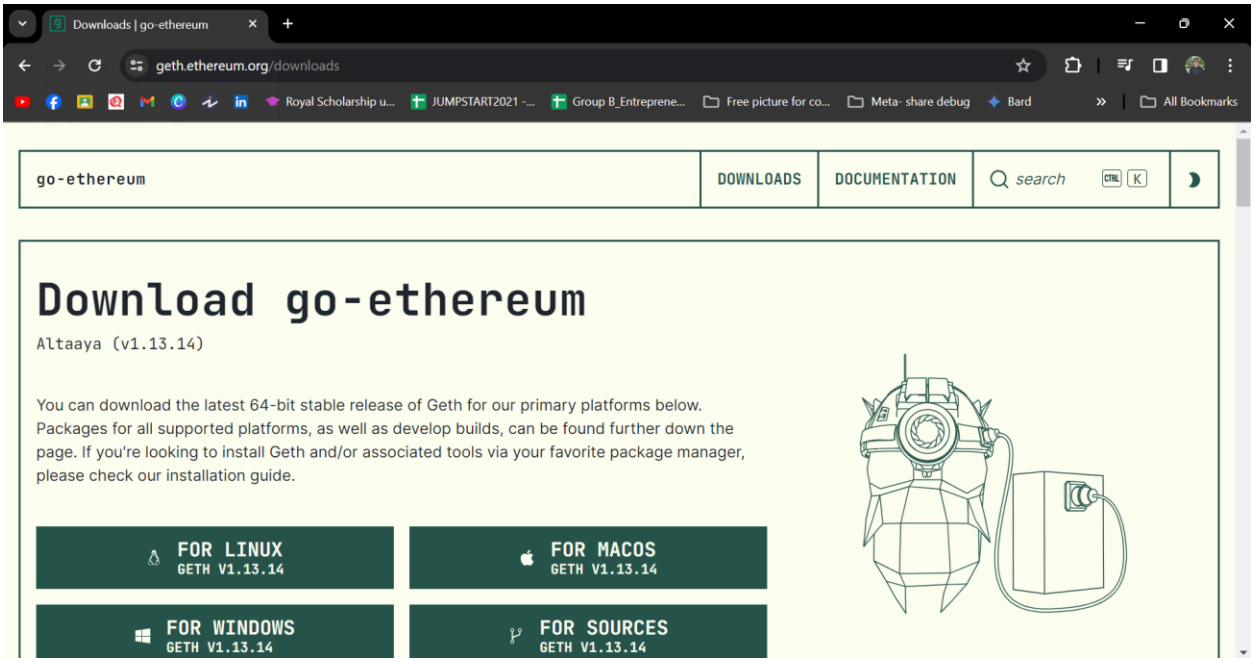
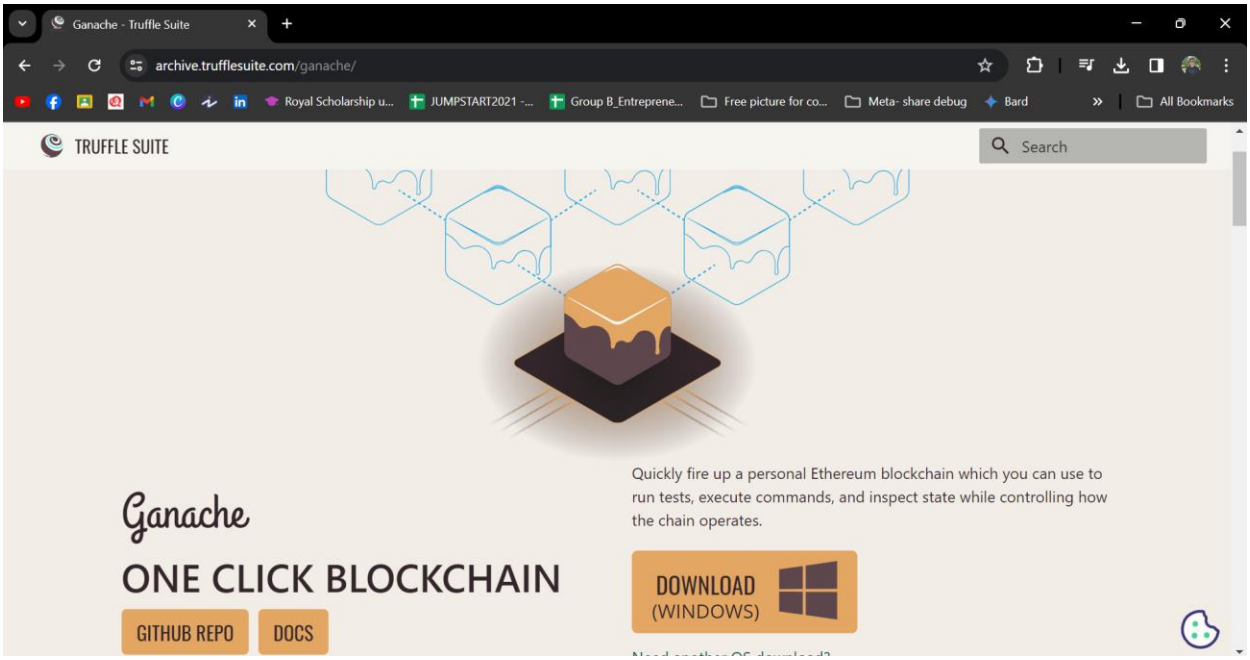
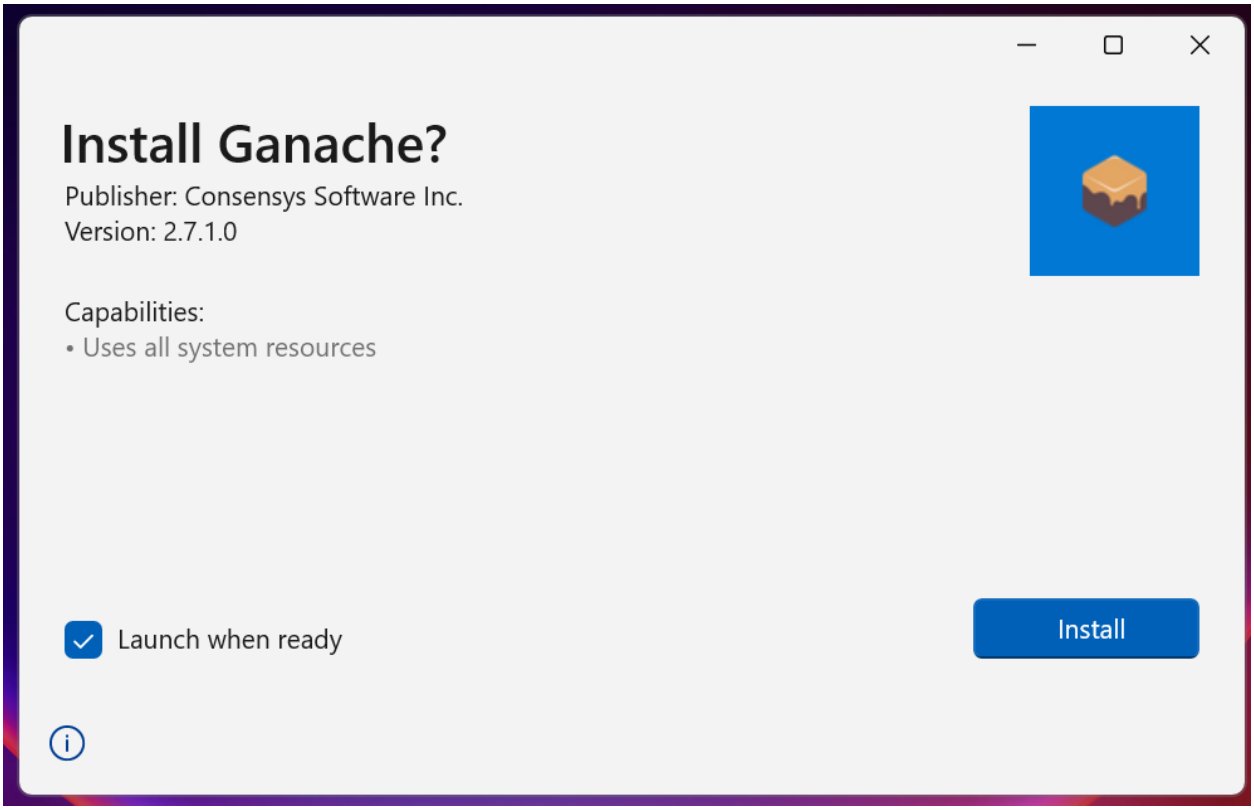
