Jan 01 1970 03:30:00 GMT+0330 (+0330))
 datadir: /home/mahdi/mhr/node03
 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
> admin.nodeInfo
{
  enode: "enode://4f525285b24b4ff17b3c8a41c1d1041eb966f7ed4a5f89d312f0c01424ee6fd04bfce9241e2cfd91d921324cb2dce102f96625d08b7c79db6c93effdae243a86@127.0.0.1
:3036?discport=0",
  enr: "enr:-Jy4QOjrfORfh4s13TjPhMTUI0hgYo_pYQXLMdQIP1ReInUed-kwqzMbkcTaI5IFbh2gMRiFnul8RR37ogvK1nm9-6CGAY-_biVdg2V0aMfGhIbrunKAgmlkgnY0gmlwhH8AAAGJc2VjcDI1
NmsxoQJPULkFsktP8Xs8ikHB0QQeuWb37UpfidMS8MAUJO5v0IRzbmFwwIN0Y3CCC9w",
  id: "ee0a656ce80f35e369141dfeb63ef1ae8451b6b238aa9e7493d625559ab59fbc",
  ip: "127.0.0.1",
  listenAddr: "[::]:3036",
  name: "Geth/node03/v1.11.6-stable-ea9e62ca/linux-amd64/go1.20.3",
  ports: {
    discovery: 0,
    listener: 3036
  },
  protocols: {
    eth: {
      config: {
        chainId: 15,
        eip150Block: 0,
        eip155Block: 0,
        eip158Block: 0,
        homesteadBlock: 0
      },
      difficulty: 400000,
      genesis: "0xf9f3c16b8db05ff882256bc730f3ad8d370fce8e2a8774f38e6bc0dd05c7d824",
      head: "0xf9f3c16b8db05ff882256bc730f3ad8d370fce8e2a8774f38e6bc0dd05c7d824",
      network: 1900
    },
    snap: {}
  }
}
```

پس از اتصال، به کمک دستور admin.nodeInfo، مقدار enode مربوط به هر نود را می‌گیریم.

## گام ششم:  اتصال نودها به یکدیگر

به کمک دستور admin.addPeer و قرار دادن enode مربوط به نودی که می‌خواهیم به آن وصل شویم، اتصال نودها صورت می‌گیرد. توجه کنید که نود شماره 2 را نود مرکزی قرار داده‌ایم. پس نود مرکزی را به دو نود دیگر متصل کرده و هر کدام از نودهای شماره یک و سه را یک‌بار به نود مرکزی وصل می‌کنیم.

```
> admin.addPeer("enode://491d1d6375623d9855962da315a2617af14d8f377de38e9244fb88926889ffe26ec01acee4268b2767090f5e24d1269d08a32636a692ec434cb74b6d1744d2c3@12
7.0.0.1:30305?discport=0")
true
```

اتصال نودهای شماره 1 و 3 به نود مرکزی:

```
> admin.addPeer("enode://491d1d6375623d9855962da315a2617af14d8f377de38e9244fb88926889ffe26ec01acee4268b2767090f5e24d1269d08a32636a692ec434cb74b6d1744d2c3@12
7.0.0.1:30305?discport=0")
true
> net.peerCount
1
> eth.accounts
["0xad915803bbe396e3ef9b3dcf6b38383f0f397f24"]
> eth.getBalance(eth.accounts[0])
>
>
```

---

```
> admin.addPeer("enode://491d1d6375623d9855962da315a2617af14d8f377de38e9244fb88926889ffe26ec01acee4268b2767090f5e24d1269d08a32636a692ec434cb74b6d1744d2c3@12
7.0.0.1:30305?discport=0")
true
> net.peerCount
1
> eth.accounts
["0xb3fcd35a4d2dd4623c44a9a0acd2dfa568703975"]
>
> eth.getBalance(eth.accounts[0])
100000000081010216
>
```

اتصال نود مرکزی (شماره 2) به دو نود دیگر:

