Blockchain_lab1_Bondarenko_Kryhin_Shanidze_FB-41mn

Лабораторна робота № 2

Тема: Реалізація смарт-контракту або анонімної криптовалюти

Мета роботи: «Отримання навичок роботи із смарт-контрактами або анонімними криптовалютами»

Виконали: Бондаренко Олексій (ФБ-41мн), Кригін Дмитро (ФБ-41мн), Шанідзе Давид (ФБ-41мн)

Хід роботи

1. Monero

https://getmonero.dev/infrastructure/networks.html
https://github.com/moneroexamples/private-testnet

Встановимо все необхідне:

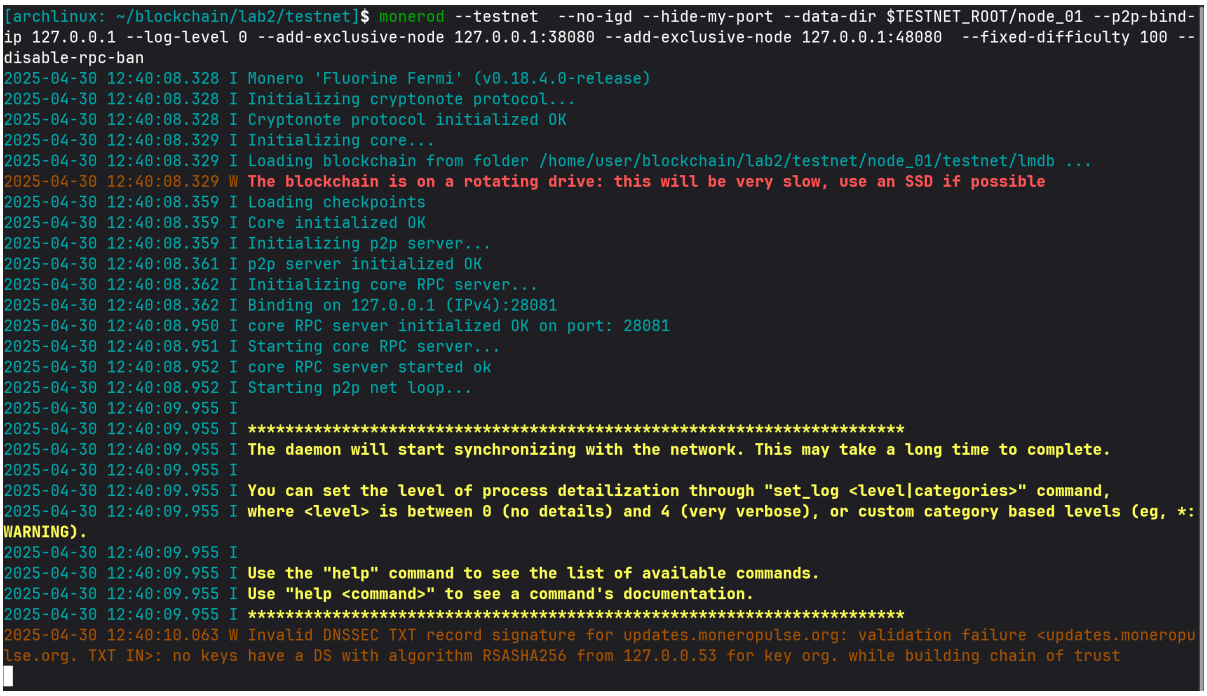
```
sudo pacman -S monero
```

Створимо кореневий каталог:

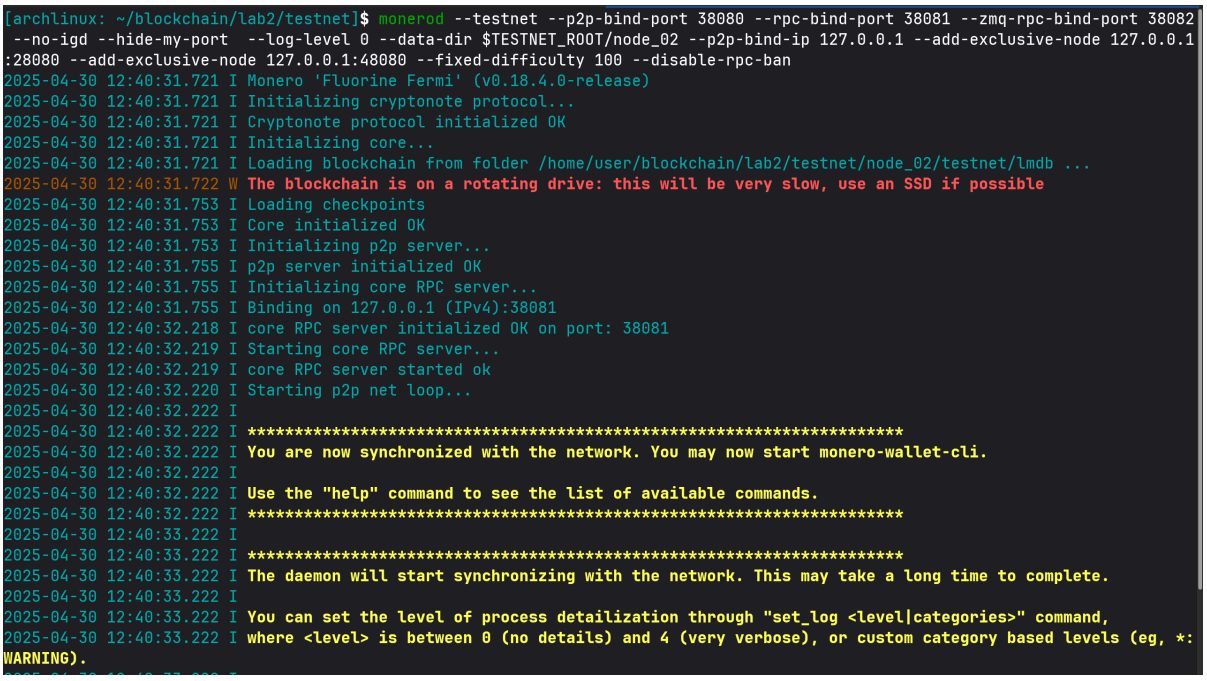
```
export TESTNET_ROOT="/home/user/blockchain/lab2/testnet"
mkdir $TESTNET_ROOT; cd $TESTNET_ROOT
```

Створимо 3 вузли:

```
monerod --testnet  --no-igd --hide-my-port --data-dir $TESTNET_ROOT/node_01 --p2p-bind-ip 127.0.0.1 --log-level 0 --add-exclusive-node 127.0.0.1:38080 --add-exclusive-node 127.0.0.1:48080  --fixed-difficulty 100 --disable-rpc-ban
```




```
monerod --testnet --p2p-bind-port 38080 --rpc-bind-port 38081 --zmq-rpc-bind-port 38082 --no-igd --hide-my-port  --log-level 0 --data-dir $TESTNET_ROOT/node_02 --p2p-bind-ip 127.0.0.1 --add-exclusive-node 127.0.0.1:28080 --add-exclusive-node 127.0.0.1:48080 --fixed-difficulty 100 --disable-rpc-ban
```




```
monerod --testnet --p2p-bind-port 48080 --rpc-bind-port 48081 --zmq-rpc-bind-port 48082 --no-igd --hide-my-port  --log-level 0 --data-dir $TESTNET_ROOT/node_03 --p2p-bind-ip 127.0.0.1 --add-exclusive-node 127.0.0.1:28080 --add-exclusive-node 127.0.0.1:38080 --fixed-difficulty 100 --disable-rpc-ban
```

```
[archlinux: ~/blockchain/lab2/testnet]$ monerod --testnet --p2p-bind-port 48080 --rpc-bind-port 48081 --zmq-rpc-bind-port 48082
 --no-igd --hide-my-port  --log-level 0 --data-dir $TESTNET_ROOT/node_03 --p2p-bind-ip 127.0.0.1 --add-exclusive-node 127.0.0.1
:28080 --add-exclusive-node 127.0.0.1:38080 --fixed-difficulty 100 --disable-rpc-ban
2025-04-30 12:41:01.924 I Monero 'Fluorine Fermi' (v0.18.4.0-release)
2025-04-30 12:41:01.924 I Initializing cryptonote protocol...
2025-04-30 12:41:01.924 I Cryptonote protocol initialized OK
2025-04-30 12:41:01.924 I Initializing core...
2025-04-30 12:41:01.924 I Loading blockchain from folder /home/user/blockchain/lab2/testnet/node_03/testnet/lmdb ...
2025-04-30 12:41:01.924 W The blockchain is on a rotating drive: this will be very slow, use an SSD if possible
2025-04-30 12:41:01.953 I Loading checkpoints
2025-04-30 12:41:01.953 I Core initialized OK
2025-04-30 12:41:01.953 I Initializing p2p server...
2025-04-30 12:41:01.955 I p2p server initialized OK
2025-04-30 12:41:01.956 I Initializing core RPC server...
2025-04-30 12:41:01.956 I Binding on 127.0.0.1 (IPv4):48081
2025-04-30 12:41:02.388 I core RPC server initialized OK on port: 48081
2025-04-30 12:41:02.389 I Starting core RPC server...
2025-04-30 12:41:02.389 I core RPC server started ok
2025-04-30 12:41:02.390 I Starting p2p net loop...
2025-04-30 12:41:02.391 I
2025-04-30 12:41:02.391 I **********************************************************************
2025-04-30 12:41:02.391 I You are now synchronized with the network. You may now start monero-wallet-cli.
2025-04-30 12:41:02.392 I
2025-04-30 12:41:02.392 I Use the "help" command to see the list of available commands.
2025-04-30 12:41:02.392 I **********************************************************************
2025-04-30 12:41:03.391 I
2025-04-30 12:41:03.391 I **********************************************************************
2025-04-30 12:41:03.391 I The daemon will start synchronizing with the network. This may take a long time to complete.
2025-04-30 12:41:03.391 I
2025-04-30 12:41:03.391 I You can set the level of process detailization through "set_log <level|categories>" command,
2025-04-30 12:41:03.391 I where <level> is between 0 (no details) and 4 (very verbose), or custom category based levels (eg, *:
WARNING).
```

