Lottery Project Compiling: A Complete Tutorial

No comments

In this tutorial, We are going to finally compile the lottery project. After making sure that it has been successfully compiled, we will run all the written tests to make sure everything works correctly in the contract. In the end, we will deploy the Decentralized Application on the Rinkeby test network. To be able to interact with the deployed contract, we will get some test Link tokens from Rinkeby Faucet.

Compiling the Lottery Project Dapp

In this part, we are going to finally compile, run and test our Lottery Dapp. `brownie compile` The result should be:

```
Brownie v1.18.1 - Python development framework for Ethereum1.33MiB [00:03, 402kiB/s] New compatible solc version available: 0.7.0 Compiling contracts... Solc version: 0.7.0 Optimizer: Enabled Runs: 200 EVM Version: Istanbul WARNING: Unable to compile OpenZeppelin/openzeppelin-contracts@3.4.0 due to a CompilerError - you may still be able to import sources from the package, but will be unable to load the package directly.Downloading from https://solc-bin.ethereum.org/linux-amd64/solc-linux-amd64-v0.4.26+commit.4563c3fc
```

```
100%|????????????????????????????????????????????????????????????????????????????|
5.24M/5.24M [00:10<00:00, 498kiB/s] solc 0.4.26 successfully installed
at: /home/mohamad/.solcx/solc-v0.4.26 Downloading from https://solc-
bin.ethereum.org/linux-amd64/solc-linux-amd64-v0.6.6+commit.6c089d02
100%|????????????????????????????????????????????????????????????????????????????|
8.92M/8.92M [00:23<00:00, 378kiB/s] solc 0.6.6 successfully installed
at: /home/mohamad/.solcx/solc-v0.6.6 Compiling contracts... Solc
version: 0.6.12 Optimizer: Enabled Runs: 200 EVM Version: Istanbul
Generating build data... - OpenZeppelin/openzeppelin-
contracts@3.4.0/Ownable - OpenZeppelin/openzeppelin-
contracts@3.4.0/Context - smartcontractkit/chainlink-brownie-
contracts@1.1.1/LinkTokenReceiver - smartcontractkit/chainlink-
brownie-contracts@1.1.1/VRFConsumerBase - smartcontractkit/chainlink-
brownie-contracts@1.1.1/VRFRequestIDBase - smartcontractkit/chainlink-
brownie-contracts@1.1.1/AggregatorInterface -
smartcontractkit/chainlink-brownie-
contracts@1.1.1/AggregatorV2V3Interface - smartcontractkit/chainlink-
brownie-contracts@1.1.1/AggregatorV3Interface -
smartcontractkit/chainlink-brownie-
contracts@1.1.1/ChainlinkRequestInterface -
smartcontractkit/chainlink-brownie-contracts@1.1.1/LinkTokenInterface
- smartcontractkit/chainlink-brownie-contracts@1.1.1/SafeMathChainlink
- Lottery - MockOracle - MockV3AggregatorCompiling contracts... Solc
version: 0.4.26 Optimizer: Enabled Runs: 200 EVM Version: Byzantium
Generating build data... - smartcontractkit/chainlink-brownie-
contracts@1.1.1/ERC677Token - smartcontractkit/chainlink-brownie-
contracts@1.1.1/ERC20 - smartcontractkit/chainlink-brownie-
contracts@1.1.1/ERC20Basic - smartcontractkit/chainlink-brownie-
contracts@1.1.1/ERC677 - smartcontractkit/chainlink-brownie-
contracts@1.1.1/ERC677Receiver - smartcontractkit/chainlink-brownie-
contracts@1.1.1/BasicToken - smartcontractkit/chainlink-brownie-
contracts@1.1.1/SafeMathChainlink - smartcontractkit/chainlink-
brownie-contracts@1.1.1/StandardToken - LinkTokenCompiling
contracts... Solc version: 0.6.6 Optimizer: Enabled Runs: 200 EVM
Version: Istanbul Generating build data... -
smartcontractkit/chainlink-brownie-contracts@1.1.1/VRFConsumerBase -
smartcontractkit/chainlink-brownie-contracts@1.1.1/VRFRequestIDBase -
smartcontractkit/chainlink-brownie-contracts@1.1.1/LinkTokenInterface
- smartcontractkit/chainlink-brownie-contracts@1.1.1/SafeMathChainlink
- VRFCoordinatorMockGenerating interface ABIs... Project has been
compiled. Build artifacts saved at
/home/mohamad/smartcontract_lottery/build/contracts
```