```
> admin.addPeer("enode://0c3190751f846ce21e1221786e750fc7ea73768c376a23d10e87310952170b9421f2c47e43c22d66b00a18c113ad9037227befba197423a08b73798717b648f5@12
7.0.0.1:30304?discport=0")
true
>
> net.peerCount
1
> admin.addPeer("enode://4f525285b24b4ff17b3c8a41c1d1041eb966f7ed4a5f89d312f0c01424ee6fd04bfce9241e2cfd91d921324cb2dce102f96625d08b7c79db6c93effdae243a86@12
7.0.0.1:3036?discport=0")
true
> net.peerCount
2
> eth.accounts
["0xf7c534f04dba15fd847b4bea2678ce9aaa741dd8"]
> eth.getBalance(eth.accounts[0])
200000000081010216
```

در انتها با دستور net.peerCount از اتصال نودها به یکدیگر مطمئن می‌شویم.

توجه کنید که با دستور eth.accounts، اکانت‌های موجود در هر نود را مشاهده می‌کنیم که در تصاویر فوق مشخص است. این اکانت‌ها، قبلا در فایل genesis وارد شده‌اند.
با دستور eth.getBalance وا دادن اکانت مدنظر در هر نود، مقدار حساب آن اکانت به دست می‌آید.

## گام هفتم: ایجاد تراکنش

به کمک دستور زیر، با وارد کردن رمز عبور، اکانت مدنظر را برای انجام تراکنش باز می‌کنیم. (اکانت نود شماره یک) در ادامه با دستور eth.sendTransaction و قرار دادن آرگومان‌های مربوط به مبدا و مقصد تراکنش و مقدار کوین تراکنش، تراکنش را وارد شبکه می‌کنیم.

```
> personal.unlockAccount(eth.accounts[0])
Unlock account 0xb3fcd35a4d2dd4623c44a9a0acd2dfa568703975
Passphrase:
true
> eth.sendTransaction({from:eth.accounts[0], to:"0xad915803bbe396e3ef9b3dcf6b38383f0f397f24", value:1000})
"0x60019a717a52180a531e5d5e0de824d3188ab53e2923ad16cfa623361339c159"
>
```

## گام هشتم: ماین کردن یک بالک جهت قرار گیری تراکنش در بلاکچین

به کمک دستور miner.setEtherbase و وارد کردن اکانت مدنظر برای واریز جایزه ماینر، این جایزه را به نود ماینر یعنی نود شماره 2 می‌دهیم. با دستور miner.start، عملیات ماین کردن شروع می‌شود.
برای ماین کردن زمان حدودی 10 دقیقه مدنظر گرفته شده که در انتها، بلوک با موفقیت ماین می‌شود.

```
> miner.setEtherbase(eth.accounts[0])
true
> miner.start()
null
>
>
```

در ترمینال مربوط به استارت خوردن نود ماینر، لاگ‌های مربوط به عملیات ماینینگ را مشاهده می‌کنیم.