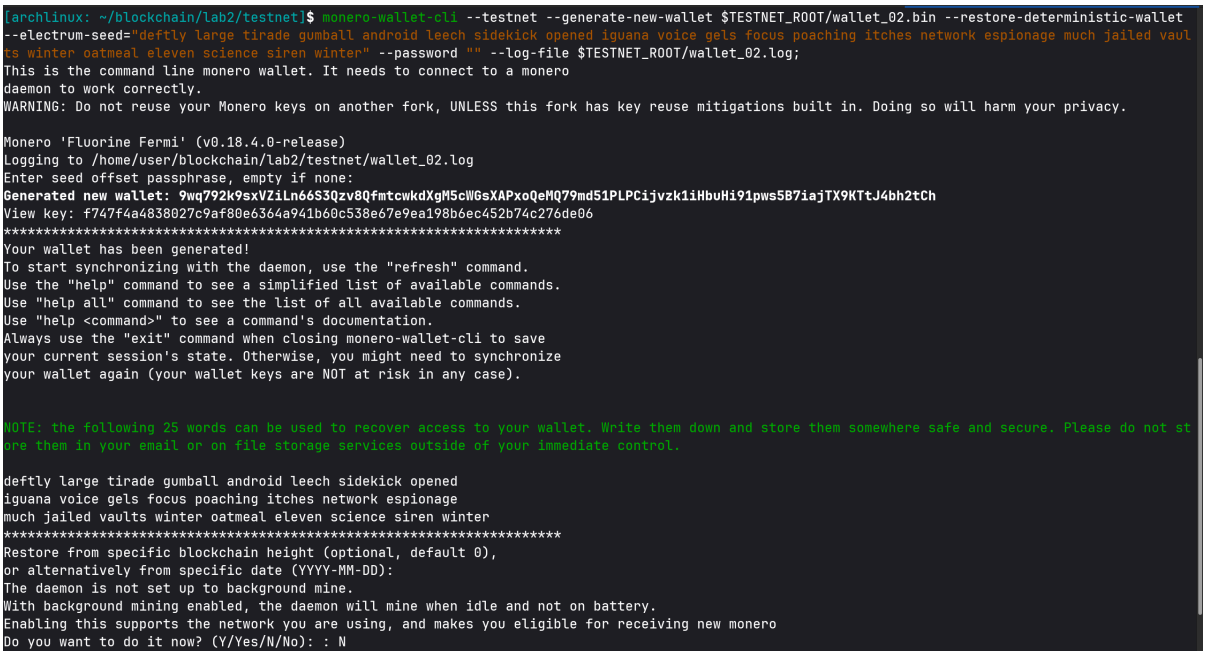
Створимо 3 гаманці:

```
monero-wallet-cli --testnet --generate-new-wallet $TESTNET_ROOT/wallet_01.bin --
restore-deterministic-wallet --electrum-seed="sequence atlas unveil summon pebbles
tuesday beer rudely snake rockets different fuselage woven tagged bested dented
vegan hover rapid fawns obvious muppet randomly seasons randomly" --password "" --
log-file $TESTNET_ROOT/wallet_01.log;
```

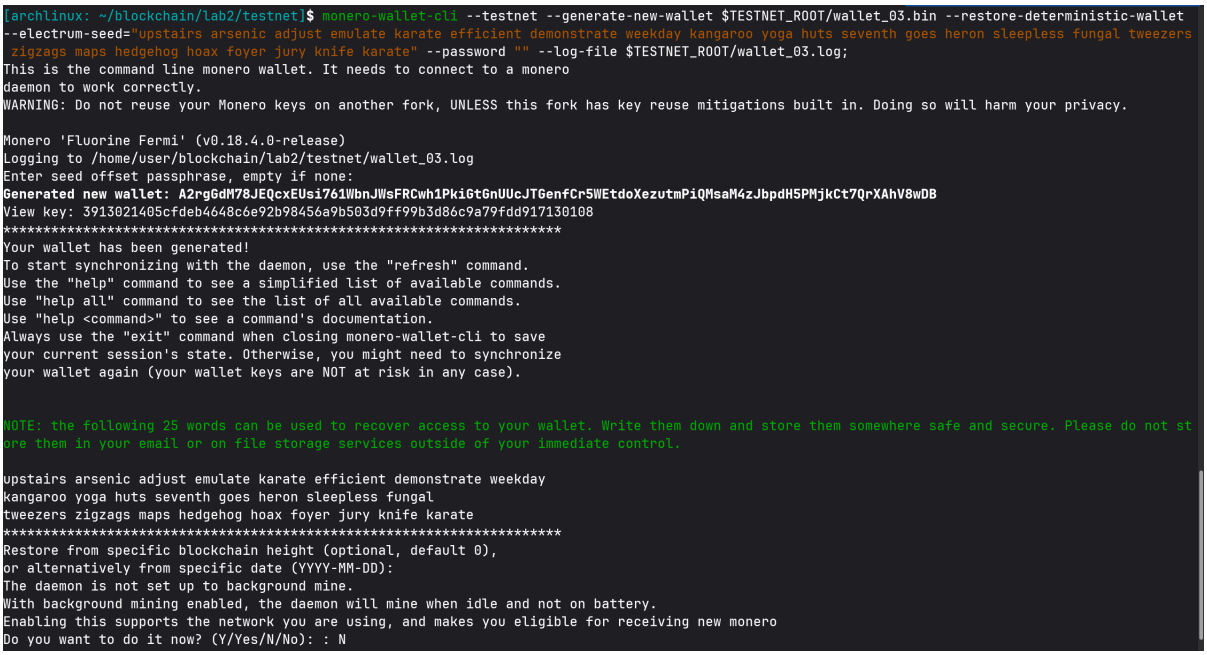
```
[archlinux: ~/blockchain/lab2/testnet]$ monero-wallet-cli --testnet --generate-new-wallet $TESTNET_ROOT/wallet_01.bin --restore-deterministic-wallet
--electrum-seed="sequence atlas unveil summon pebbles tuesday beer rudely snake rockets different fuselage woven tagged bested dented vegan hover rap
id fawns obvious muppet randomly seasons randomly" --password "" --log-file $TESTNET_ROOT/wallet_01.log;
This is the command line monero wallet. It needs to connect to a monero
daemon to work correctly.
WARNING: Do not reuse your Monero keys on another fork, UNLESS this fork has key reuse mitigations built in. Doing so will harm your privacy.

Monero 'Fluorine Fermi' (v0.18.4.0-release)
Logging to /home/user/blockchain/lab2/testnet/wallet_01.log
Enter seed offset passphrase, empty if none:
Generated new wallet: 9wviCeWe2D8XS82k2ovp5EUYLzBt9pYNW2LXUFsZiv8S3Mt21FZ5qQaAroko1enzw3e6r9qC7X1D7Geoo2RrAotYPwq9Gm8
View key: 42ba20adb337e5eca797565be11c9adb0a8bef8c830bccc2df712535d3b8f608
**********************************************************************
Your wallet has been generated!
To start synchronizing with the daemon, use the "refresh" command.
Use the "help" command to see a simplified list of available commands.
Use "help all" command to see the list of all available commands.
Use "help <command>" to see a command's documentation.
Always use the "exit" command when closing monero-wallet-cli to save
your current session's state. Otherwise, you might need to synchronize
your wallet again (your wallet keys are NOT at risk in any case).


NOTE: the following 25 words can be used to recover access to your wallet. Write them down and store them somewhere safe and secure. Please do not st
ore them in your email or on file storage services outside of your immediate control.

sequence atlas unveil summon pebbles tuesday beer rudely
snake rockets different fuselage woven tagged bested dented
vegan hover rapid fawns obvious muppet randomly seasons randomly
**********************************************************************
Restore from specific blockchain height (optional, default 0),
or alternatively from specific date (YYYY-MM-DD):
The daemon is not set up to background mine.
With background mining enabled, the daemon will mine when idle and not on battery.
Enabling this supports the network you are using, and makes you eligible for receiving new monero
Do you want to do it now? (Y/Yes/N/No): : N
```

```
monero-wallet-cli --testnet --generate-new-wallet $TESTNET_ROOT/wallet_02.bin --
restore-deterministic-wallet --electrum-seed="deftly large tirade gumball android
leech sidekick opened iguana voice gels focus poaching itches network espionage
```

```
much jailed vaults winter oatmeal eleven science siren winter" --password "" --log-
file $TESTNET_ROOT/wallet_02.log;
```




```
monero-wallet-cli --testnet --generate-new-wallet $TESTNET_ROOT/wallet_03.bin --
restore-deterministic-wallet --electrum-seed="upstairs arsenic adjust emulate
karate efficient demonstrate weekday kangaroo yoga huts seventh goes heron
sleepless fungal tweezers zigzags maps hedgehog hoax foyer jury knife karate" --
password "" --log-file $TESTNET_ROOT/wallet_03.log;
```