Applying the Tests

Now, we apply our tests with following command in the terminal: `brownie test -k test_get_entrance_fee` Result:

```
Brownie v1.18.1 - Python development framework for EthereumPython-dotenv could not parse statement starting at line 1 Python-dotenv could not parse statement starting at line 1
============================================= test session starts
============================================= platform linux --
```

```
Python 3.8.10, pytest-6.2.5, py-1.11.0, pluggy-1.0.0 rootdir:
/home/mohamad/smartcontract_lottery plugins: eth-brownie-1.18.1, web3-
5.27.0, xdist-1.34.0, forked-1.4.0, hypothesis-6.27.3 collected 6
items / 5 deselected / 1 selectedLaunching 'ganache-cli --
chain.vmErrorsOnRPCResponse true --wallet.totalAccounts 10 --hardfork
istanbul --miner.blockGasLimit 12000000 --wallet.mnemonic brownie --
server.port 8545'...tests/test_lottery_unit.py .
[100%]========================================= 1 passed, 5 deselected
in 2.42s ========================================= Terminating local
RPC client...
```

Test Number 2

```
brownie test -k test_cant_enter_unless_started  Result:   Brownie v1.18.1
- Python development framework for EthereumPython-dotenv could not
parse statement starting at line 1 Python-dotenv could not parse
statement starting at line 1
============================================= test session starts
============================================= platform linux --
Python 3.8.10, pytest-6.2.5, py-1.11.0, pluggy-1.0.0 rootdir:
/home/mohamad/smartcontract_lottery plugins: eth-brownie-1.18.1, web3-
5.27.0, xdist-1.34.0, forked-1.4.0, hypothesis-6.27.3 collected 6
items / 5 deselected / 1 selectedLaunching 'ganache-cli --
chain.vmErrorsOnRPCResponse true --wallet.totalAccounts 10 --hardfork
istanbul --miner.blockGasLimit 12000000 --wallet.mnemonic brownie --
server.port 8545'...tests/test_lottery_unit.py .
[100%]========================================= 1 passed, 5 deselected
in 2.10s ========================================= Terminating local
RPC client...
```

Test Number 3

```
brownie test -k test_can_start_and_enter_lottery  Result:   Brownie
v1.18.1 - Python development framework for EthereumPython-dotenv could
not parse statement starting at line 1 Python-dotenv could not parse
statement starting at line 1
============================================= test session starts
============================================= platform linux --
Python 3.8.10, pytest-6.2.5, py-1.11.0, pluggy-1.0.0 rootdir:
/home/mohamad/smartcontract_lottery plugins: eth-brownie-1.18.1, web3-
5.27.0, xdist-1.34.0, forked-1.4.0, hypothesis-6.27.3 collected 6
items / 5 deselected / 1 selectedLaunching 'ganache-cli --
chain.vmErrorsOnRPCResponse true --wallet.totalAccounts 10 --hardfork
istanbul --miner.blockGasLimit 12000000 --wallet.mnemonic brownie --
server.port 8545'...tests/test_lottery_unit.py .
[100%]========================================= 1 passed, 5 deselected
in 2.10s ========================================= Terminating local
RPC client...
```

Test Number 4

```
brownie test -k test_can_end_lottery Result:   Brownie v1.18.1 - Python development framework for EthereumPython-dotenv could not parse statement starting at line 1 Python-dotenv could not parse statement starting at line 1 ========================================= test session starts ========================================= platform linux -- Python 3.8.10, pytest-6.2.5, py-1.11.0, pluggy-1.0.0 rootdir: /home/mohamad/smartcontract_lottery plugins: eth-brownie-1.18.1, web3-5.27.0, xdist-1.34.0, forked-1.4.0, hypothesis-6.27.3 collected 6 items / 5 deselected / 1 selectedLaunching 'ganache-cli --chain.vmErrorsOnRPCResponse true --wallet.totalAccounts 10 --hardfork istanbul --miner.blockGasLimit 12000000 --wallet.mnemonic brownie --server.port 8545'...tests/test_lottery_unit.py . [100%]====================================== 1 passed, 5 deselected in 2.34s ======================================= Terminating local RPC client...
```