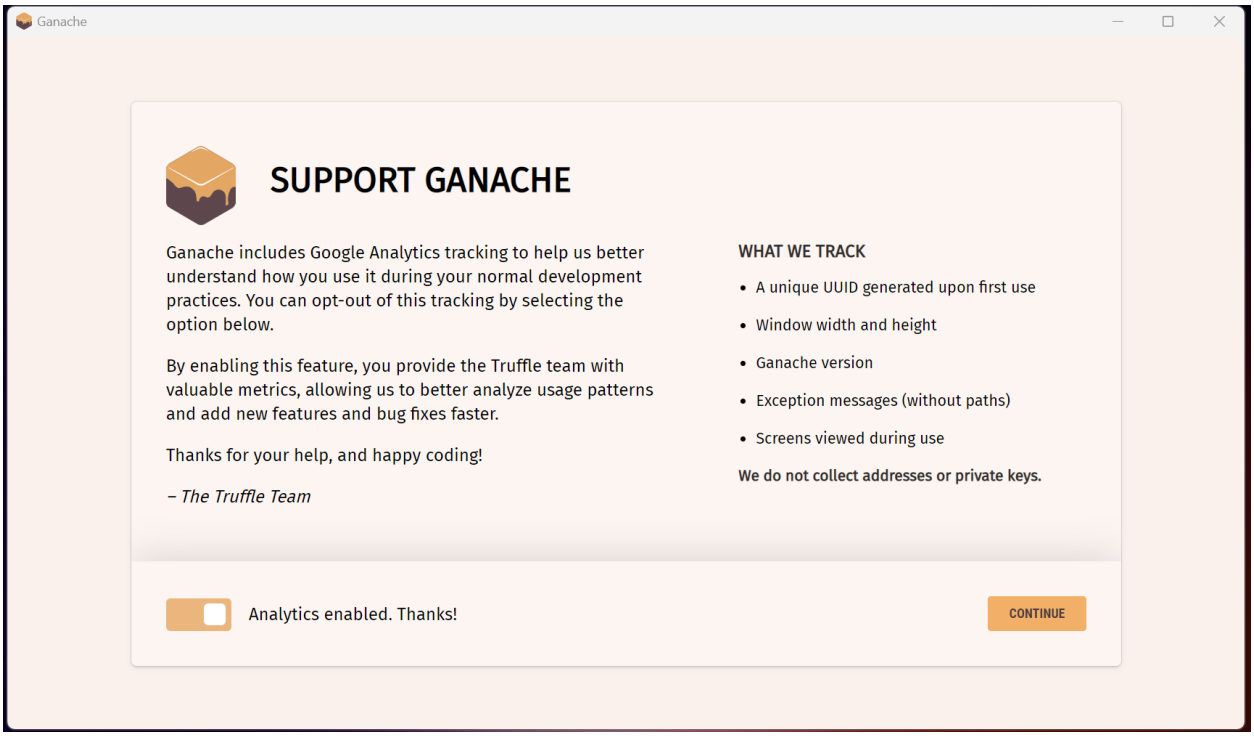
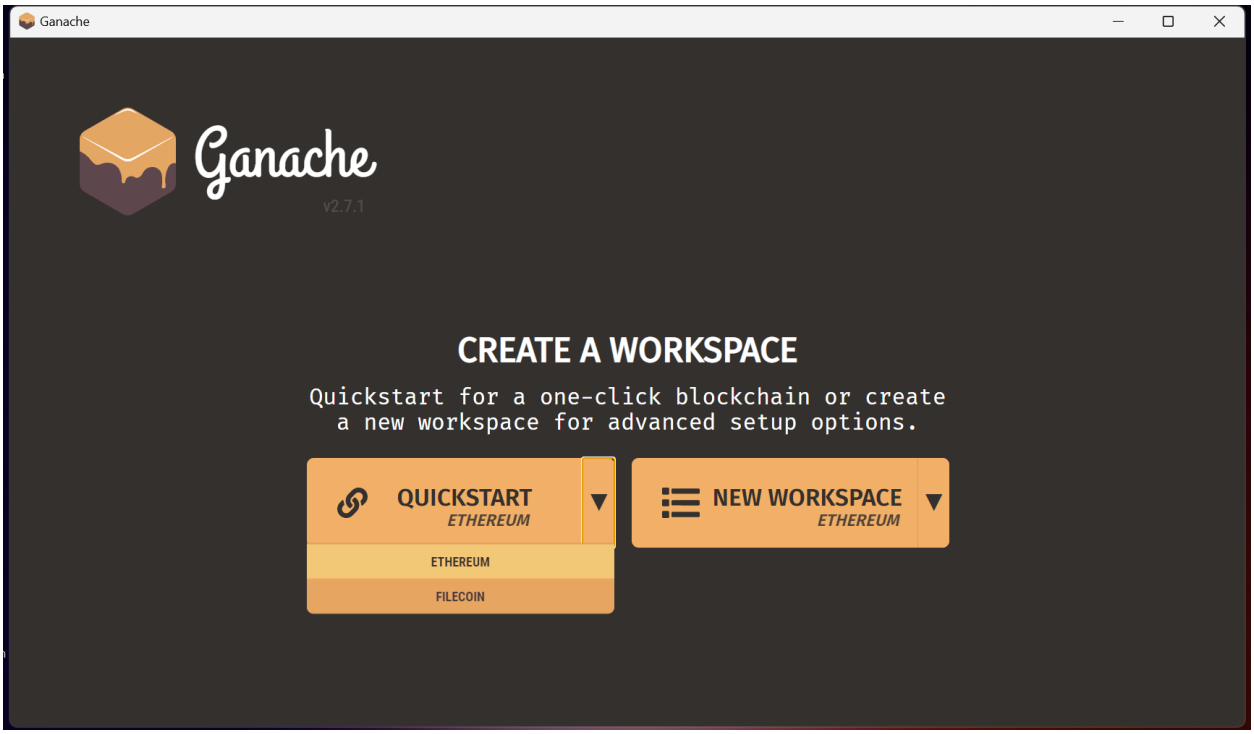
9.3 Create Blockchain with ***Geth*** and ***Ganache***