Майнинг Monero

Запустимо майнинг для перщого гаманця (в інтерактивному режимі):

```
monero-wallet-cli --testnet --trusted-daemon --daemon-address 127.0.0.1:28081 --
wallet-file $TESTNET_ROOT/wallet_01.bin --password '' --log-file
$TESTNET_ROOT/wallet_01.log
...
start_mining 1
```

```
[wallet 9wviCe]: status
Refreshed 64/64, synced, daemon RPC v3.14, SSL
[wallet 9wviCe]: balance
Currently selected account: [0] Primary account
Tag: (No tag assigned)
Balance: 1108.273898597315, unlocked balance: 70.368576405662 (59 block(s) to unlock)
[wallet 9wviCe]: stop_mining
Mining stopped in daemon
[wallet 9wviCe]:
```

Запустимо майнинг для другого гаманця (без інтерактивного режиму):

```
monero-wallet-cli --testnet --trusted-daemon --daemon-address 127.0.0.1:28081 --
wallet-file $TESTNET_ROOT/wallet_01.bin --password '' --log-file
$TESTNET_ROOT/wallet_01.log
```

```
[archlinux: ~/blockchain/lab2/testnet]$ monero-wallet-cli --testnet --trusted-daemon --daemon-address 127.0.0.1:38081 --wallet-file $TESTNET_ROOT/wal
let_02.bin --password '' --log-file $TESTNET_ROOT/wallet_02.log start_mining 8
This is the command line monero wallet. It needs to connect to a monero
daemon to work correctly.
WARNING: Do not reuse your Monero keys on another fork, UNLESS this fork has key reuse mitigations built in. Doing so will harm your privacy.

Monero 'Fluorine Fermi' (v0.18.4.0-release)
Logging to /home/user/blockchain/lab2/testnet/wallet_02.log
Opened wallet: 9wq792k9sxVZiLn66S3Qzv8QfmtcwkdXgM5cWGsXAPxoQeMQ79md51PLPCijvzk1iHbuHi91pws5B7iajTX9KTtJ4bh2tCh
******************************************************************
Use the "help" command to see a simplified list of available commands.
Use "help all" to see the list of all available commands.
Use "help <command>" to see a command's documentation.
******************************************************************
Mining started in daemon
[archlinux: ~/blockchain/lab2/testnet]$
```

```
[wallet 9wq792]: status
Refreshed 309/309, synced, daemon RPC v3.14, SSL
[wallet 9wq792]: balance
Currently selected account: [0] Primary account
Tag: (No tag assigned)
Balance: 3887.139054550819, unlocked balance: 2849.476210939997 (59 block(s) to unlock)
[wallet 9wq792]: stop_mining
Mining stopped in daemon
[wallet 9wq792]:
```

Як видно, намайнили трохи Monero. Логи другої ноди:

```
2025-04-30 13:07:53.308 I Found block <d69ec9aa87cc25cbed658d80f5b0cf7a6d5d7c7921d85f521b584aad7ee23418> at height 357 for difficulty: 100
2025-04-30 13:07:53.309 I Found block <4ffd5e0eff1d7725cb8d7c77fea056a8d6c5e6c4673d6be312b5fba25bb9e4ee> at height 357 for difficulty: 100
2025-04-30 13:07:53.354 I ----- BLOCK ADDED AS ALTERNATIVE ON HEIGHT 357
2025-04-30 13:07:53.354 I id:   <4ffd5e0eff1d7725cb8d7c77fea056a8d6c5e6c4673d6be312b5fba25bb9e4ee>
2025-04-30 13:07:53.354 I PoW:  <f5c599638445db8690dd388f7c561e19a02aa473926822cc55bc77e41dcb7e02>
2025-04-30 13:07:53.354 I difficulty:   100
2025-04-30 13:07:53.507 I Found block <7dc42ca214b01cdef56bec483a79959cc1fe8b839d9d42d8590040ebc1729f6a> at height 358 for difficulty: 100
2025-04-30 13:07:53.514 I Found block <eb4d8b01f64561a5db3109f4b9f8785a0cebd05d1df14aa3a99be80f3f34b7b2> at height 358 for difficulty: 100
2025-04-30 13:07:53.514 I Found block <197e0554cec0a4266defcad9cf4387483297dfe2b074afcc20570488206acefa> at height 358 for difficulty: 100
2025-04-30 13:07:53.551 I ----- BLOCK ADDED AS ALTERNATIVE ON HEIGHT 358
2025-04-30 13:07:53.551 I id:   <eb4d8b01f64561a5db3109f4b9f8785a0cebd05d1df14aa3a99be80f3f34b7b2>
2025-04-30 13:07:53.551 I PoW:  <8752b3709eae18c4461ce70f72236a0dd2d054c7938505b96397952e32c54c00>
2025-04-30 13:07:53.551 I difficulty:   100
2025-04-30 13:07:53.578 I ----- BLOCK ADDED AS ALTERNATIVE ON HEIGHT 358
2025-04-30 13:07:53.578 I id:   <197e0554cec0a4266defcad9cf4387483297dfe2b074afcc20570488206acefa>
2025-04-30 13:07:53.578 I PoW:  <b72a8dac032b276505527c829de6961c3908ad694877af4420224453ece6cc01>
2025-04-30 13:07:53.578 I difficulty:   100
2025-04-30 13:07:53.829 I Found block <ed4e8daad06740bda39c1db887c2a7ae21fb954502a3009377315405e786b1af> at height 359 for difficulty: 100
2025-04-30 13:07:53.919 I Found block <18cc1cbf89a435d63dedad5f7cfe536a9b6faccd123e6f27d14bd0a1e82e39cd> at height 360 for difficulty: 100
2025-04-30 13:07:54.125 I Found block <759fc74df9f4ed1331d0b867161091acc18171595b7b6488688bbfbc0b1b513> at height 361 for difficulty: 100
2025-04-30 13:07:54.714 I Found block <2c75c1f2fc7c42617af95ffbb9ecdfd70ec58a5f65887c331f3b0f33206764ed> at height 362 for difficulty: 100
2025-04-30 13:07:54.855 I Found block <cefcee52aa2ec4ea1ae35f6db3081d068f2c708d8580483c0acf67f9fb484ec5> at height 363 for difficulty: 100
2025-04-30 13:07:55.025 I Found block <348de6fd971679d7376857306af39e7c9da604c4ffef254e497f2063cf9a2c50> at height 364 for difficulty: 100
2025-04-30 13:07:55.141 I Found block <f55c43fdc25d20533944566a8d428d59905c64f8c9e57c201e86b35f507c1505> at height 365 for difficulty: 100
2025-04-30 13:07:55.535 I Found block <81fd7fc0c45232dd073b527c26dac14585f270386ebe69dad9c93d5fe7b0cbbe> at height 366 for difficulty: 100
2025-04-30 13:07:55.540 I Miner thread stopped [2]
2025-04-30 13:07:55.544 I Miner thread stopped [5]
2025-04-30 13:07:55.544 I Miner thread stopped [1]
2025-04-30 13:07:55.546 I Miner thread stopped [4]
2025-04-30 13:07:55.547 I Miner thread stopped [7]
2025-04-30 13:07:55.549 I Miner thread stopped [6]
2025-04-30 13:07:55.549 I Miner thread stopped [3]
2025-04-30 13:07:55.553 I Miner thread stopped [0]
```

Трансфер Monero

Перекинемо монеро з гаманця 1 на гаманець 3.

```
[wallet 9wviCe]: transfer normal 1 A2rgGdM78JEQcxEUsi761WbnJWsFRCwh1PkiGtGnUUcJTGenfCr5WEtdoXezutmPiQMsaM4zJbpdH5PMjkCt7QrXAhV8wDB 1337
Wallet password:

Transaction 1/3:
Spending from address index 0

Transaction 2/3:
Spending from address index 0

Transaction 3/3:
Spending from address index 0
Sending 1337.000000000000.  Your transaction needs to be split into 3 transactions.  This will result in a transaction fee being applied to each tran
saction, for a total fee of 0.168000000000


WARNING: this is a non default ring size, which may harm your privacy. Default is recommended.
Is this okay?  (Y/Yes/N/No): Y
Transaction successfully submitted, transaction <37f0d0a89a194f0bd3b77baba78a7391bdd06ee474f395b8300c0db5d5502aa1>
You can check its status by using the `show_transfers` command.
Transaction successfully submitted, transaction <01be8771cb35a04a0fa3bfef7a5b320a6357262518b36e9029f0523f611b0a24>
You can check its status by using the `show_transfers` command.
Transaction successfully submitted, transaction <b098888ba0b0a2fbbf802c1e5c01cf1ed89f0d9618798fb4affb5ac724911a4b>
You can check its status by using the `show_transfers` command.
```