Test Number 5

```
brownie test -k test_can_pick_winner_correctly Result:   Brownie v1.18.1 - Python development framework for EthereumPython-dotenv could not parse statement starting at line 1 Python-dotenv could not parse statement starting at line 1 ========================================= test session starts ========================================= platform linux -- Python 3.8.10, pytest-6.2.5, py-1.11.0, pluggy-1.0.0 rootdir: /home/mohamad/smartcontract_lottery plugins: eth-brownie-1.18.1, web3-5.27.0, xdist-1.34.0, forked-1.4.0, hypothesis-6.27.3 collected 6 items / 5 deselected / 1 selectedLaunching 'ganache-cli --chain.vmErrorsOnRPCResponse true --wallet.totalAccounts 10 --hardfork istanbul --miner.blockGasLimit 12000000 --wallet.mnemonic brownie --server.port 8545'...tests/test_lottery_unit.py . [100%]====================================== 1 passed, 5 deselected in 2.53s ======================================= Terminating local RPC client…
```

Deploying Our Lottery Contract

All the tests have been successfully run. Now it is time to finally deploy our contract: `brownie run scripts/deploy_lottery.py` Result:

```
Brownie v1.18.1 - Python development framework for EthereumPython-dotenv could not parse statement starting at line 1 Python-dotenv could not parse statement starting at line 1 SmartcontractLotteryProject is the active project.Launching 'ganache-cli --chain.vmErrorsOnRPCResponse true --wallet.totalAccounts 10 --hardfork istanbul --miner.blockGasLimit 12000000 --wallet.mnemonic brownie --server.port 8545'...Running 'scripts/deploy_lottery.py::main'... Transaction sent: 0x8dece070c593b0dcea9caa4740f89c80ff2fe29744f8e6b6ed26bdf879c7ec9f Gas price: 0.0 gwei Gas limit: 12000000 Nonce: 0
```

```
MockV3Aggregator.constructor confirmed Block: 1 Gas used: 430659
(3.59%) MockV3Aggregator deployed at:
0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87Transaction sent:
0xab119cd52792e53c56eee35f433c3f4b8ae409e8a5d397ac2dd16aa2d9d88e27 Gas
price: 0.0 gwei Gas limit: 12000000 Nonce: 1 LinkToken.constructor
confirmed Block: 2 Gas used: 669136 (5.58%) LinkToken deployed at:
0x602C71e4DAC47a042Ee7f46E0aee17F94A3bA0B6Transaction sent:
0xb549afbc42364cb3e72877bf6d3634d655fbe0b608a09e921a3f881983dff488 Gas
price: 0.0 gwei Gas limit: 12000000 Nonce: 2
VRFCoordinatorMock.constructor confirmed Block: 3 Gas used: 276395
(2.30%) VRFCoordinatorMock deployed at:
0xE7eD6747FaC5360f88a2EFC03E00d25789F69291Deployed! Transaction sent:
0xbb65e39ea0771823b51b07af3e19122d0a9be34b51e768a536a215a9571fdcc7 Gas
price: 0.0 gwei Gas limit: 12000000 Nonce: 3 Lottery.constructor
confirmed Block: 4 Gas used: 897156 (7.48%) Lottery deployed at:
0x6951b5Bd815043E3F842c1b026b0Fa888Cc2DD85Deployed lottery!
Transaction sent:
0xdada264c84b388a9674518ac1fff2a6e7c2851f9adf5dc7ecd871bed9d0c8629 Gas
price: 0.0 gwei Gas limit: 12000000 Nonce: 4 Lottery.startLottery
confirmed Block: 5 Gas used: 28902 (0.24%)Lottery.startLottery
confirmed Block: 5 Gas used: 28902 (0.24%)The lottery is started!
Transaction sent:
0x9556bd5fecc58be55b36e84b6612ae264d00fe685e4862f0458846a30e305e2e Gas
price: 0.0 gwei Gas limit: 12000000 Nonce: 5 Lottery.enter confirmed
Block: 6 Gas used: 70995 (0.59%)Lottery.enter confirmed Block: 6 Gas
used: 70995 (0.59%)You entered the lottery! Transaction sent:
0x94f77b768b68faefacdc57046a42caadf5d954a0d0e089bb551f6594cb44037e Gas
price: 0.0 gwei Gas limit: 12000000 Nonce: 6 LinkToken.transfer
confirmed Block: 7 Gas used: 51398 (0.43%)LinkToken.transfer confirmed
Block: 7 Gas used: 51398 (0.43%)Fund contract! LinkToken.transfer
confirmed Block: 7 Gas used: 51398 (0.43%)Transaction sent:
0x6591ec2cc1bfcddb9aea2f80e2723a773506e71bc8b649d58cf02c9708da1847 Gas
price: 0.0 gwei Gas limit: 12000000 Nonce: 7 Lottery.endLottery
confirmed Block: 8 Gas used: 79626 (0.66%)Lottery.endLottery confirmed
Block: 8 Gas used: 79626
(0.66%)0x0000000000000000000000000000000000000000 is the new winner!
Terminating local RPC client...
```

