**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

**ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

**Кафедра Інформаційної Безпеки**

# Комп’ютерний практикум №1

**з дисципліни**

**«Блокчейн та децентралізовані системи»**

**Виконали:**

**Студенти 5 курсу ФТІ**

**групи ФБ-41мн**

**Бондаренко О.Ю., Кригін Д.О.,**

**Шанідзе Д.Л.**

**Перевірила:**

**асистент**

**Селюх П.В.**

Київ 2025

**Комп’ютерний практикум №1**
**Розгортання систем Ethereum та криптовалют**

**Мета роботи:** Отримання навичок налаштування платформ виконання смарт контрактів та криптовалют.

**Завдання**: Провести налаштування системи Ethereum та виконати тестові операції в системі.

## Хід Роботи

Почнемо із вибору інструментів. Ми вирішили скористатися https://hub.docker.com/r/ethereum/client-go через простоту у використанні і підтримці контейнерів, до того ж їх легко розгортати і масштабовувати у різних оточеннях.

Перший блок у блокчейні - genesis block, визначимо його параметри у конфігураційному файлі:

genesis.json

```
{
   "config": {
      "chainId": 1337,
      "homesteadBlock": 0,
      "eip150Block": 0,
      "eip155Block": 0,
      "eip158Block": 0,
      "byzantiumBlock": 0,
      "constantinopleBlock": 0,
      "petersburgBlock": 0,
      "ethash": {}
   },
   "difficulty": "1",
   "gasLimit": "12000000",
   "alloc": {}
}
```

Створимо файл зі змінними оточення для подальшого їх використання у коді:

.env

```
NETWORK_ID=1337
ACCOUNT_PASSWORD=<REDACTED>
```

Створимо Dockerfile на основі ethereum/client-go.
Ініціалізуємо блокчейн з файлу генезис блоку і створимо обліковий запис із встановленим паролем.

Dockerfile:

```
FROM ethereum/client-go:v1.10.1

ARG ACCOUNT_PASSWORD

COPY genesis.json .

RUN geth init ./genesis.json \
    && rm -f ~/.ethereum/geth/nodekey \
    && echo ${ACCOUNT_PASSWORD} > ./password.txt \
    && geth account new --password ./password.txt \
    && rm -f ./password.txt

ENTRYPOINT ["geth"]
```

Далі маємо створити nodekey і enode для bootnode

```
./bootnode -genkey bootnode.key
```

```
cat bootnode.key
4ae3b4381e02ba395606c2eedaa6050e8b03a3ea98c2e015c7cccb103
4aa5c5d
```

```
./bootnode -nodekeyhex
4ae3b4381e02ba395606c2eedaa6050e8b03a3ea98c2e015c7cccb103
4aa5c5d -writeaddress
```

Output enode value:

```
6c0bdbf52eff691d706a8348e375c0218006b051f7886f7513ff11f65f541
10fffadf83b9f1c14ac49298f4dc0501c84b2a1615a4a1703347684084e
40b66a3d
```

Таким чином ми зможемо звертатися до нод за:

```
enode://6c0bdbf52eff691d706a8348e375c0218006b051f7886f7513ff1
1f65f54110fffadf83b9f1c14ac49298f4dc0501c84b2a1615a4a17033476
84084e40b66a3d@<IP>:<PORT>
```

Створимо одну bootnode і 3 інші ноди -майнери:

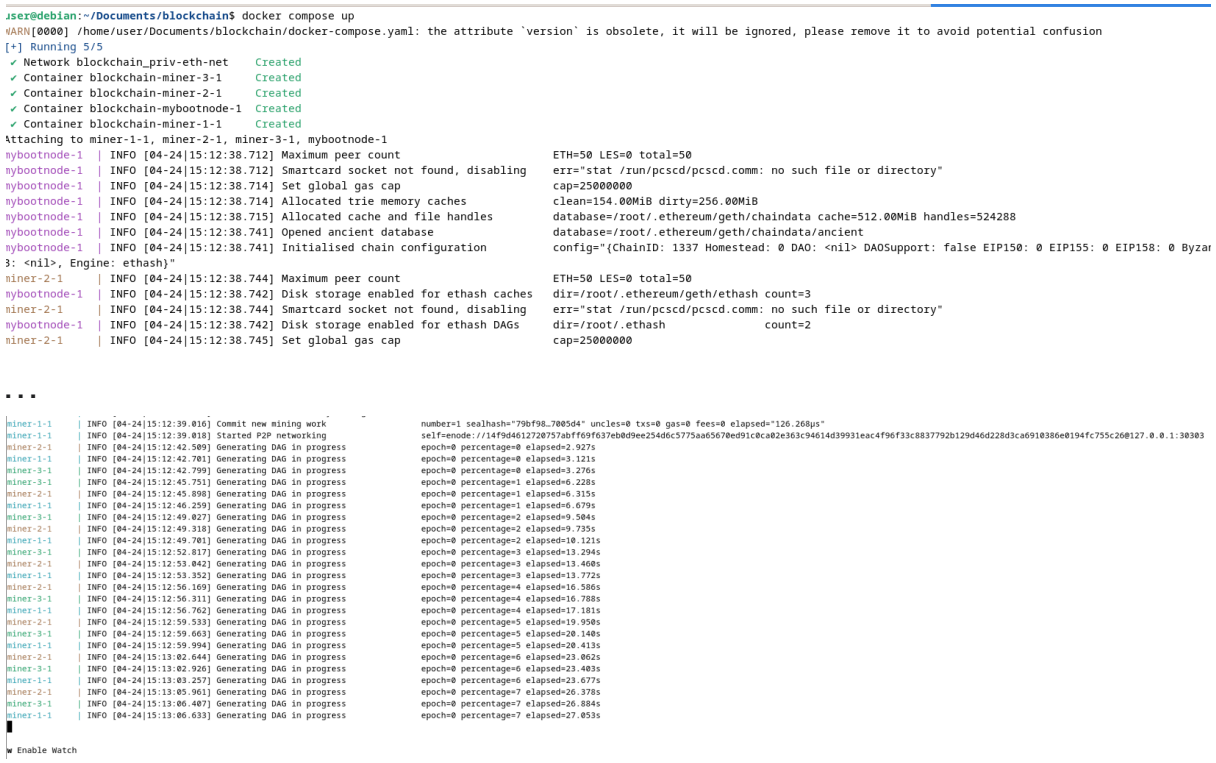
```
services:
  mybootnode:
    hostname: mybootnode
    env_file:
      - .env
    build:
      context: .
      args:
        - ACCOUNT_PASSWORD=${ACCOUNT_PASSWORD}
    command:
--nodekeyhex="4ae3b4381e02ba395606c2eedaa6050e8b03a3ea98c2
e015c7cccb1034aa5c5d" --nodiscover --ipcdisable
--networkid=${NETWORK_ID} --netrestrict="172.13.254.0/24"
    networks:
      priv-eth-net:
```

```
  miner-1:
    hostname: miner-1
    env_file:
      - .env
    build:
      context: .
```

```
      args:
        - ACCOUNT_PASSWORD=${ACCOUNT_PASSWORD}
    command:
--bootnodes="enode://6c0bdbf52eff691d706a8348e375c0218006b051
f7886f7513ff11f65f54110fffadf83b9f1c14ac49298f4dc0501c84b2a1615
a4a1703347684084e40b66a3d@mybootnode:30303" --mine
--miner.threads=1 --networkid=${NETWORK_ID}
--netrestrict="172.13.254.0/24"
    networks:
      priv-eth-net:
```