Лог транзакцій на гаманці 1:

```
    6945  block unlocked       2025-04-30 13:52:04      17.476053237817 f7125d050af6b858144160dbf735abbf35bcabb188ca8f790d387153b6dd95ac 0000000000
0000 0.000000000000 9wviCeWe2D8XS82k2ovp5EUYLzBt9pYNW2LXUFsZiv8S3Mt21FZ5qQaAroko1enzw3eGr9qC7X1D7Geoo2RrAotYPwq9Gm8:17.476053237817 0 -
    6946    out        -       2025-04-30 13:52:04     156.837000000000 37f0d0a89a194f0bd3b77baba78a7391bdd06ee474f395b8300c0db5d5502aa1 0000000000
0000 0.024000000000 A2rgGdM78JEQcxEUsi761WbnJWsFRCwh1PkiGtGnUUcJTGenfCr5WEtdoXezutmPiQMsaM4zJbpdH5PMjkCt7QrXAhV8wDB:156.837000000000 0 -
    6946    out        -       2025-04-30 13:52:04     553.741000000000 01be8771cb35a04a0fa3bfef7a5b320a6357262518b36e9029f0523f611b0a24 0000000000
0000 0.072000000000 A2rgGdM78JEQcxEUsi761WbnJWsFRCwh1PkiGtGnUUcJTGenfCr5WEtdoXezutmPiQMsaM4zJbpdH5PMjkCt7QrXAhV8wDB:553.741000000000 0 -
    6946  block unlocked       2025-04-30 13:52:04      17.026692961726 da3ba2cf80fff4dc2835b26ffef5ea3a8aa1d33bb8983549f716dc2e9ce5c66a 0000000000
0000 0.000000000000 9wviCeWe2D8XS82k2ovp5EUYLzBt9pYNW2LXUFsZiv8S3Mt21FZ5qQaAroko1enzw3eGr9qC7X1D7Geoo2RrAotYPwq9Gm8:17.026692961726 0 -
    6947  block unlocked       2025-04-30 13:52:04      17.548020424987 e8ce8eb1a3ab02eb9874d034cc66cac0d0e11ca8f746fc7b82870bb274c03c00 0000000000
0000 0.000000000000 9wviCeWe2D8XS82k2ovp5EUYLzBt9pYNW2LXUFsZiv8S3Mt21FZ5qQaAroko1enzw3eGr9qC7X1D7Geoo2RrAotYPwq9Gm8:17.548020424987 0 -
    6947    out        -       2025-04-30 13:52:04     626.422000000000 b098888ba0b0a2fbbf802c1e5c01cf1ed89f0d9618798fb4affb5ac724911a4b 0000000000
0000 0.072000000000 A2rgGdM78JEQcxEUsi761WbnJWsFRCwh1PkiGtGnUUcJTGenfCr5WEtdoXezutmPiQMsaM4zJbpdH5PMjkCt7QrXAhV8wDB:626.422000000000 0 -
    6948  block unlocked       2025-04-30 13:52:05      17.476003758555 6822b88655617ced2c70b39cb7c21b5ab6d7f21e5555b0028a43fa6c5c80e16f 0000000000
0000 0.000000000000 9wviCeWe2D8XS82k2ovp5EUYLzBt9pYNW2LXUFsZiv8S3Mt21FZ5qQaAroko1enzw3eGr9qC7X1D7Geoo2RrAotYPwq9Gm8:17.476003758555 0 -
```

Лог транзакцій на гаманці 3:

```
[wallet A2rgGd]: balance
Currently selected account: [0] Primary account
Tag: (No tag assigned)
Balance: 1338.000000000000, unlocked balance: 1338.000000000000
[wallet A2rgGd]: show_transfers
    6441     in unlocked       2025-04-30 13:49:34       1.000000000000 f097b683b44cd9012141f059b0f85d2871c51c75f3dfb57a2b2bec6289762ec6 0000000000
0000 0.000000000000 A2rgGd:1.000000000000 0 -
    6946     in unlocked       2025-04-30 13:52:04     553.741000000000 01be8771cb35a04a0fa3bfef7a5b320a6357262518b36e9029f0523f611b0a24 0000000000
0000 0.000000000000 A2rgGd:553.741000000000 0 -
    6946     in unlocked       2025-04-30 13:52:04     156.837000000000 37f0d0a89a194f0bd3b77baba78a7391bdd06ee474f395b8300c0db5d5502aa1 0000000000
0000 0.000000000000 A2rgGd:156.837000000000 0 -
    6947     in unlocked       2025-04-30 13:52:04     626.422000000000 b098888ba0b0a2fbbf802c1e5c01cf1ed89f0d9618798fb4affb5ac724911a4b 0000000000
0000 0.000000000000 A2rgGd:626.422000000000 0 -
[wallet A2rgGd]:
```

Деанонімізація в блокчейні

Один із найпоширеніших методів — це блокчейн-аналіз. Оскільки більшість криптовалют мають відкритий реєстр транзакцій, аналітики можуть відстежувати рух коштів між адресами. За допомогою кластеризації гаманців, аналізу патернів транзакцій, а також співставлення з відомими біржами або сервісами, можна ідентифікувати користувачів. Компанії на кшталт Chainalysis, CipherTrace або Elliptic спеціалізуються на таких розслідуваннях.

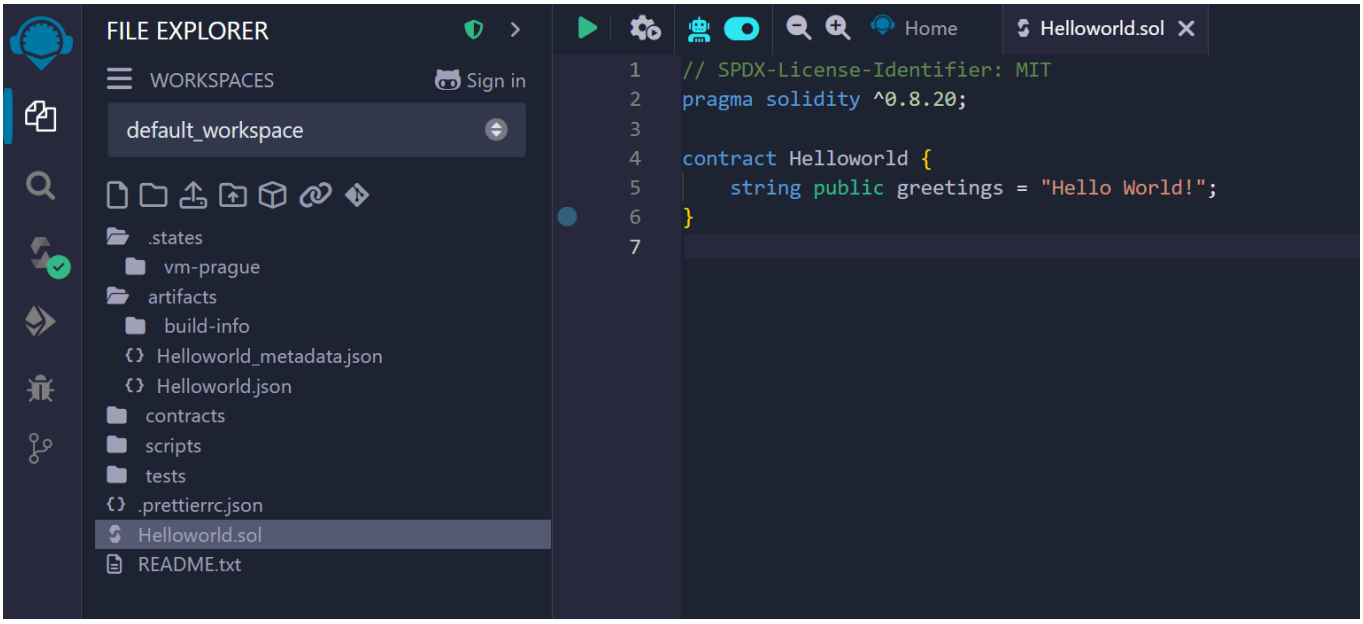
Інший напрям — це використання позаблокчейнових даних, зокрема інформації, отриманої з криптовалютних бірж, форумів або витоків даних. Оскільки більшість бірж зобов’язані дотримуватися вимог KYC (know your customer), правоохоронні органи можуть отримати реєстраційні дані користувачів за запитом. Крім того, IP-адреси, цифрові відбитки браузера, активність у соціальних мережах і навіть шаблони введення з клавіатури можуть бути використані для зв’язування криптовалютної активності з реальною особою. Таким чином, повна анонімність у блокчейні — це радше міф, особливо коли мова йде про активність на регульованих платформах.

Деанонімізація Monero можлива через *rogue nodes* та ґрунтується на тому, що мережа Monero використовує децентралізовану структуру для маршрутизації транзакцій, і деякі з цих вузлів можуть бути контрольовані атакуючим. Якщо зловмисник розгортає велику кількість вузлів у мережі, він може перехоплювати трафік, фіксувати IP-адреси користувачів та аналізувати мережеву активність до моменту застосування методів анонімізації (наприклад, протоколів Dandelion++ або Kovri, які покликані приховувати джерело транзакції). За допомогою часової кореляції між створенням транзакції та її розповсюдженням мережею, а також шляхом перехоплення «сирих» транзакцій до їхнього змішування з іншими, можна частково розкрити інформацію про відправника. Хоча Monero має потужні вбудовані механізми анонімності, такі атаки демонструють, що навіть в анонімних криптовалютах існують потенційні вектори deanonymization через контроль мережевої інфраструктури.