```
INFO [05-28|17:42:30.288] Commit new sealing work                  number=684 sealhash=ec0a7f..160e64 uncles=0 txs=0 gas=0     fees=0       elapsed=13.038ms
INFO [05-28|17:42:30.326] Successfully sealed new block            number=684 sealhash=ec0a7f..160e64 hash=0dae90..d60348 elapsed=50.982ms
INFO [05-28|17:42:30.326] "🔗 block reached canonical chain"        number=677 hash=77e7c9..524d4c
INFO [05-28|17:42:30.327] "🔨 mined potential block"                number=684 hash=0dae90..d60348
INFO [05-28|17:42:30.327] Commit new sealing work                  number=685 sealhash=301bf4..e2e135 uncles=0 txs=0 gas=0     fees=0       elapsed=1.001ms
INFO [05-28|17:42:30.327] Commit new sealing work                  number=685 sealhash=301bf4..e2e135 uncles=0 txs=0 gas=0     fees=0       elapsed=1.140ms
INFO [05-28|17:42:30.864] Successfully sealed new block            number=685 sealhash=301bf4..e2e135 hash=cba944..9e69bd elapsed=537.514ms
INFO [05-28|17:42:30.864] "🔗 block reached canonical chain"        number=678 hash=e160d7..c6c5ce
INFO [05-28|17:42:30.864] "🔨 mined potential block"                number=685 hash=cba944..9e69bd
INFO [05-28|17:42:30.864] Commit new sealing work                  number=686 sealhash=32614d..980825 uncles=0 txs=0 gas=0     fees=0       elapsed="127.814
µs"
INFO [05-28|17:42:30.864] Commit new sealing work                  number=686 sealhash=32614d..980825 uncles=0 txs=0 gas=0     fees=0       elapsed="255.257
µs"
INFO [05-28|17:42:33.822] Successfully sealed new block            number=686 sealhash=32614d..980825 hash=3313a0..5efaf3 elapsed=2.957s
INFO [05-28|17:42:33.822] "🔗 block reached canonical chain"        number=679 hash=d1dc0c..0b05b8
INFO [05-28|17:42:33.822] "🔨 mined potential block"                number=686 hash=3313a0..5efaf3
INFO [05-28|17:42:33.822] Commit new sealing work                  number=687 sealhash=8c80e9..30a182 uncles=0 txs=0 gas=0     fees=0       elapsed="159.573
µs"
INFO [05-28|17:42:33.822] Commit new sealing work                  number=687 sealhash=8c80e9..30a182 uncles=0 txs=0 gas=0     fees=0       elapsed="310.09µ
s"
INFO [05-28|17:42:33.976] Successfully sealed new block            number=687 sealhash=8c80e9..30a182 hash=0c2663..4ebe9e elapsed=154.292ms
INFO [05-28|17:42:33.976] "🔗 block reached canonical chain"        number=680 hash=764d85..b0de14
INFO [05-28|17:42:33.976] "🔨 mined potential block"                number=687 hash=0c2663..4ebe9e
INFO [05-28|17:42:33.976] Commit new sealing work                  number=688 sealhash=f6da4c..34007b uncles=0 txs=0 gas=0     fees=0       elapsed="145.982
µs"
INFO [05-28|17:42:33.977] Commit new sealing work                  number=688 sealhash=f6da4c..34007b uncles=0 txs=0 gas=0     fees=0       elapsed="278.693
µs"
INFO [05-28|17:42:34.032] Successfully sealed new block            number=688 sealhash=f6da4c..34007b hash=a562bc..0282be elapsed=55.085ms
INFO [05-28|17:42:34.032] "🔗 block reached canonical chain"        number=681 hash=faab3b..c3c6d3
INFO [05-28|17:42:34.032] "🔨 mined potential block"                number=688 hash=a562bc..0282be
INFO [05-28|17:42:34.032] Commit new sealing work                  number=689 sealhash=45fa4f..ef4900 uncles=0 txs=0 gas=0     fees=0       elapsed="166.857
µs"
INFO [05-28|17:42:34.032] Commit new sealing work                  number=689 sealhash=45fa4f..ef4900 uncles=0 txs=0 gas=0     fees=0       elapsed="288.423
µs"
INFO [05-28|17:42:36.923] Successfully sealed new block            number=689 sealhash=45fa4f..ef4900 hash=0375cf..ce1c78 elapsed=2.891s
INFO [05-28|17:42:36.923] "🔗 block reached canonical chain"        number=682 hash=e38823..f5e3f6
INFO [05-28|17:42:36.923] "🔨 mined potential block"                number=689 hash=0375cf..ce1c78
INFO [05-28|17:42:36.923] Commit new sealing work                  number=690 sealhash=40e667..bdfd9e uncles=0 txs=0 gas=0     fees=0       elapsed="262.187
µs"
INFO [05-28|17:42:36.924] Commit new sealing work                  number=690 sealhash=40e667..bdfd9e uncles=0 txs=0 gas=0     fees=0       elapsed="474.034
µs"
```

با دستور miner.stop، به عملیات ماینینگ خاتمه می‌دهیم.

با گرفتن مقدار balance در حساب نود ماینر، از تعلق جایزه استخراج بلوک اطمینان حاصل می‌کنیم.

```
> miner.setEtherbase(eth.accounts[0])
true
>
> miner.start()
null
> miner.stop()
null
> eth.getBalance(eth.accounts[0])
3.6820000210008101002 16e+21
>
```

با وارد شدن تراکنش در بلوک و سپس قرار گرفتن در زنجیره اصلی شبکه، مقدار پول انتقال یافته و با گرفتن balance مربوط به حساب اکانت انجام دهنده‌ی تراکنش (نود شماره یک)، اکانت مقصد تراکنش (نود شماره سه) و balance اکانت ماین کننده (نود شماره دو)، از انتقال وجه‌ها مطمئن می‌شویم.

همانطور که می‌بینیم، به حساب نود شماره سه، 1000 کوین وارد شده است:

```
> eth.getBalance(eth.accounts[0])
1500000000810100216
> eth.getBalance(eth.accounts[0])
1500000000810100216
> eth.getBalance(eth.accounts[0])
1500000000810101216
>
```

و از حساب نود شماره یک، هزار کوین کاسته شده است:

```
>
> eth.getBalance(eth.accounts[0])
1000000000810100216
>
> eth.getBalance(eth.accounts[0])
9999790000810099216
>
```

بخش

# سوال 1:

## کارکرد هر نود در یک شبکه

### 1. اعتبار سنجی تراکنش:

اعتبار سنجی تراکنش‌ها با بررسی اینکه از قوانین پروتکل شبکه پیروی می‌کنند یا خیر  انجام می‌شود. در این فرایند، بررسی صحت امضای دیجیتال، بررسی وجود balance مشخص شده در تراکنش در حساب فرستنده، بررسی انجام نشدن double-spend با ایجاد تراکنش صورت می‌گیرد.

### 2. اعتبار سنجی بلوک:

بلوک‌های تازه ماین شده توسط نودهای شبکه تایید می‌شوند. این تاییدیه بنابر پیروی از پروتکل اجماع شبکه صورت می‌گیرد. بخشی از آن شامل proof of work و proof of stake می‌شود.

### 3. انتشار اطلاعات شبکه (عدم وجود مرکزیت واحد):

نودها مسئول انتقال تراکنش‌ها و بلوک‌ها به بقیه نودهای شبکه هستند. یعنی در صورت تایید شدن، آن را به peer های خودش انتقال می‌دهد و نیاز به مرکزیت وجود ندارد.

### 4. محافظت و نگهداری از بلاک چین (عدم وجود مرکزیت واحد):

نودها یک کپی از بلاک چین مربوطه را که حکم public ledger همگی تراکنش‌ها می‌باشد، نگه‌داری می‌کنند.

### 5. امنیت شبکه:

با اعمال شدن پروتکل شبکه توسط نودها، امنیت شبکه و جلوگیری از انجام حملات به شبکه صورت می‌گیرد. حملات شبکه بلاک چین، تغییر پروتکل شبکه بوده و وظیفه اعمال پروتکل بر عهده نودهای شبکه می‌باشد.

## Full Node

یک نود کامل، نقش ستون فقرات در یک شبکه کریپتوکارنسی دارد. اساس شبکه Full Node های آن است. هر فول نود، به طور مستقل هر تراکنش را طبق قوانین و سیاست‌های پروتکل، verify می‌کنند. هر نود کامل، همه تاریخچه‌ی بلاک چین را در حافظه خود ذخیره دارد تا یکپارچگی و در دسترس بودن داده را تضمین کند. با بررسی قوانین پروتکل، تراکنش‌های verify شده را به اطلاع بقیه نودها می‌رسانند. بنابراین مسئول چک کردن برقراری پروتکل شبکه، همگی نودهای کامل هستند و با رد تراکنش‌ها و بلوک‌های فاقد صلاحیت، از اجماع بر روی آن‌ها جلوگیری می‌کنند.

## Light Node

یک لایت نود یا SPV NODE، نسبت به نودهای کامل، منابع کمتری دارند. فقط block header ها را دانلود می‌کنند و در حافظه خود نگه می‌دارند و در صورت نیاز به verify کردن، دیتای مربوط به آن تراکنش خاص را می‌گیرند. بالا آوردن light node به علت نیاز به حافظه کمتر ارزان‌تر و سریع‌تر از full node است و این دسته از نودها برای ارائه اطلاعات مربوط به تراکنش‌ها و بلوک‌ها به full node ها وابسته می‌شوند.

## سوال 2:

در این لاگ ها تیتر های زیر مشاهده می‌شود که به شرح هر کدام می‌پردازیم:

پیام پیدا شدن بلوک دارای پتانسیل برای قرارگیری در زنجیره بلاک چین:


[05-28|17:42:34.032]: Timestamp indicating when the mining process finished successfully.

number=688: The block number.

hash=a562bc..0282be: The has of the block.

در این پیام هش بلوک پیدا شده و شماره بلوک و زمان دقیق پیدا شدن آن نشان داده شده است.

---