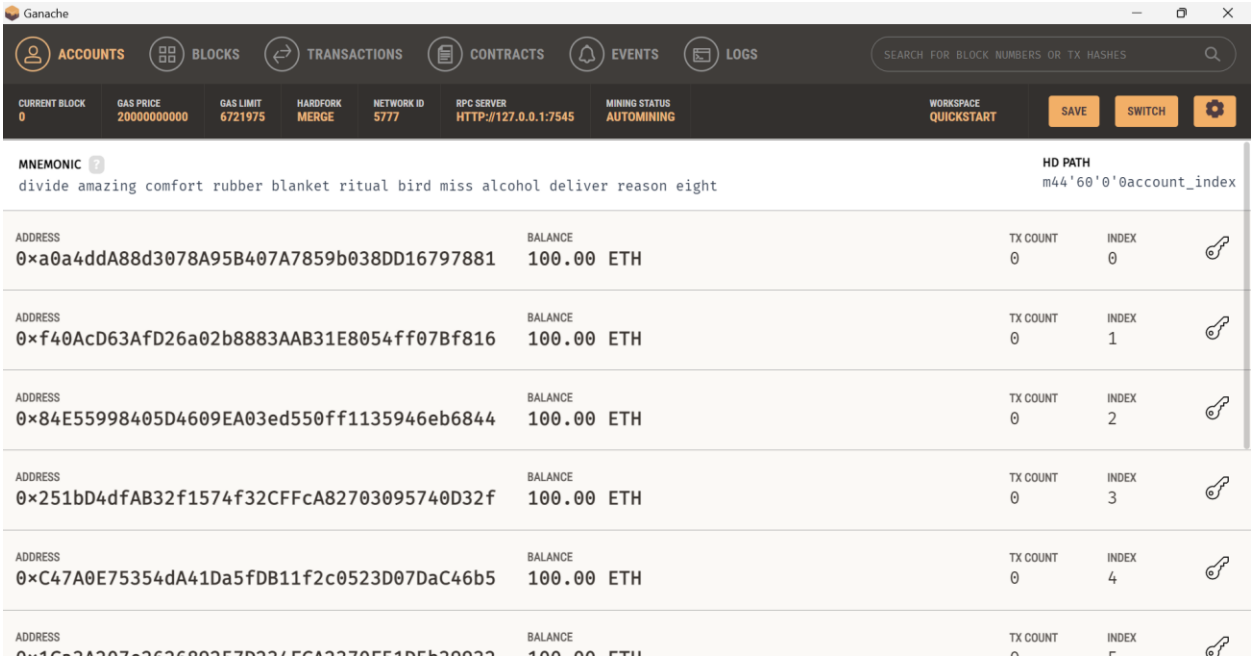


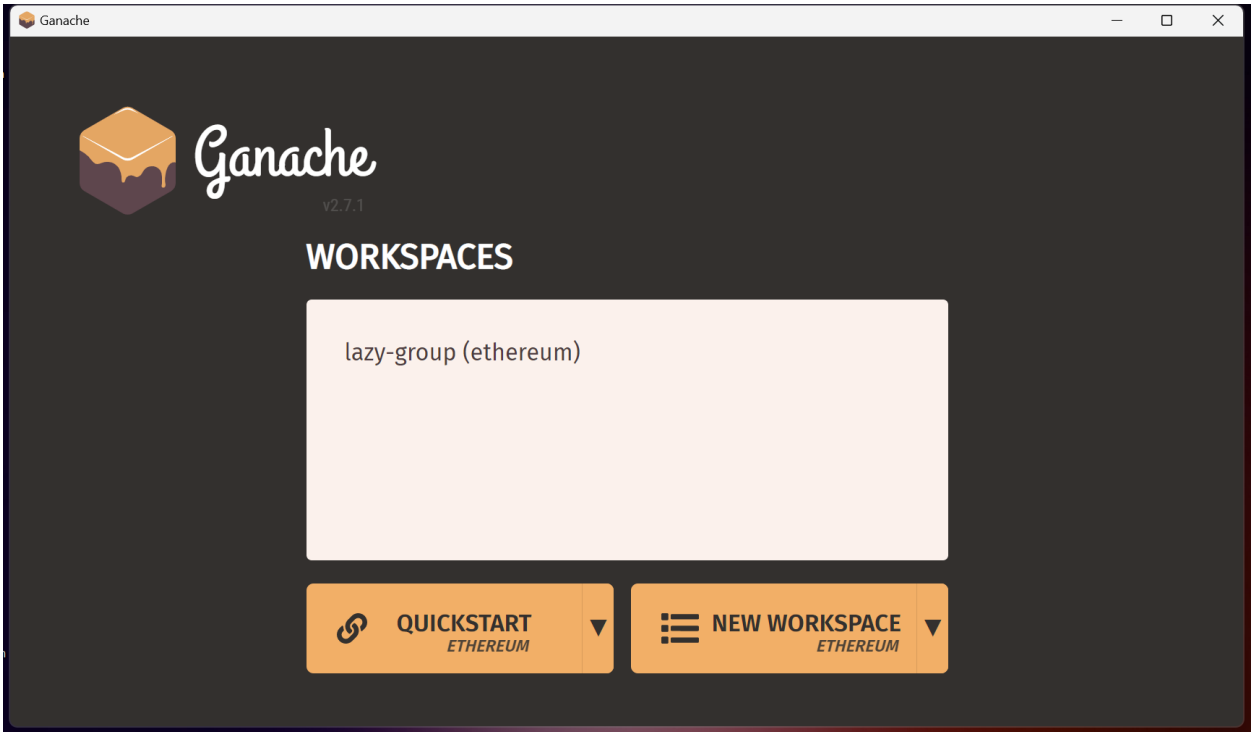



Install Ganache?
Publisher: Consensys Software Inc.
Version: 2.7.1.0
Capabilities:
• Uses all system resources
Launch when ready
Install

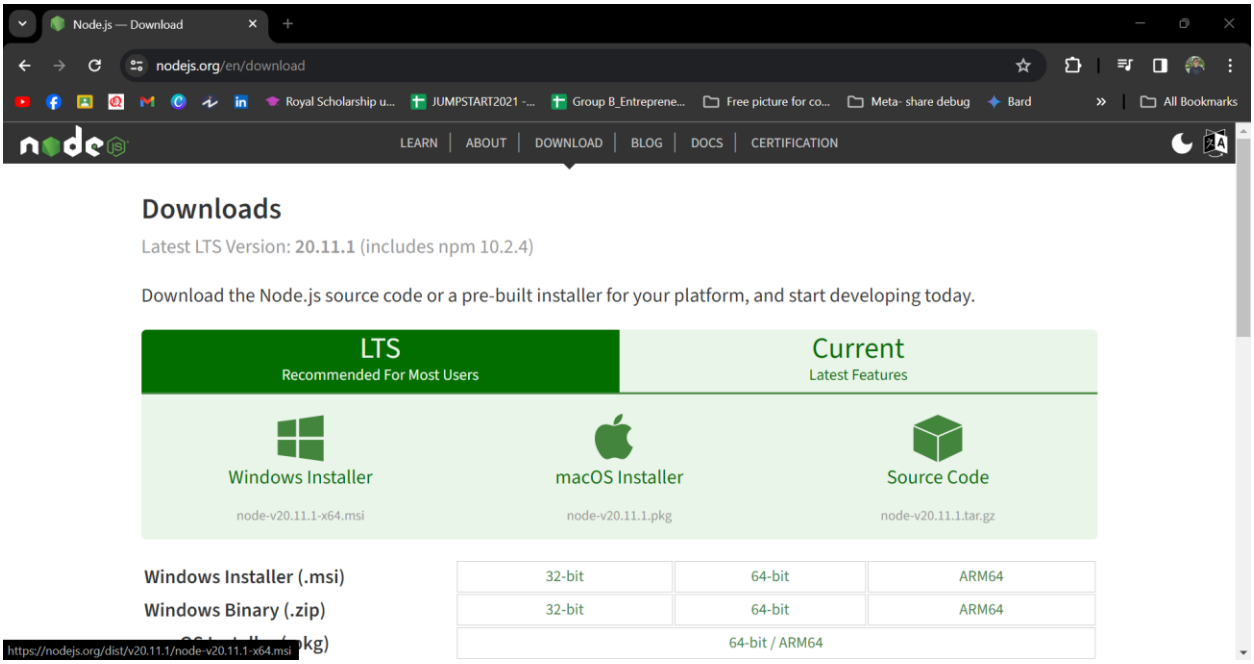

Ganache
SUPPORT GANACHE
Ganache includes Google Analytics tracking to help us better understand how you use it during your normal development practices. You can opt-out of this tracking by selecting the option below.
By enabling this feature, you provide the Truffle team with valuable metrics, allowing us to better analyze usage patterns and add new features and bug fixes faster.
Thanks for your help, and happy coding!
– The Truffle Team
WHAT WE TRACK
• A unique UUID generated upon first use
• Window width and height
• Ganache version
• Exception messages (without paths)
• Screens viewed during use
We do not collect addresses or private keys.
Analytics enabled. Thanks!
CONTINUE