Getting Some Test Link

By the way, before the deployment of this contract, make sure you have some Link Rinkeby tokens in your Metamask wallet. If you don't have any, you can first add a Link Rinkeby token from Rinkeby Etherscan (By entering the decimals and contract address and so on). Then use Rinkeby Faucet which gives you 10 Link tokens. By using this link, you can get a Rinkeby Link token to be able to test your project:

https://docs.chain.link/docs/link-token-contracts/#rinkeby

Deploying Lottery Project on the Rinkeby Testnet

After running all the tests successfully and deploying it on Ganache-CLI, it is finally time to deploy our Lottery smart contract on Rinkeby chain, and inter-act with it using Rinkeby etherscan. To do so, in the terminal, we type:

```
brownie run scripts/deploy_lottery.py --network rinkeby Result:   Brownie v1.18.1 - Python development framework for EthereumSmartcontractLotteryProject is the active project.Running 'scripts/deploy_lottery.py::main'... Transaction sent: 0xd0172e3f2d1caf7a56b7a1f88b5b95d152843f38f8e2f0bba68d2a5c3b534df7 Gas price: 1.000006975 gwei Gas limit: 995904 Nonce: 54 Lottery.constructor confirmed Block: 10461557 Gas used: 905368 (90.91%) Lottery deployed at: 0x2a5D4140962F09f4a5B4F5f7C563af4Eb45B79b4Waiting for https://api-rinkeby.etherscan.io/api to process contract... Verification submitted successfully. Waiting for result... Verification pending... Verification complete. Result: Pass - Verified Deployed lottery! Transaction sent: 0x7b803fc2bedd3a1be5fa689e527e317baa4177c35ba589c1ea69ce3d6834298c Gas price: 1.000008155 gwei Gas limit: 31572 Nonce: 55 Lottery.startLottery confirmed Block: 10461565 Gas used: 28702 (90.91%)Lottery.startLottery confirmed Block: 10461565 Gas used: 28702 (90.91%)The lottery is started! Transaction sent: 0x7f2f38f4cfc0d34753c2635e8477b4c32118c3211ea3a3e6613a034a83242f99 Gas price: 1.000008155 gwei Gas limit: 97460 Nonce: 56 Lottery.enter confirmed Block: 10461566 Gas used: 88600 (90.91%)Lottery.enter confirmed Block: 10461566 Gas used: 88600 (90.91%)You entered the lottery! Transaction sent: 0xe1c566be0f8aff444154cf2d655cfc75082d83a7d13b9fb04f4315db3cece877 Gas price: 1.000008524 gwei Gas limit: 56992 Nonce: 57 LinkToken.transfer confirmed Block: 10461567 Gas used: 51811 (90.91%)LinkToken.transfer confirmed Block: 10461567 Gas used: 51811 (90.91%)Fund contract! LinkToken.transfer confirmed Block: 10461567 Gas used: 51811 (90.91%)Transaction sent: 0x585048e0b048833c35f3bd8257425e1e6f1f781cbd507df97abd756e7343c606 Gas price: 1.000008079 gwei Gas limit: 174062 Nonce: 58 Lottery.endLottery confirmed Block: 10461568 Gas used: 153439 (88.15%)Lottery.endLottery confirmed Block: 10461568 Gas used: 153439 (88.15%)0x25E681EE76469E4cF846567b772e94e082907117 is the new winner!
```