این پیام نشان می‌دهد فرایند ماین‌کردن، از نو برای seal کردن بلوک بعدی شروع می‌شود.

---


نشان می‌دهد که بلوک، با موفقیت seal شده است. یعنی ماین یا استخراج شده است. در این پیام، شماره بلوک و هش مربوط به فرایند sealing نشان داده شده است.

---


این پیام نشان می‌دهد بلوک یافت شده با موفقیت در زنجیره اصلی بلاک چین قبول شده است.

در این لاگ هم شماره بلوک و هش مربوطه ذکر شده است.

---


این پیام نشان می‌دهد که گراف جهت‌دار بدون دور یا همان DAG مربوط به فرایند ماینینگ تولید شده است.

اطلاعاتی نظیر شماره epoch، درصد تکمیل و مدت زمان سپری (elapsed) شده برای رسیدن به زنجیره اصلی در این لاگ قابل مشاهده می‌باشد.

طبق برداشت از لاگ‌ها، فرایند استخراج کلیت زیر را شامل می‌شود:

1. مرحله **Mined Potential Block**

در این مرحله بلوکی که پتانسیل وارد شدن به بلاک چین را دارد، استخراج شده و هش و تمامی اطلاعات مربوط به آن فراهم شده است.

2. مرحله **Commit New Sealing Work**

در این مرحله بلوک استخراج شده با موفقیت commit شده و ماینر شروع به استخراج بلوک بعدی در بلاک چین (به دنبال بلوک قبلی) میکند.

3. مرحله **Successfully Sealed New Block**

در این مرحله بلوک جدید با موفقیت seal شده است و آماده وارد شدن به بلاک چین می‌باشد.

4. مرحله **Block Reached Canonical Chain**

در این مرحله بلوک استخراج شده در طی فرایند اجماع، قبول شده و در زنجیره اصلی وارد می‌شود.

5. مرحله **تولید DAG**

فرایند ماین کردن، یک فرایند دوره‌ای و استخراج کردن یک فایل DAG می‌باشد. این فایل استخراج شده، در الگوریتم Ethash proof-of-work مورد استفاده قرار می‌گیرد و برای گذر از این مرحله، لازم می‌باشد.

## سوال 3:

این فرایند شامل ایجاد یک زنجیره خصوصی از بلوک‌هاست که به صورت محلی استخراج شده‌اند و تا زمانی که ارتفاع آن با ارتفاع زنجیره اصلی یکسان نشود، منتشر نمی‌شود. با انتشار زنجیره روی شبکه، ادعای زنجیره اصلی بودن انجام می‌شود. در پروتکل شبکه اتریوم، از PoW و PoS استفاده می‌شود و چالش‌های مربوطه به شرح زیر است:

**Difficulty and Cumulative Work | PoW**

فرایند PoW در شبکه اتریوم شامل یک difficulty bomb است که استخراج بلوک‌های جدید را از اردر زمانی exponentially، سخت می‌کند. هدف از این پروتکل، تشویق تراکنش‌ها به سمت PoS و غیرممکن شدن استخراج تعداد زیادی بلوک در مدت زمان محدود روی یک زنجیره خصوصی می‌باشد.

شبکه اتریوم، طولانی‌ترین زنجیره شبکه را زنجیره اصلی در نظر می‌گیرد که معادل داشتن بیشتری کار تجمعی در زنجیره است (Cumulative work). برای رسیدن به این هدف با توجه به پروتکل‌های تعریف شده روی شبکه، نیاز به توان محاسباتی خیلی زیاد و عملا ناممکن است.

**Validator Consensus and Finality | PoS**

برای تایید شدن تراکنش، به یک سهم و کنترل قابل توجهی بر بخش بزرگی از اعتبارسنج‌های روی شبکه نیاز است.

وقتی بلوک‌ها در PoS نهایی بشوند، نمی‌توان آن‌ها را بدون اجماع کامل بازگرداند. بدیت شکل تضمین امنیتی روی شبکه در برابر حملات برقرار می‌شود.

**Network Acceptance**

حتی اگر توان محاسباتی ساخت این زنجیره را داشته باشیم و زنجیره را به شبکه منتشر کنیم، نودها طبق پروتکل با آن برخورد می‌کنند و احتمال به اجماع رسیدن روی این زنجیره جدید بسیار کم است. زنجیره اصلی لزوما باید از پروتکل پیروی کند.