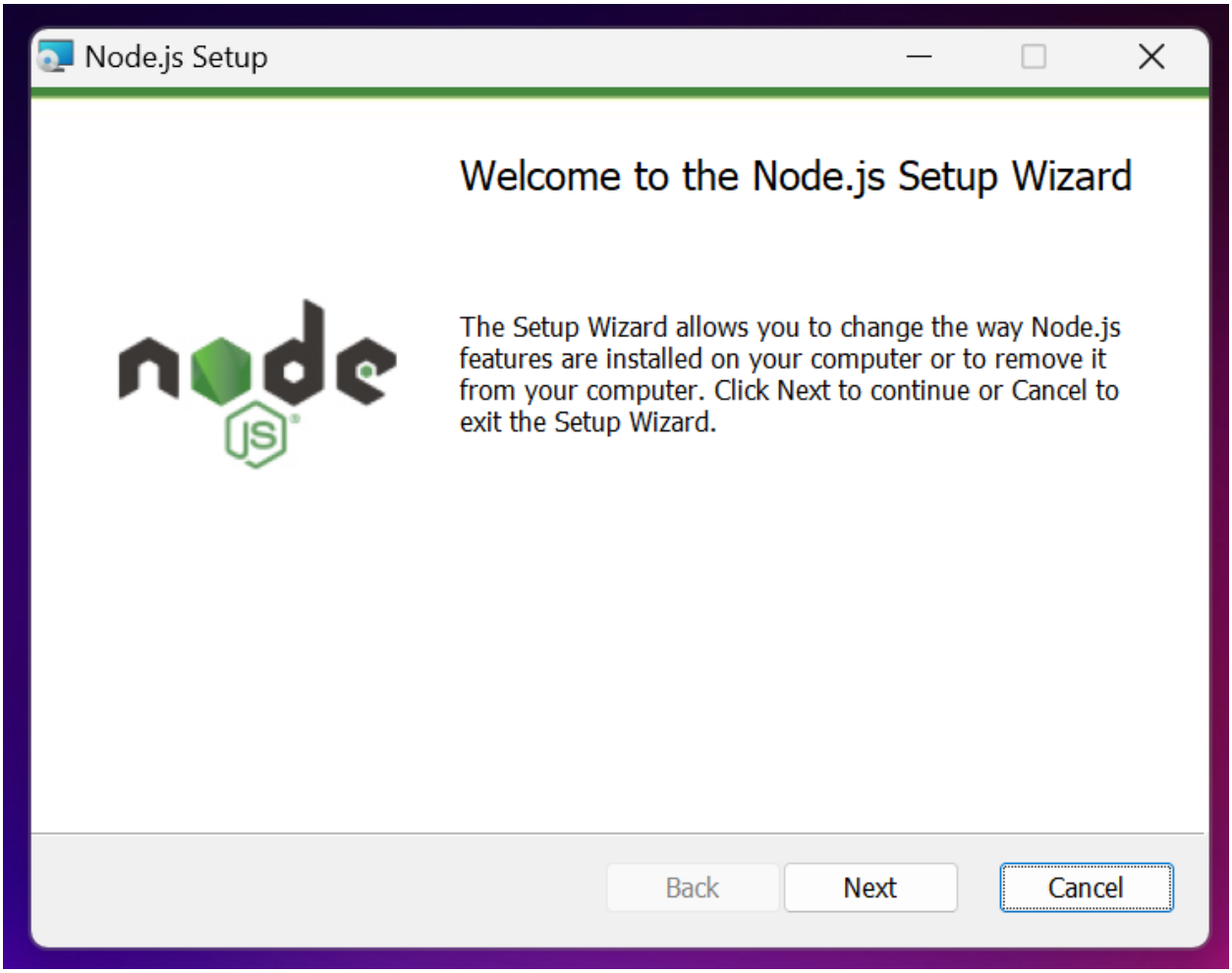

Ganache
Ganache
v2.7.1
CREATE A WORKSPACE
Quickstart for a one-click blockchain or create
a new workspace for advanced setup options.
QUICKSTART
ETHEREUM
ETHEREUM
FILECOIN
NEW WORKSPACE
ETHEREUM


Ganache
ACCOUNTS
BLOCKS
TRANSACTIONS
CONTRACTS
EVENTS
LOGS
SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 0
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK MERGE
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:7545
MINING STATUS AUTOMINING
WORKSPACE QUICKSTART
SAVE
SWITCH
MNEMONIC
divide amazing comfort rubber blanket ritual bird miss alcohol deliver reason eight
HD PATH
m44'60'0'0account_index
ADDRESS 0×a0a4ddA88d3078A95B407A7859b038DD16797881 BALANCE 100.00 ETH TX COUNT 0 INDEX 0
ADDRESS 0×f40AcD63AfD26a02b8883AAB31E8054ff07Bf816 BALANCE 100.00 ETH TX COUNT 0 INDEX 1
ADDRESS 0×84E55998405D4609EA03ed550ff1135946eb6844 BALANCE 100.00 ETH TX COUNT 0 INDEX 2
ADDRESS 0×251bD4dfAB32f1574f32CFFcA82703095740D32f BALANCE 100.00 ETH TX COUNT 0 INDEX 3
ADDRESS 0×C47A0E75354dA41Da5fDB11f2c0523D07DaC46b5 BALANCE 100.00 ETH TX COUNT 0 INDEX 4
ADDRESS BALANCE TX COUNT INDEX