Rinkeby Etherscan

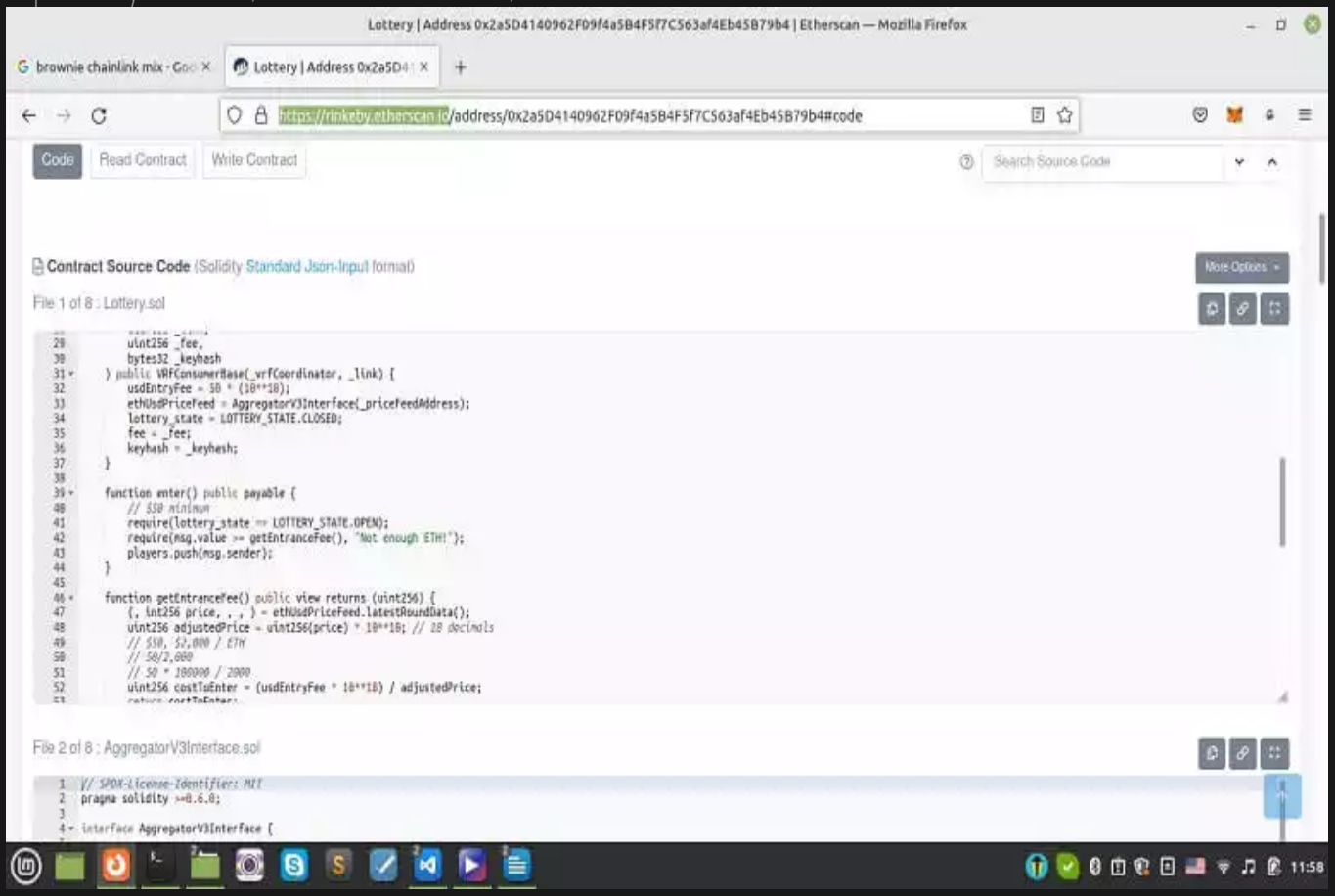
Now, we can copy the and paste 0x2a5D4140962F09f4a5B4F5f7C563af4Eb45B79b4 address in Rinkeby etherscan to be able to interact with your smart contract:

https://rinkeby.etherscan.io

lottery project 1.jpg Lottery Project Compiling: A Complete Tutorial

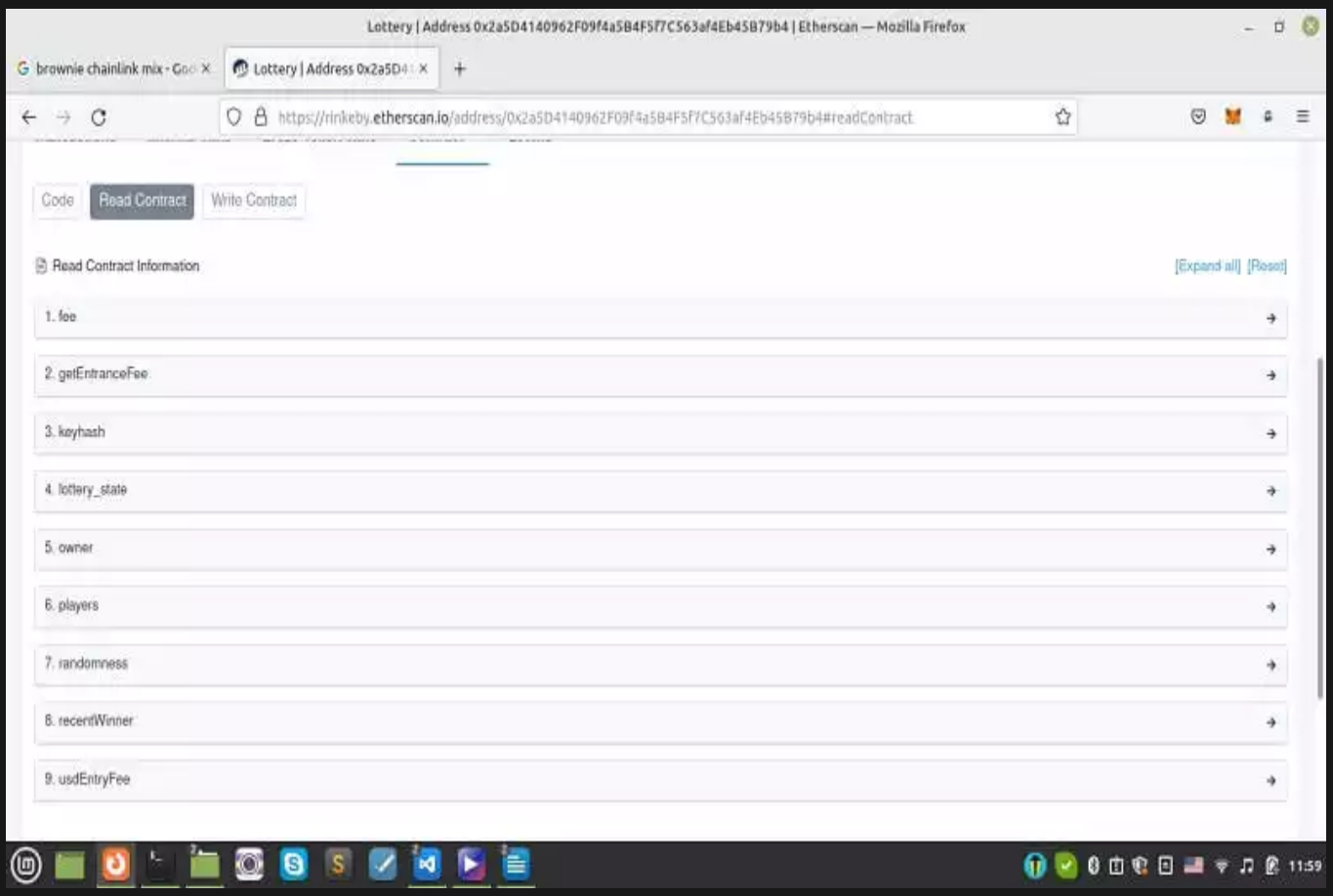
Once you enter Etherscan and paste your address, you will be able to see your address, ABI of the contract,

dependency contracts, contract creation code, and so on.