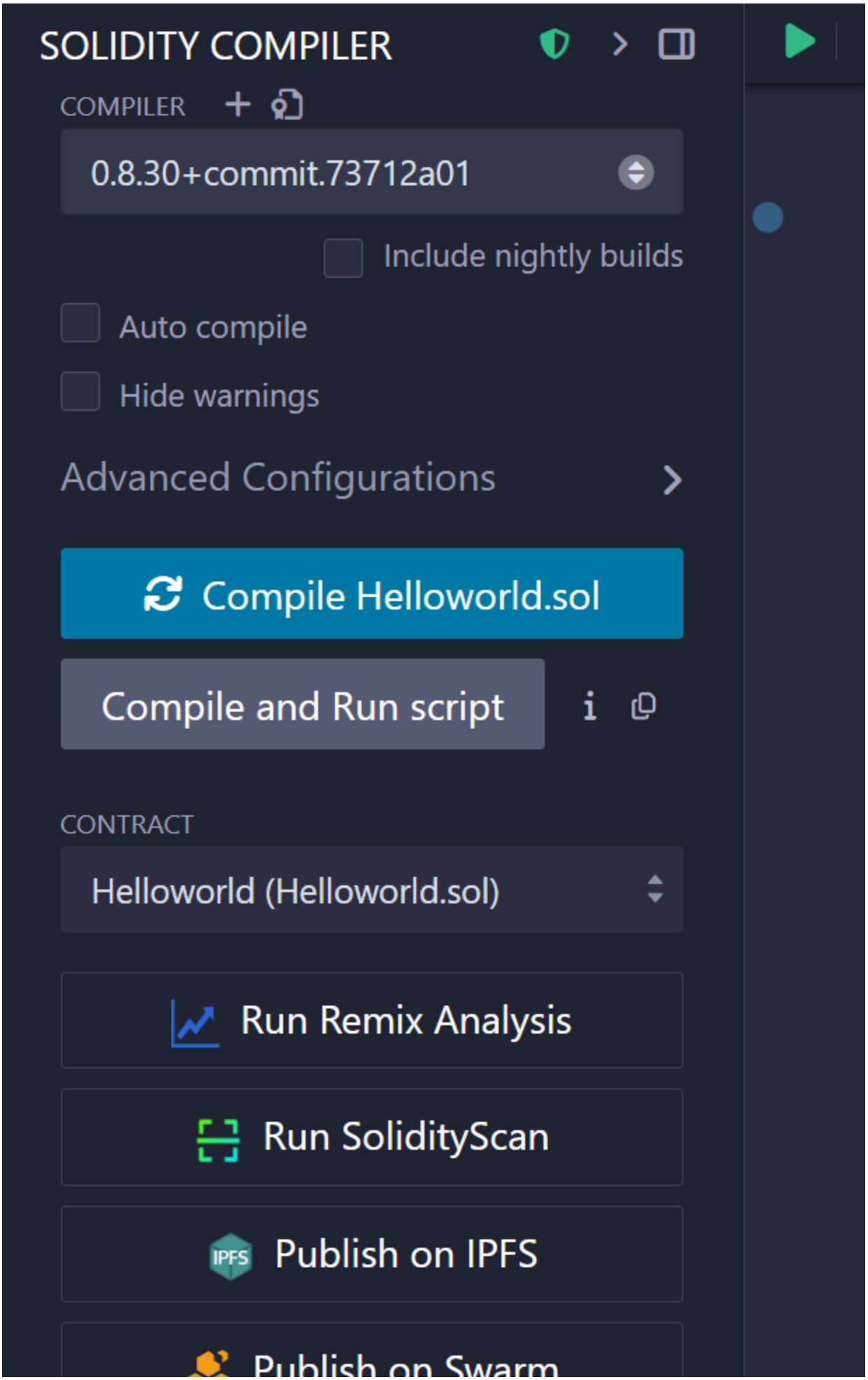
2. Смарт-контракт Ethereum

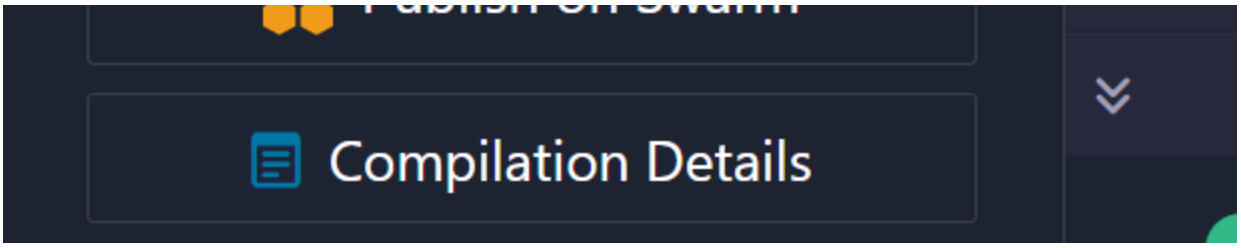
Для виконання завдання використаємо Remix IDE та задеплоїмо смарт констракт у Ethereum Sepolia Testnet. Необхідно отримати Sepolia ETH, https://faucets.chain.link/sepolia

Задеплоїмо Hello World смартконтракт:


FILE EXPLORER
WORKSPACES
Sign in
default_workspace
.states
vm-prague
artifacts
build-info
Helloworld_metadata.json
Helloworld.json
contracts
scripts
tests
.prettierrc.json
Helloworld.sol
README.txt
Home
Helloworld.sol
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.20;
3
4 contract Helloworld {
5 string public greetings = "Hello World!";
6 }
7

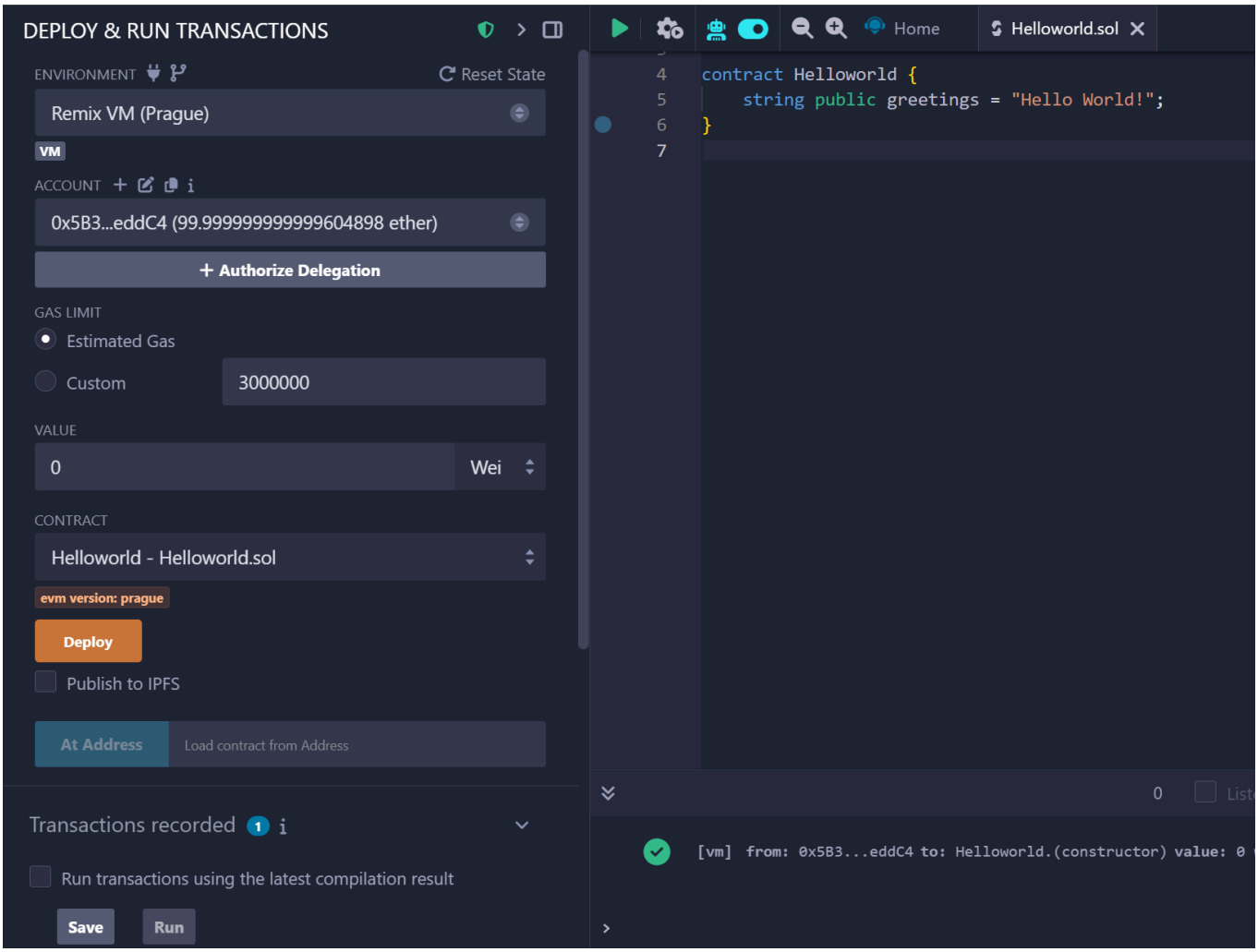
Натиснемо Compile


SOLIDITY COMPILER
COMPILER
0.8.30+commit.73712a01
Include nightly builds
Auto compile
Hide warnings
Advanced Configurations
Compile Helloworld.sol
Compile and Run script
CONTRACT
Helloworld (Helloworld.sol)
Run Remix Analysis
Run SolidityScan
Publish on IPFS
Publish on Swarm