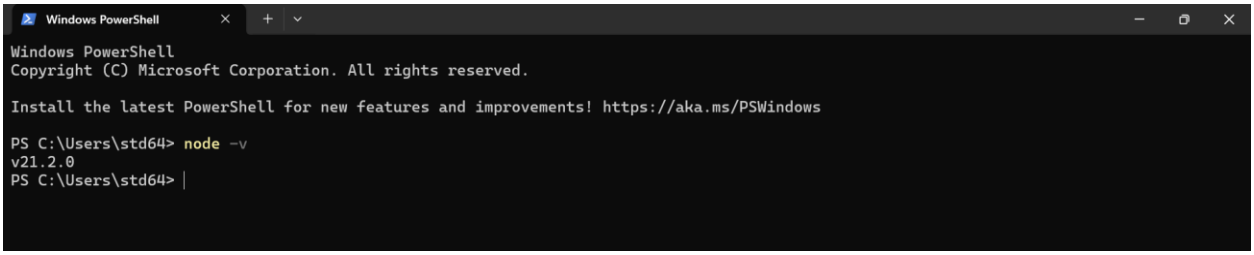


9.4 Build smart contracts with ***Truffle Suite***

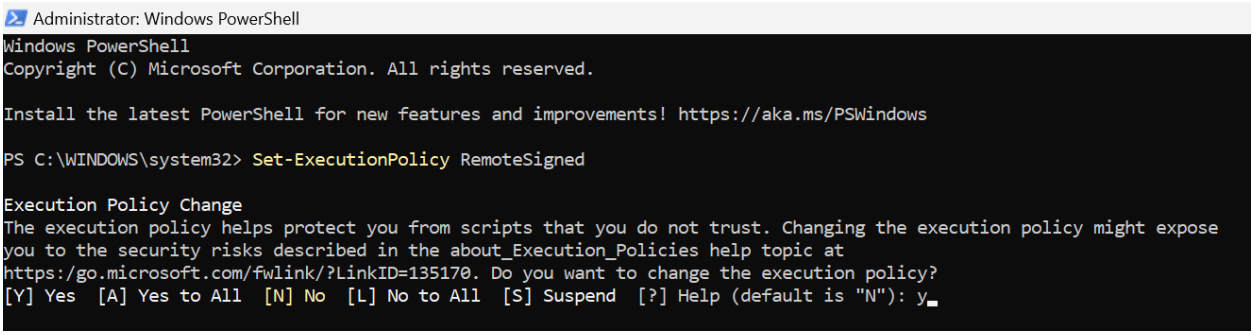




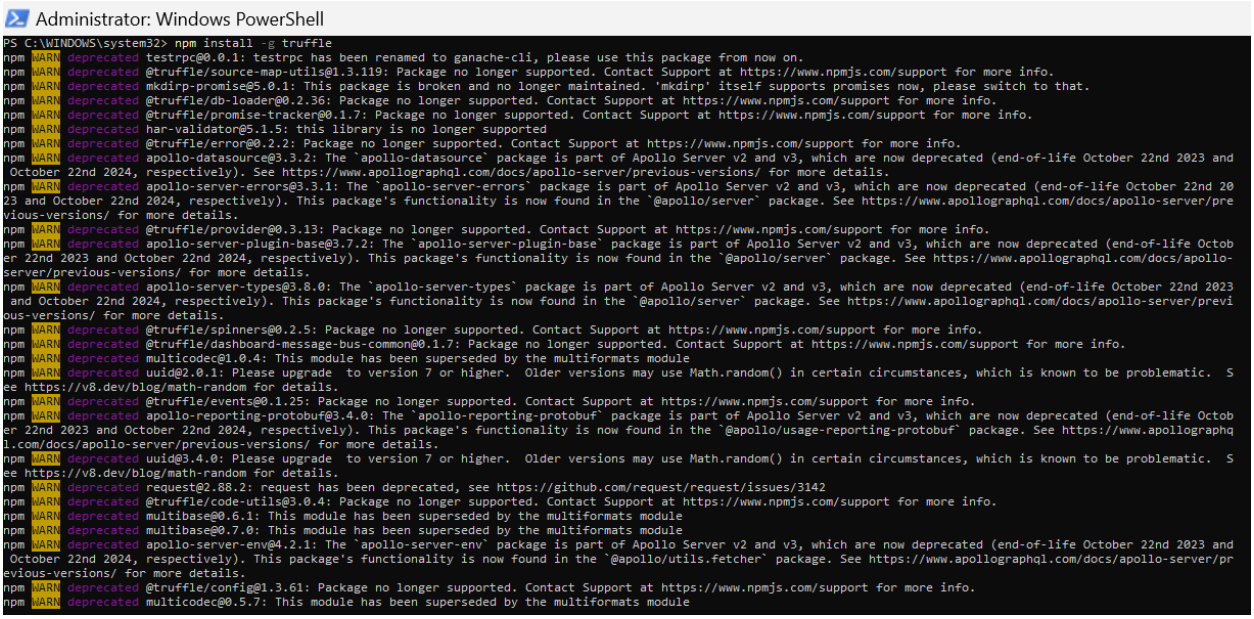
Because I already installed the node, so I prefer to skip the installation guideline.



Allowing *PowerShell* to install program.



Installing *truffle library*



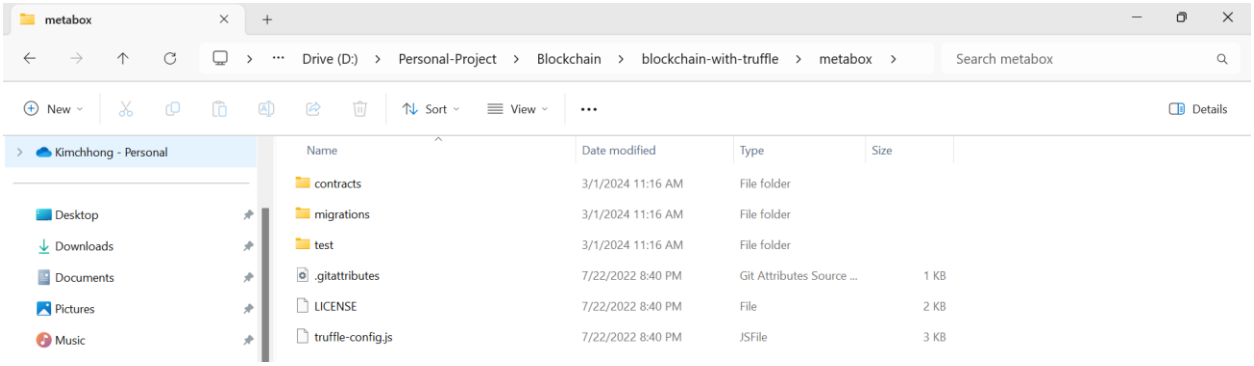
Getting started the *Truffle*

```
Administrator: Windows PowerShell
PS D:\personal-project\blockchain\blockchain-with-truffle> truffle init

Starting init...
================

> Copying project files to D:\personal-project\blockchain\blockchain-with-truffle

Init successful, sweet!

Try our scaffold commands to get started:
  $ truffle create contract YourContractName # scaffold a contract
  $ truffle create test YourTestName         # scaffold a test

http://trufflesuite.com/docs

PS D:\personal-project\blockchain\blockchain-with-truffle>
```

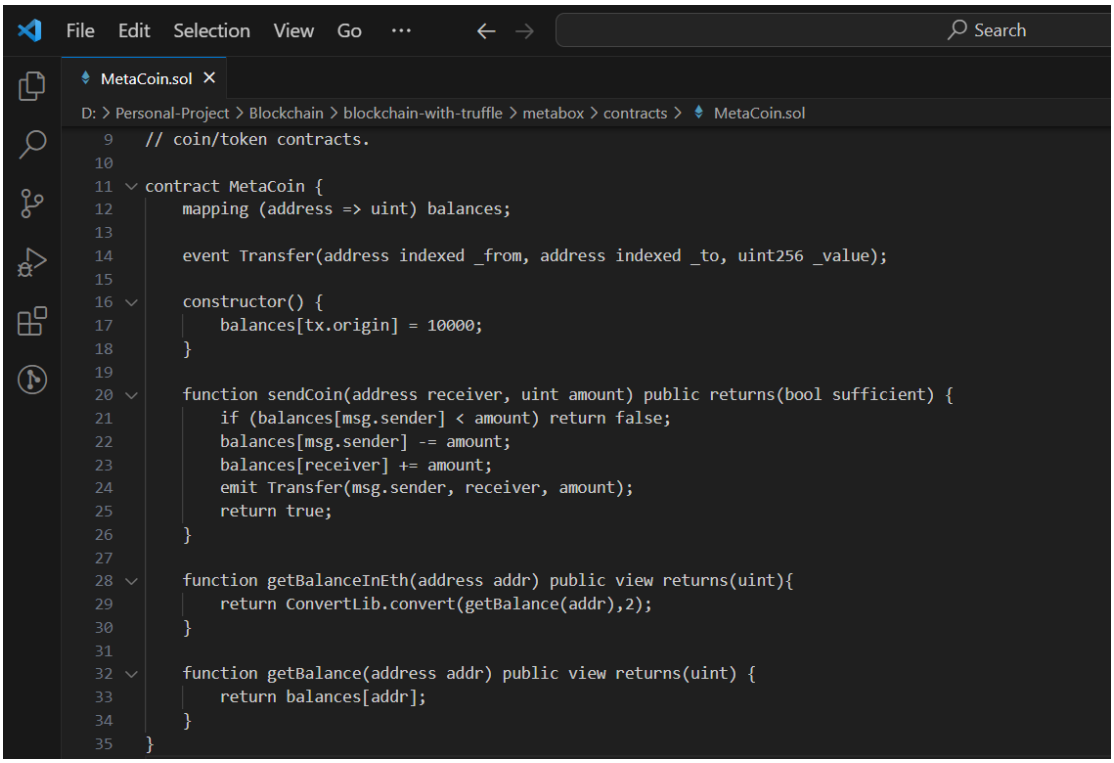
Create metacoin box

```
PS D:\personal-project\blockchain\blockchain-with-truffle> truffle unbox metacoin D:\personal-project\blockchain\blockchain-with-truffle\metabox

Starting unbox...
================

√ Preparing to download box
√ Downloading
√ Cleaning up temporary files
√ Setting up box

Unbox successful, sweet!

Commands:

  Compile contracts: truffle compile
  Migrate contracts: truffle migrate
  Test contracts:    truffle test
```