Збілдимо і запустимо контейнери:

```
docker compose build
docker compose up
```




Як бачимо, всі ноди успішно розпочали P2P нетворкінг.
Зайдемо у JavaScript консоль на одній із нод:

```
user@debian:~/Documents/blockchain$ docker exec -it blockchain-miner-1-1 /bin/sh
/ # geth attach
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.1-stable-c2d2f4ed/linux-amd64/go1.16
coinbase: 0x676af56daf37d971c9a9a9359a2d0eec748a8e16
at block: 0 (Thu Jan 01 1970 00:00:00 GMT+0000 (UTC))
 datadir: /root/.ethereum
 modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d
> admin.nodeInfo.enode
"enode://14f9d4612720757abff69f637eb0d9ee254d6c5775aa65670ed91c0ca02e363c94614d39931eac4f96f33c8837792b129d46d228d3ca6910386e0194fc755c26@127.0.0.1:30303"
>
```

## Проведемо транзакцію:

```
> eth.accounts
["0x676af56daf37d971c9a9a9359a2d0eec748a8e16"]
> eth.getBalance(eth.accounts[0])
0
>
```

## Тут треба трохи зачекати через довгу ініціалізацію

```
> eth.getBalance(eth.accounts[0])
140000000000000000000
>
```

```
> personal.unlockAccount(eth.accounts[0], "oPAJqp20Pa012jpNAlao29030OJBao")
true
```

## Створимо нового користувача:

```
> personal.newAccount()
Passphrase:
Repeat passphrase:
"0xdc00901b7a262c0d187c26c283d392142a9b34ac"
> eth.accounts
["0x676af56daf37d971c9a9a9359a2d0eec748a8e16", "0xdc00901b7a262c0d187c26c283d392142a9b34ac"]
> eth.getBalance(eth.accounts[1])
0
> personal.unlockAccount(eth.accounts[1], "password")
true
>
```

## Спробуємо провести транзакцію:

```
> eth.sendTransaction({from:eth.accounts[1], to:"0x676af56daf37d971c9a9a9359a2d0eec748a8e16", value: web3.toWei(1, "ether")})
Error: insufficient funds for transfer
        at web3.js:6347:37(47)
        at web3.js:5081:62(37)
        at <eval>:1:20(18)
```

А тепер з аккаунту, на якому є ефіри, надішлемо кошти, перевіримо баланс та переглянемо транзакцію:

```
> eth.sendTransaction({from:eth.accounts[0], to:"0xdc00901b7a262c0d187c26c283d392142a9b34ac", value: web3.toWei(1, "ether")})
"0xecff78120bb6294778ac02a00ee71a8f1c7de3cd9fa6382baee1f46f14fff859"
> eth.getBalance(eth.accounts[1])
100000000000000000
> eth.getTransaction("0xecff78120bb6294778ac02a00ee71a8f1c7de3cd9fa6382baee1f46f14fff859")
{
  blockHash: "0x427d6f412083f3b4b48a3ba15ac0a1fd11e2f49d3c04fa4a3f3b98a0e4b7fe4b",
  blockNumber: 98,
  from: "0x676af56daf37d971c9a9a9359a2d0eec748a8e16",
  gas: 21000,
  gasPrice: 1000000000,
  hash: "0xecff78120bb6294778ac02a00ee71a8f1c7de3cd9fa6382baee1f46f14fff859",
  input: "0x",
  nonce: 0,
  r: "0xebf466011d9ce91a8a03b36c00011a6c4f6f2996e9dffaeab611c830c980f184",
  s: "0x6890cae73387754bb23f7b07f9296a86a2659506d67bf86ac587aea5aa08ab68",
  to: "0xdc00901b7a262c0d187c26c283d392142a9b34ac",
  transactionIndex: 0,
  type: "0x0",
  v: "0xa96",
  value: 100000000000000000
}
>
```