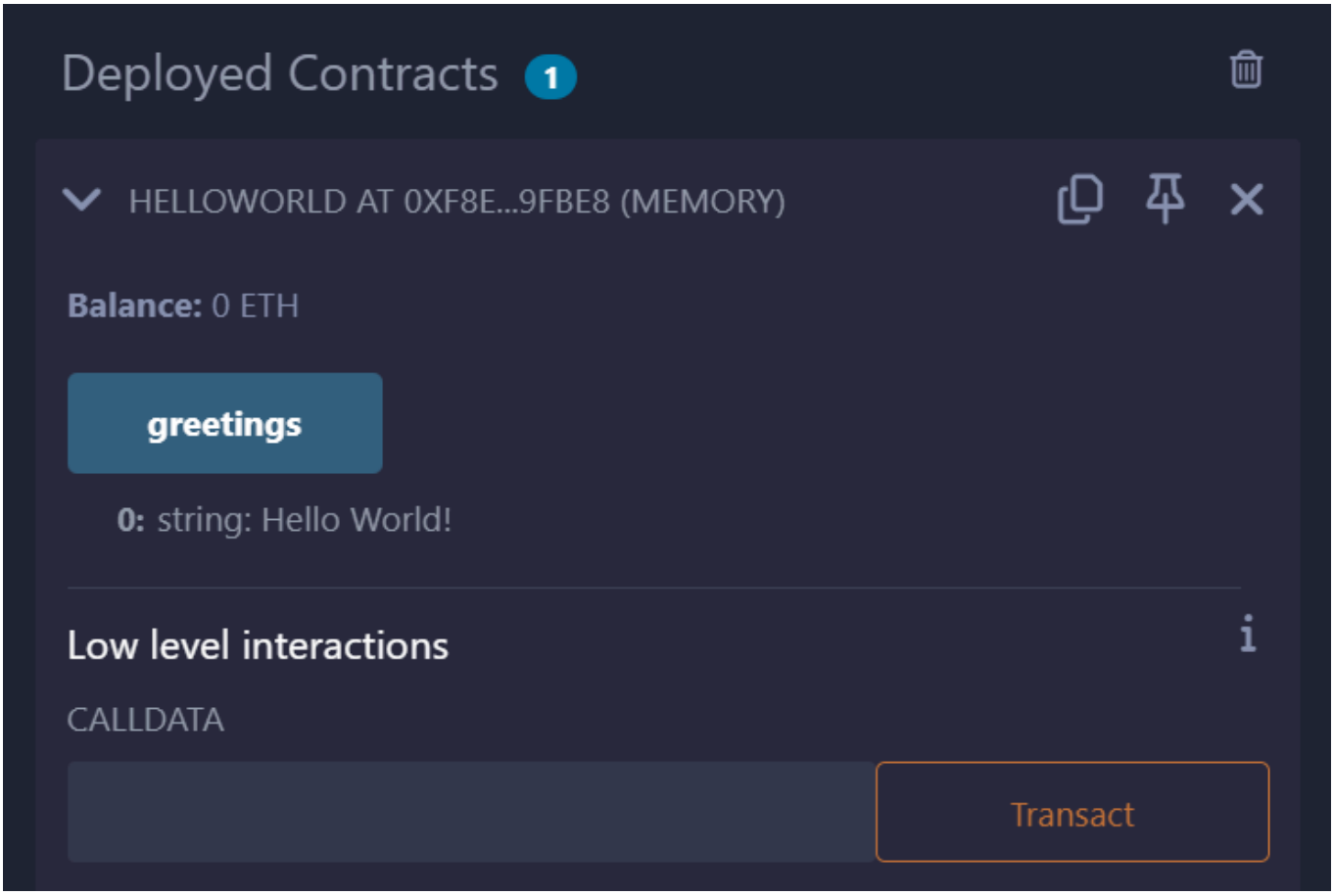


Цікаво, що є превстановлені засоби для аудиту безпеки смартконтрактів.

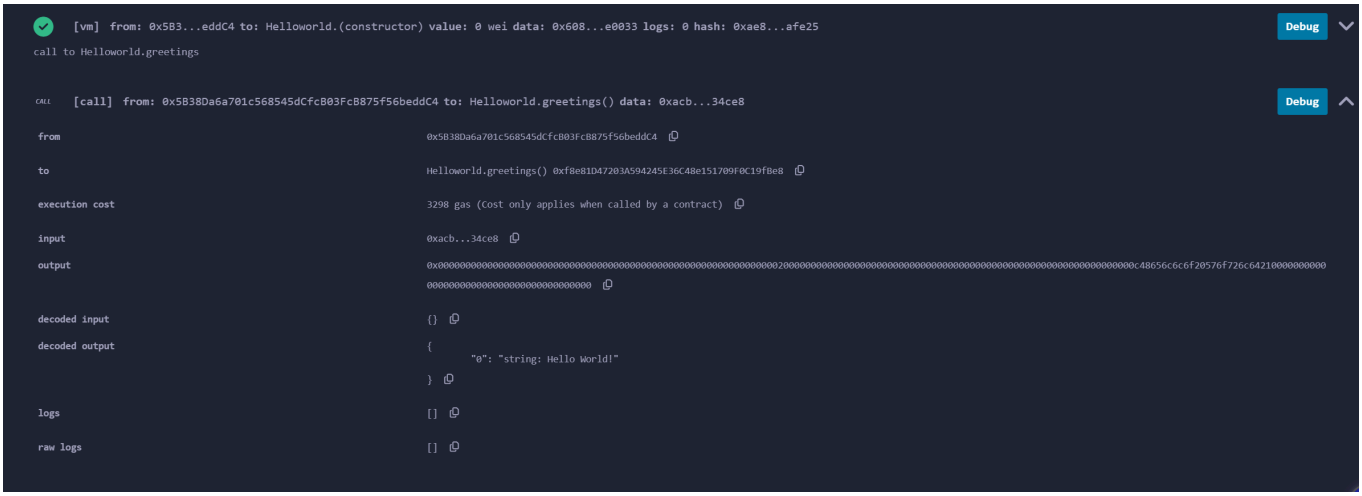
Зайдемо у Deploy&Run Transactions



Натиснемо Deploy



Переглянемо більше інформації:

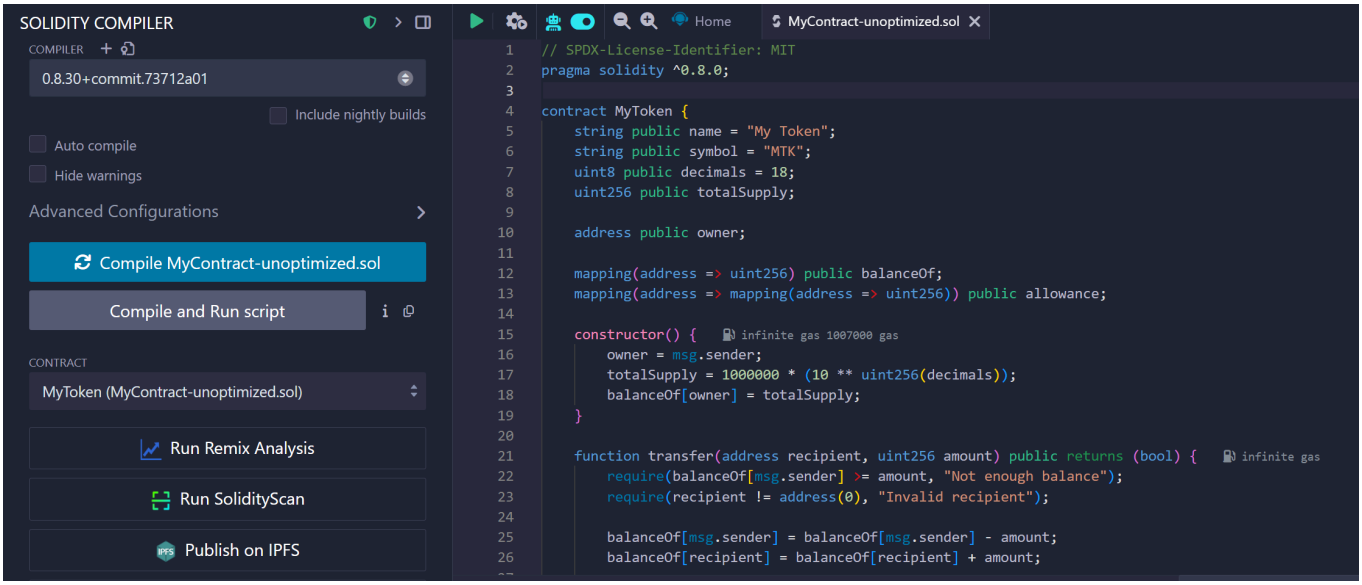


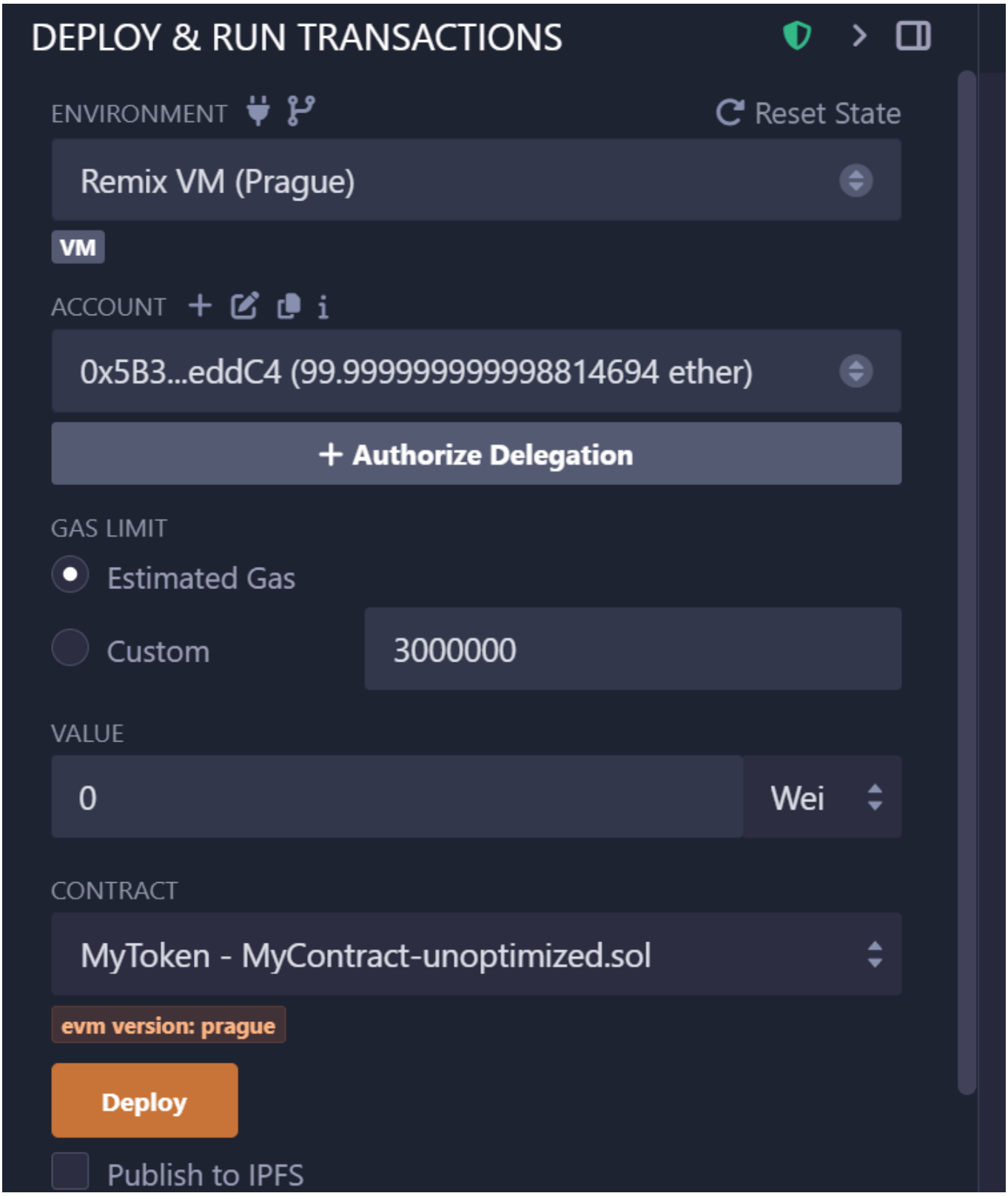
Тепер, коли ми навчилися деплоїти базовий смарт-контракт, спробуємо вирішити задачу оптимізації на складнішому прикладі.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract MyToken {
    string public name = "My Token";
```

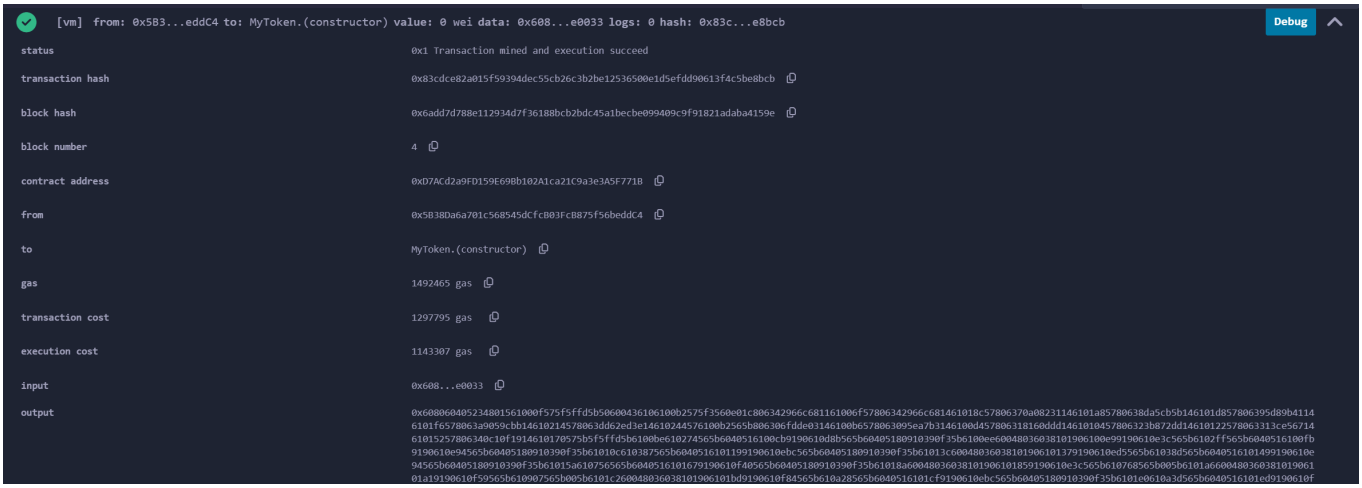
```
    string public symbol = "MTK";
    uint8 public decimals = 18;
    uint256 public totalSupply;

    address public owner;

    mapping(address => uint256) public balanceOf;
    mapping(address => mapping(address => uint256)) public allowance;

    constructor() {
        owner = msg.sender;
        totalSupply = 1000000 * (10 ** uint256(decimals));
        balanceOf[owner] = totalSupply;
    }

    function transfer(address recipient, uint256 amount) public returns (bool) {
        require(balanceOf[msg.sender] >= amount, "Not enough balance");
        require(recipient != address(0), "Invalid recipient");

        balanceOf[msg.sender] = balanceOf[msg.sender] - amount;
        balanceOf[recipient] = balanceOf[recipient] + amount;

        return true;
    }

    function approve(address spender, uint256 amount) public returns (bool) {
        allowance[msg.sender][spender] = amount;
        return true;
    }

    function transferFrom(address sender, address recipient, uint256 amount) public
returns (bool) {
        require(balanceOf[sender] >= amount, "Not enough balance");
        require(allowance[sender][msg.sender] >= amount, "Allowance exceeded");
        require(recipient != address(0), "Invalid recipient");

        balanceOf[sender] = balanceOf[sender] - amount;
        balanceOf[recipient] = balanceOf[recipient] + amount;
        allowance[sender][msg.sender] = allowance[sender][msg.sender] - amount;

        return true;
    }

    function mint(address account, uint256 amount) public {
        require(msg.sender == owner, "Only owner can mint");
        require(account != address(0), "Invalid account");
```

```
        totalSupply = totalSupply + amount;
        balanceOf[account] = balanceOf[account] + amount;
    }

    function burn(uint256 amount) public {
        require(balanceOf[msg.sender] >= amount, "Not enough to burn");

        balanceOf[msg.sender] = balanceOf[msg.sender] - amount;
        totalSupply = totalSupply - amount;
    }
}
```




DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Reset State
Remix VM (Prague)
VM
ACCOUNT
0x5B3...eddC4 (99.999999999998814694 ether)
+ Authorize Delegation
GAS LIMIT
Estimated Gas
Custom
3000000
VALUE
0
Wei
CONTRACT
MyToken - MyContract-unoptimized.sol
evm version: prague
Deploy
Publish to IPFS

Успішно задеплоїли смартконтракт:



...

1492465 gas

Проаналізуємо, чи можна оптимізувати наш смартконтракт:

У констракті імплементовано основний функціонал ERC-20: transfer, approve, transferFrom, mint і burn.

Змінні у контракті визначені не оптимально, `constant` або `immutable` значення зберігаються в байткоді, що зекономить gas, тому змінимо декларацію змінних з

```
string public name = "My Token";
string public symbol = "MTK";
uint8 public decimals = 18;
address public owner;
```

на

```
string public constant name = "My Token";
string public constant symbol = "MTK";
uint8 public constant decimals = 18;
address public immutable owner;
```

Також підрахунок нижче можна оптимізувати з

```
totalSupply = 1000000 * (10 ** uint256(decimals));
```

на

```
uint256 public constant INITIAL_SUPPLY = 1_000_000 * 10 ** 18;
```

А також можна зменшити кількість операцій зі сховищем:

```
balanceOf[msg.sender] = balanceOf[msg.sender] - amount;
```

на

```
uint256 senderBalance = balanceOf[msg.sender];
require(senderBalance >= amount, "Not enough balance");
balanceOf[msg.sender] = senderBalance - amount;
```

Таким чином ми закешуємо значення у змінній і не будемо двічі читати `balanceOf[msg.sender]`.

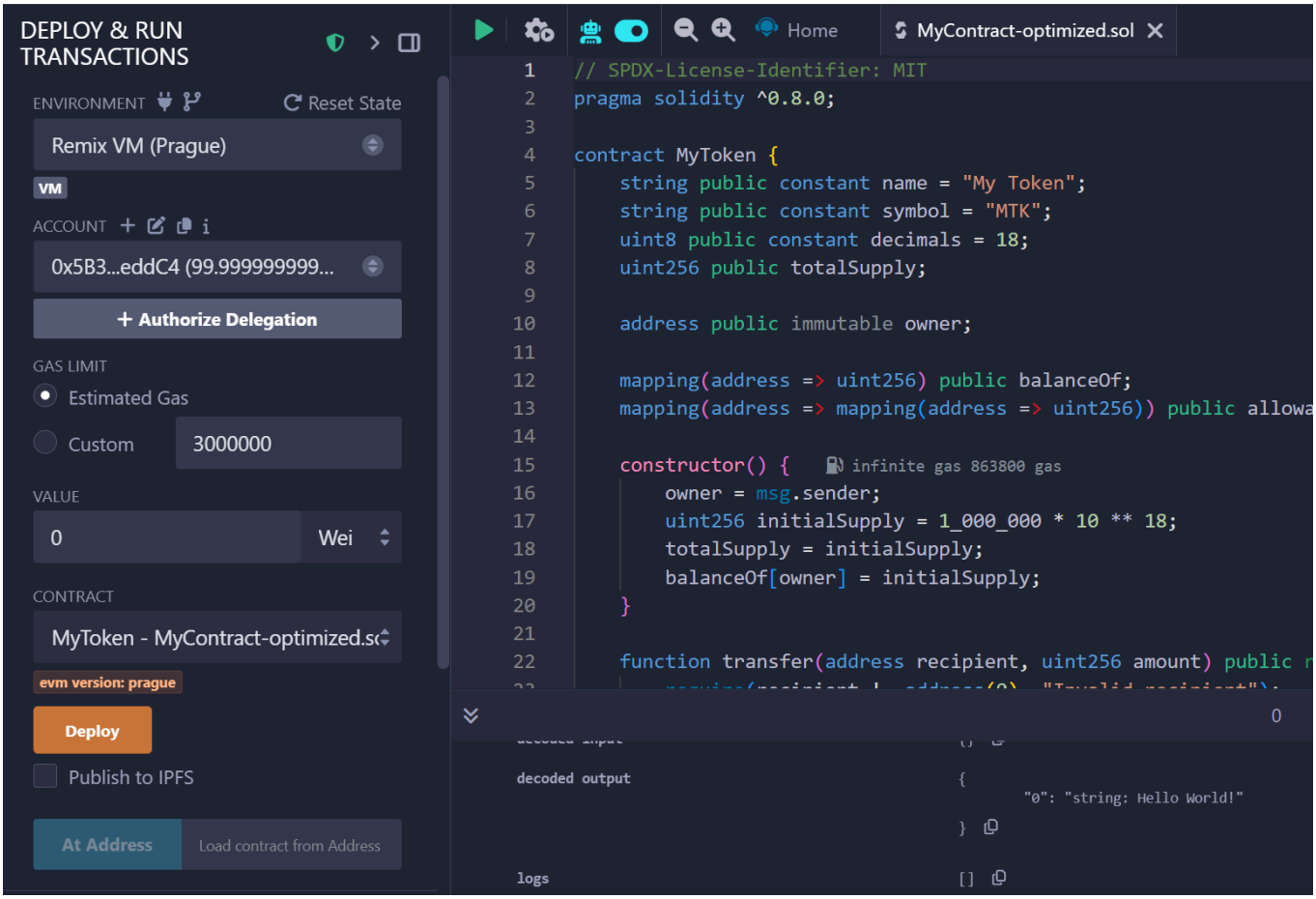
Після оптимізації смартконтракт матиме вигляд:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract MyToken {
    string public constant name = "My Token";
    string public constant symbol = "MTK";
    uint8 public constant decimals = 18;
    uint256 public totalSupply;

    address public immutable owner;

    mapping(address => uint256) public balanceOf;
    mapping(address => mapping(address => uint256)) public allowance;

    constructor() {
        owner = msg.sender;
        uint256 initialSupply = 1_000_000 * 10 ** 18;
        totalSupply = initialSupply;
        balanceOf[owner] = initialSupply;
    }
```

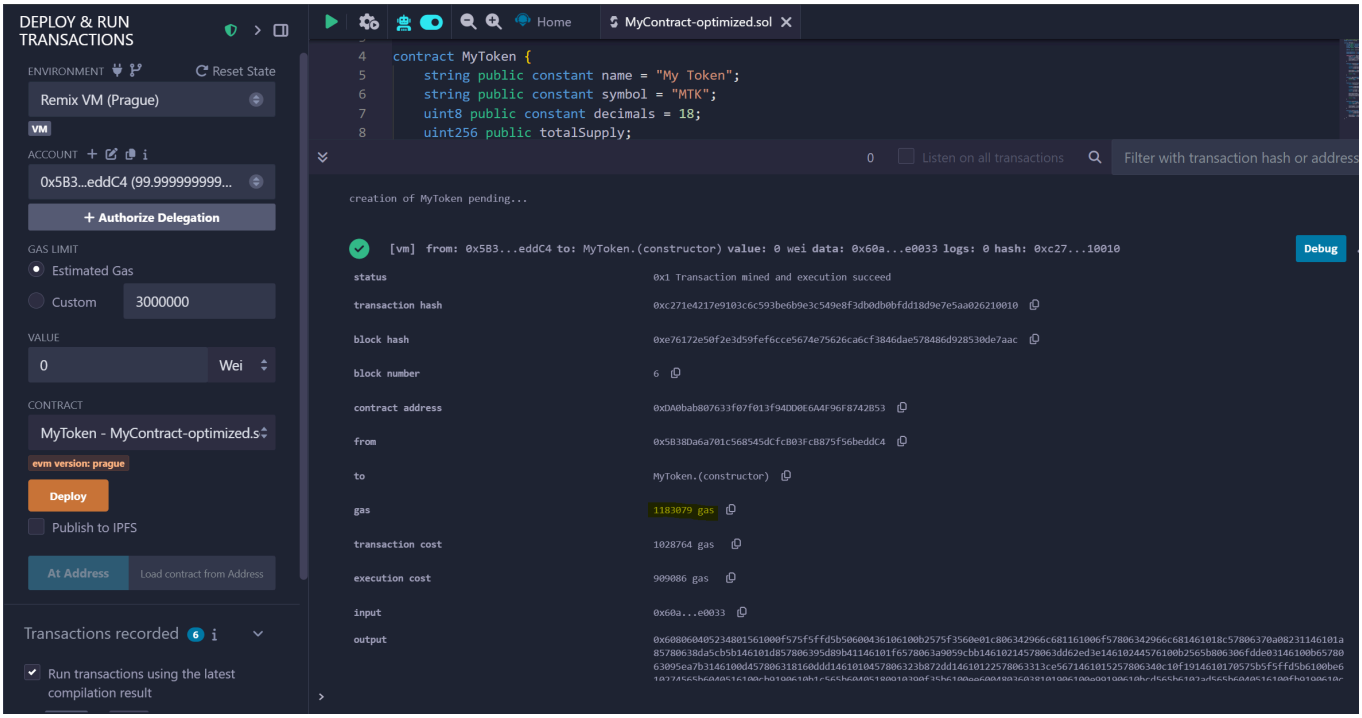
```
    function transfer(address recipient, uint256 amount) public returns (bool) {
        require(recipient != address(0), "Invalid recipient");

        uint256 senderBalance = balanceOf[msg.sender];
        require(senderBalance >= amount, "Not enough balance");

        balanceOf[msg.sender] = senderBalance - amount;
        balanceOf[recipient] += amount;

        return true;
    }

    function approve(address spender, uint256 amount) public returns (bool) {
        allowance[msg.sender][spender] = amount;
        return true;
    }

    function transferFrom(address sender, address recipient, uint256 amount) public
returns (bool) {
        require(recipient != address(0), "Invalid recipient");

        uint256 senderBalance = balanceOf[sender];
        require(senderBalance >= amount, "Not enough balance");

        uint256 allowed = allowance[sender][msg.sender];
        require(allowed >= amount, "Allowance exceeded");

        balanceOf[sender] = senderBalance - amount;
        balanceOf[recipient] += amount;
        allowance[sender][msg.sender] = allowed - amount;

        return true;
    }

    function mint(address account, uint256 amount) public {
        require(msg.sender == owner, "Only owner can mint");
        require(account != address(0), "Invalid account");

        totalSupply += amount;
        balanceOf[account] += amount;
    }

    function burn(uint256 amount) public {
        uint256 senderBalance = balanceOf[msg.sender];
        require(senderBalance >= amount, "Not enough to burn");
```

```
        balanceOf[msg.sender] = senderBalance - amount;
        totalSupply -= amount;
    }
}
```

Зкомпілюємо і задеплоїмо:



У результаті маємо 1183079 gas!



Ми оптимізували смарт-контракт, оголосивши фіксовані значення `name`, `symbol`, `decimals`, `constants` а адресу власника як `immutable`, що дозволяє зберігати їх безпосередньо в байт-коді, а не в дорогих слотах пам'яті, зменшуючи витрати на розгортання і час виконання. Ми також усунули надлишкові обчислення (`10 ** uint256(decimals)`), попередньо обчисливши початковий запас, щоб уникнути непотрібних обчислень під час виконання. Нарешті, ми зменшили кількість повторних зчитувань зі сховища (`balanceOf[msg.sender]`) за рахунок кешування у локальних змінних, оскільки доступ до сховища є значно більш енерговитратним, ніж доступ до пам'яті. Разом ці зміни зменшили витрати газу на розгортання та виконання без зміни функціональності.