Folder Structure of MetaCoin box



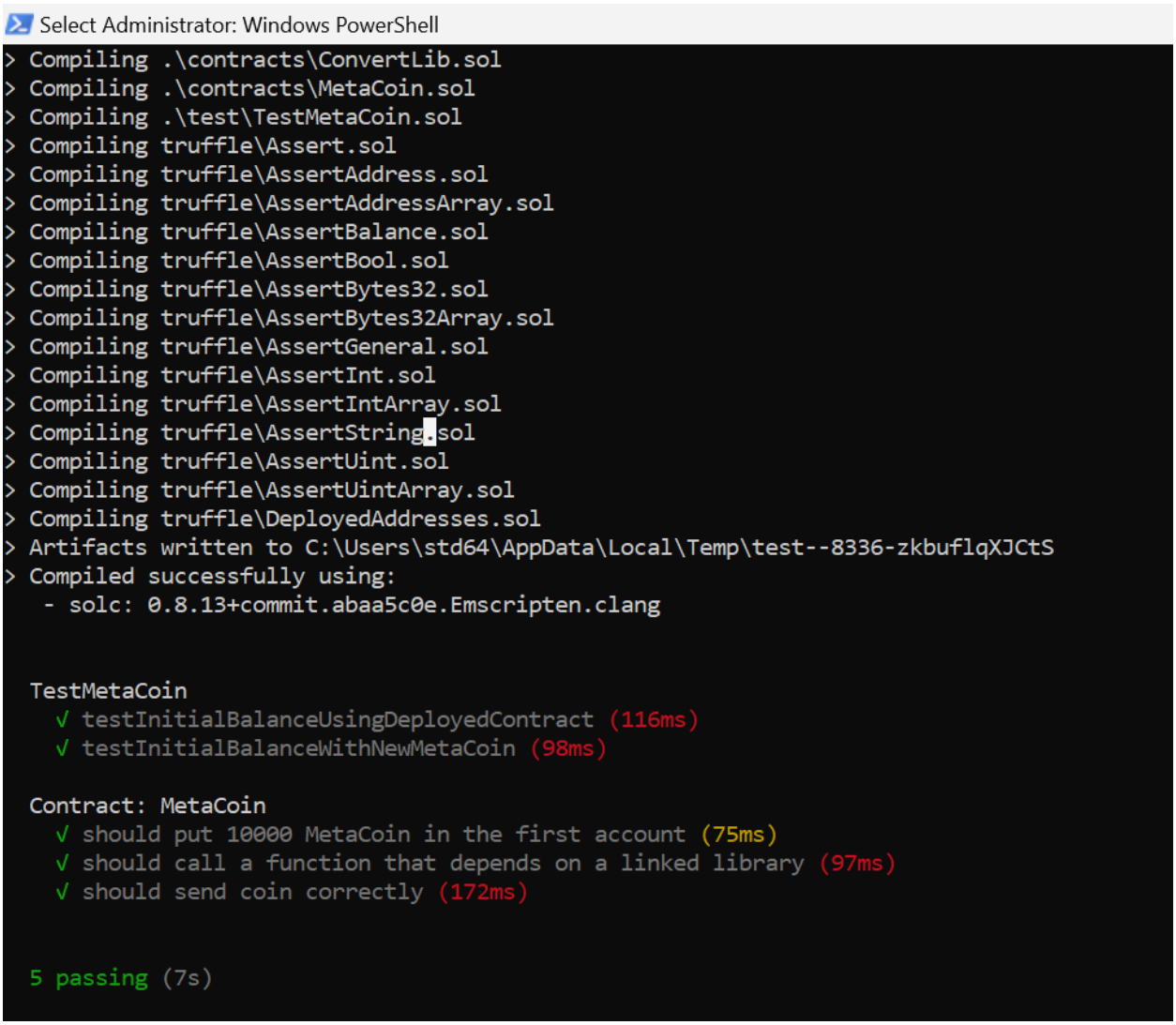
D:\Personal-Project\Blockchain\blockchain-with-truffle\metabox\contracts\MetaCoin.sol



Truffle compile

```
PS D:\personal-project\blockchain\blockchain-with-truffle\metabox> truffle  compile

Compiling your contracts...
===========================
√ Fetching solc version list from solc-bin. Attempt #1
√ Downloading compiler. Attempt #1.
> Compiling .\contracts\ConvertLib.sol
> Compiling .\contracts\MetaCoin.sol
> Artifacts written to D:\personal-project\blockchain\blockchain-with-truffle\metabox\build\contracts
> Compiled successfully using:
   - solc: 0.8.13+commit.abaa5c0e.Emscripten.clang
```

Truffle test command



Config *truffle-config.js*

```
module.exports = {
  networks: {
    development: {
     host: "127.0.0.1",     // Localhost (default: none)
     port: 8545,            // Standard Ethereum port (default: none)
     network_id: "*",       // Any network (default: none)
    },
    // goerli: {
    //   provider: () => new HDWalletProvider(mnemonic, `https://goerli.infura.io/v3/${infuraProjectId}`),
    //   network_id: 5,       // Goerli's id
    //   chain_id: 5
    // }
  },

  // Set default mocha options here, use special reporters etc.
  mocha: {
    // timeout: 100000
  },

  // Configure your compilers
  compilers: {
    solc: {
      version: "0.8.13",      // Fetch exact version from solc-bin
    }
  }
};

```

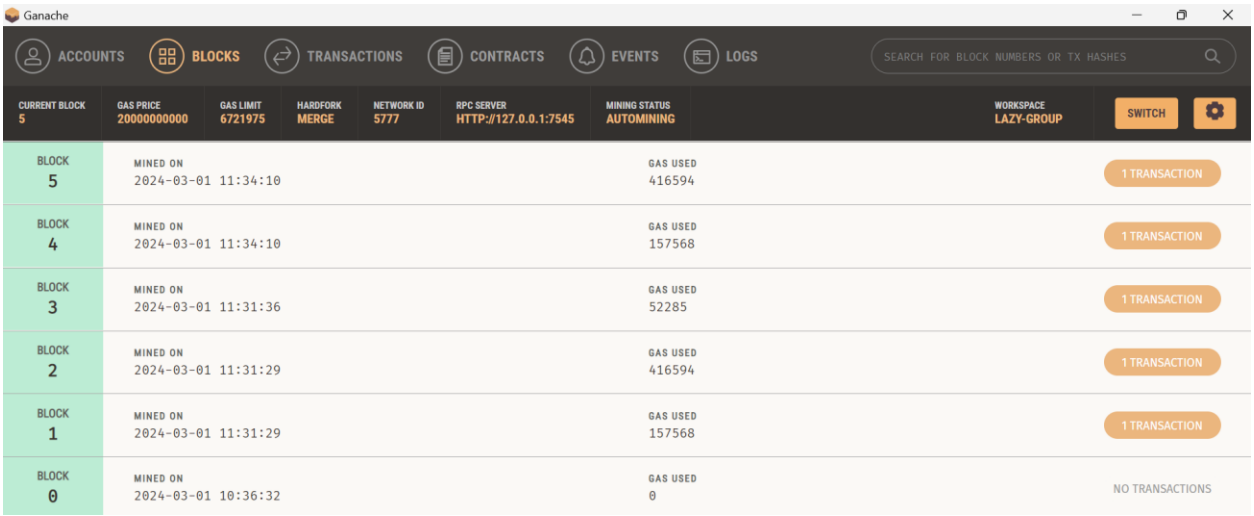
Truffle compile command

```
Administrator: Windows PowerShell
PS D:\personal-project\blockchain\blockchain-with-truffle\metabox> truffle compile

Compiling your contracts...
===========================
> Compiling .\contracts\ConvertLib.sol
> Compiling .\contracts\MetaCoin.sol
> Artifacts written to D:\personal-project\blockchain\blockchain-with-truffle\metabox\build\contracts
> Compiled successfully using:
   - solc: 0.8.13+commit.abaa5c0e.Emscripten.clang
```

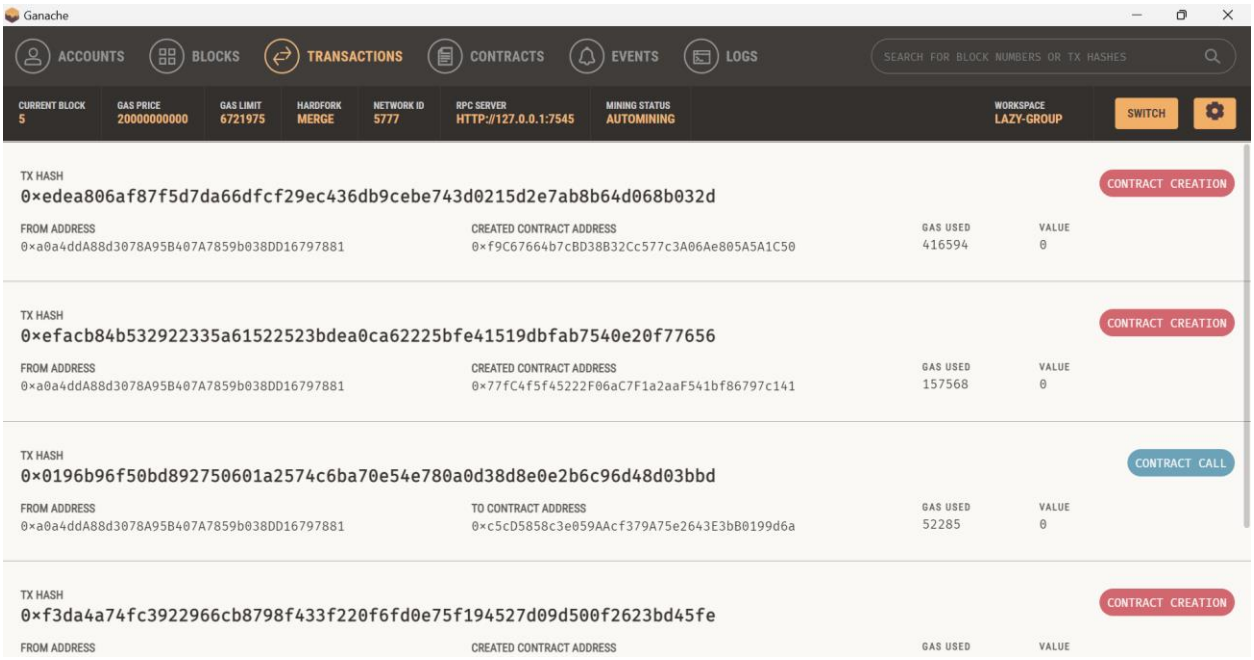
Truffle migrate

```
Administrator: Windows PowerShell
   Deploying 'MetaCoin'
   --------------------
   > transaction hash:    0xedea806af87f5d7da66dfcf29ec436db9cebe743d0215d2e7ab8b64d068b032d
   > Blocks: 0            Seconds: 0
   > contract address:    0xf9C67664b7cBD38B32Cc577c3A06Ae805A5A1C50
   > block number:        5
   > block timestamp:     1709267650
   > account:             0xa0a4ddA88d3078A95B407A7859b038DD16797881
   > balance:             99.996187919271288746
   > gas used:            416594 (0x65b52)
   > gas price:           3.030182049 gwei
   > value sent:          0 ETH
   > total cost:          0.001262355660521106 ETH

   > Saving artifacts
   -------------------------------------
   > Total cost:     0.00175111446304949 ETH

Summary
=======
> Total deployments:   2
> Final cost:          0.00175111446304949 ETH
```

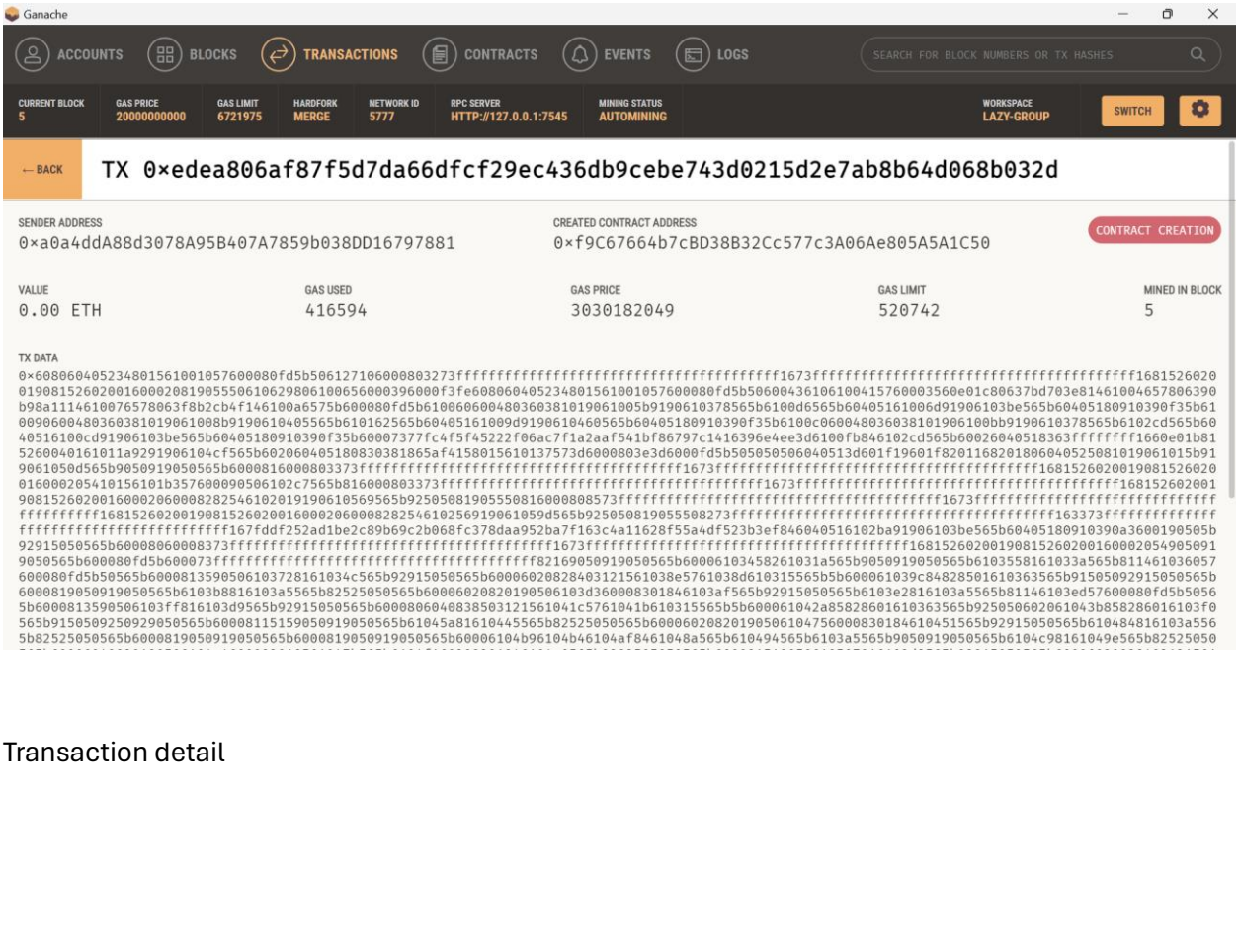
Set up smart contracts successfully



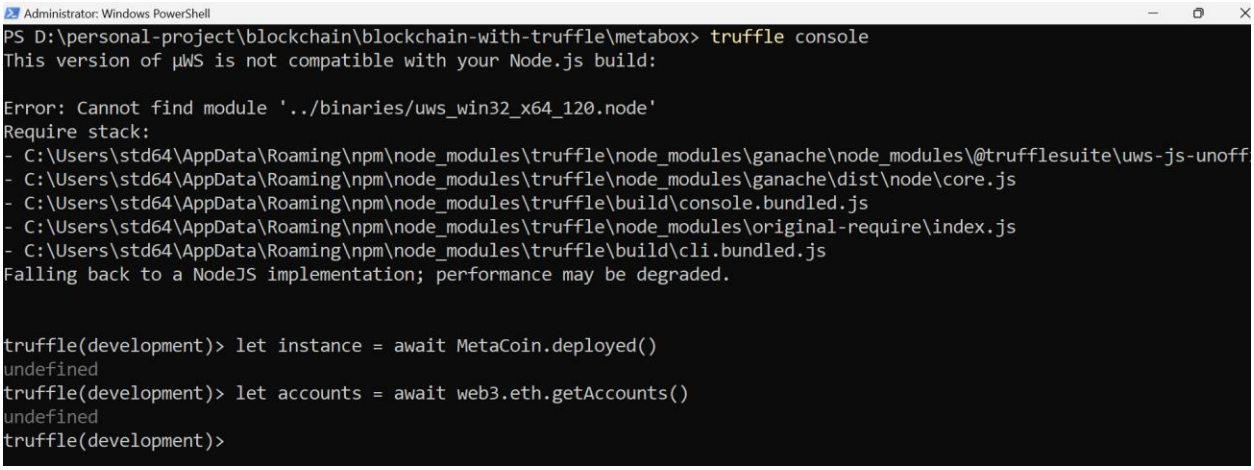
Set up smart contracts in blockchain successfully



Display transaction report that happen after created smart contracts



Transaction detail



let balance = await instance.getBalance(accounts[0])

balance.toNumber()

let ether = await instance.getBalanceinEth(accounts[0])

ether.toNumber()

```
truffle(development)> let instance = await MetaCoin.deployed()
undefined
truffle(development)> let accounts = await web3.eth.getAccounts()
undefined
truffle(development)> let balance = await instance.getBalance(accounts[0])
undefined
truffle(development)> balance.toNumber()
10000
```

```
truffle(development)> let ether = await instance.getBalanceInEth(accounts[0])
undefined
truffle(development)>
```

Instance.sendCoin(accounts[1], 500)

```
Administrator: Windows PowerShell
truffle(development)> instance.sendCoin(accounts[1], 500)
{
  tx: '0x95a322a973c5d2e19daf8c8a9e92660cdf47ff4e97787184aef3a44a3c552212',
  receipt: {
    transactionHash: '0x95a322a973c5d2e19daf8c8a9e92660cdf47ff4e97787184aef3a44a3c552212',
    transactionIndex: 0,
    blockNumber: 6,
    blockHash: '0xefcb3f0456b0f3ad8020360becae6c8cbd7a49ef214005af2a04be143888b3d8',
    from: '0xa0a4dda88d3078a95b407a7859b038dd16797881',
    to: '0xf9c67664b7cbd38b32cc577c3a06ae805a5a1c50',
    cumulativeGasUsed: 52297,
    gasUsed: 52297,
    contractAddress: null,
    logs: [ [Object] ],
    logsBloom: '0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000010000000820000000000000000000000000000000000000000000000000000000000000000000000000000
00000000001000000000000000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000600000000001000000000000000000000000000000000000000000000000000000000000
00080000000000004000000000000000000000000000000000000000',
    status: true,
    effectiveGasPrice: 2972123795,
    type: '0x2',
    rawLogs: [ [Object] ]
  },
  logs: [
    {
      address: '0xf9C67664b7cBD38B32Cc577c3A06Ae805A5A1C50',
      blockHash: '0xefcb3f0456b0f3ad8020360becae6c8cbd7a49ef214005af2a04be143888b3d8',
      blockNumber: 6,
      logIndex: 0,
      removed: false,
      transactionHash: '0x95a322a973c5d2e19daf8c8a9e92660cdf47ff4e97787184aef3a44a3c552212',
      transactionIndex: 0,
```

Let received = await instance.getBalance(accounts[1])

Received.toNumber()

Let newBalance = await instance.getBalance(accounts[0])

newBalance.toNumber()

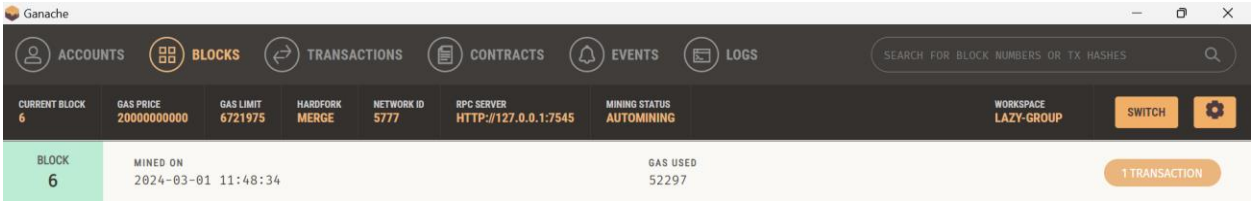
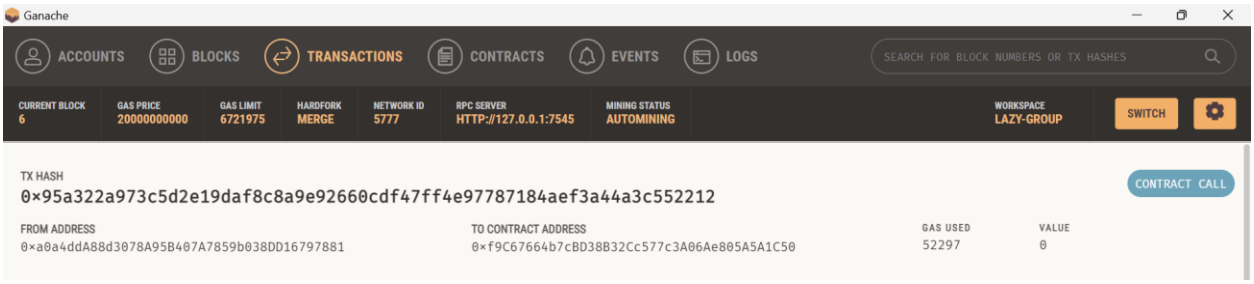
```
truffle(development)> let recieved = await instance.getBalance(accounts[1])
undefined
```

```
truffle(development)> recieved.toNumber()
500
```

```
truffle(development)> let newBalance = await instance.getBalance(accounts[0])
undefined
```

```
truffle(development)> newBalance.toNumber()
9500
```

New block that created after transaction



Contract call type