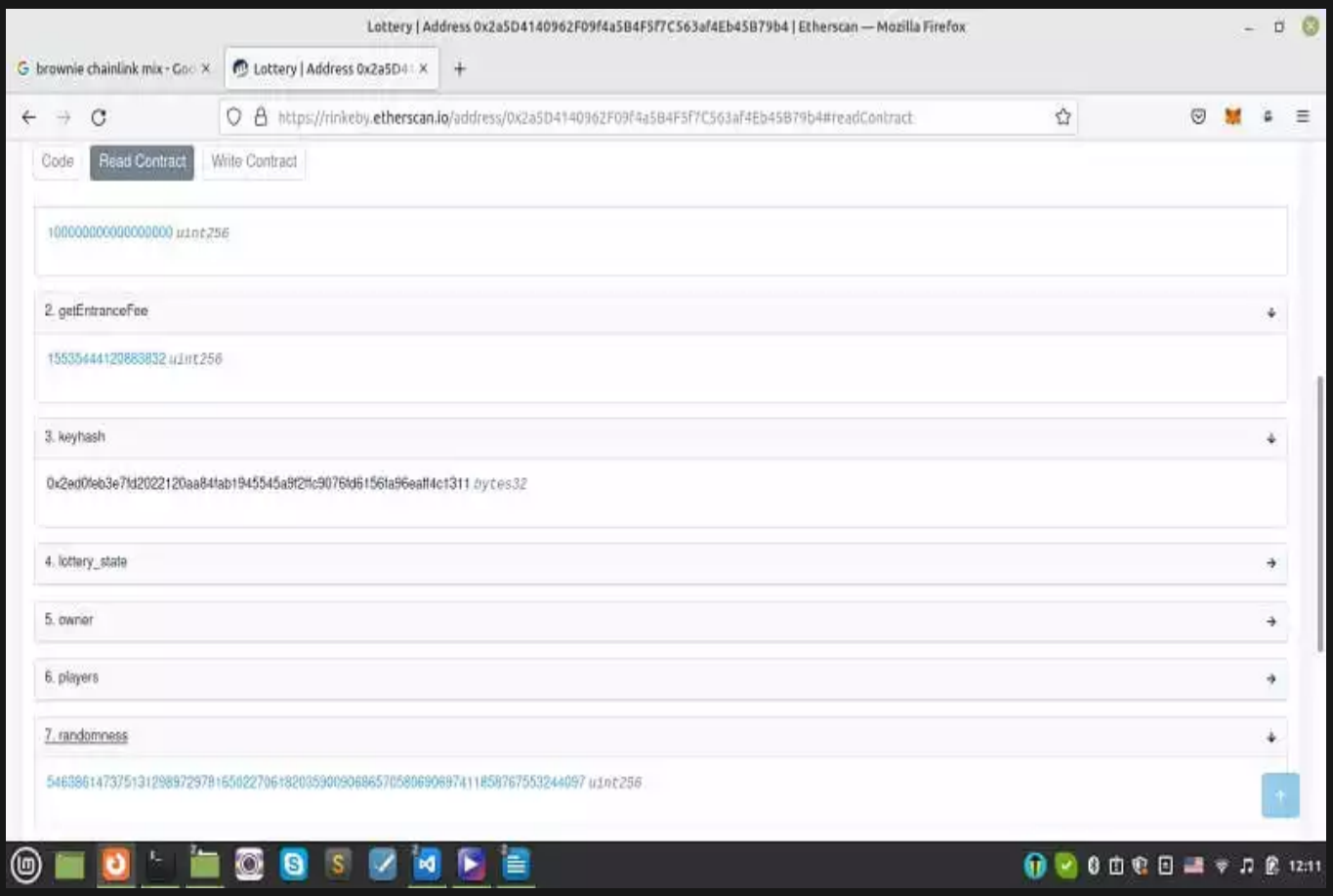


Interacting with the Deployed Contract

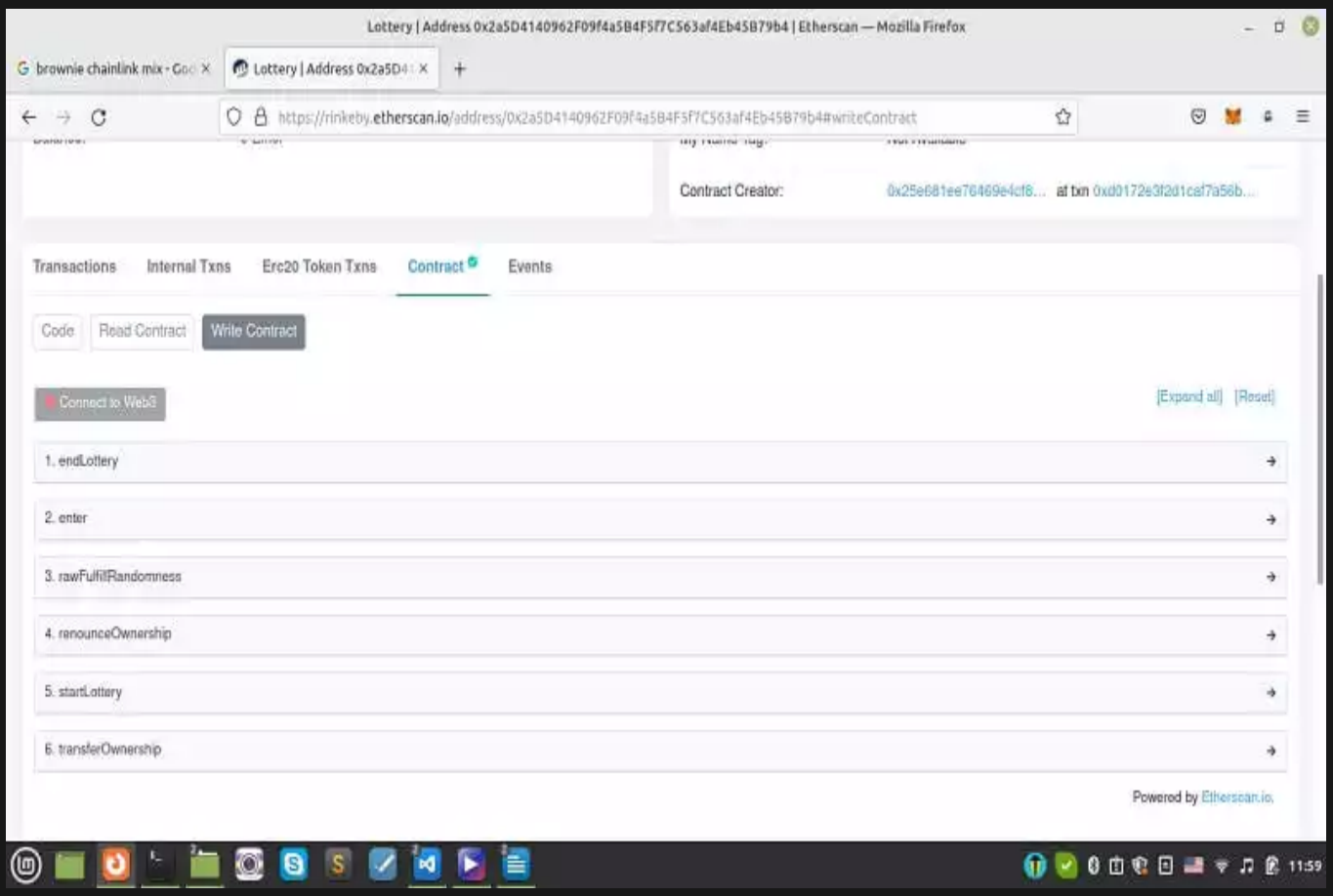
To interact with the deployed smart contract, head over to Read the contract and write the contract.



You can read some of the details of the public variables after deployment in the Read contract section:

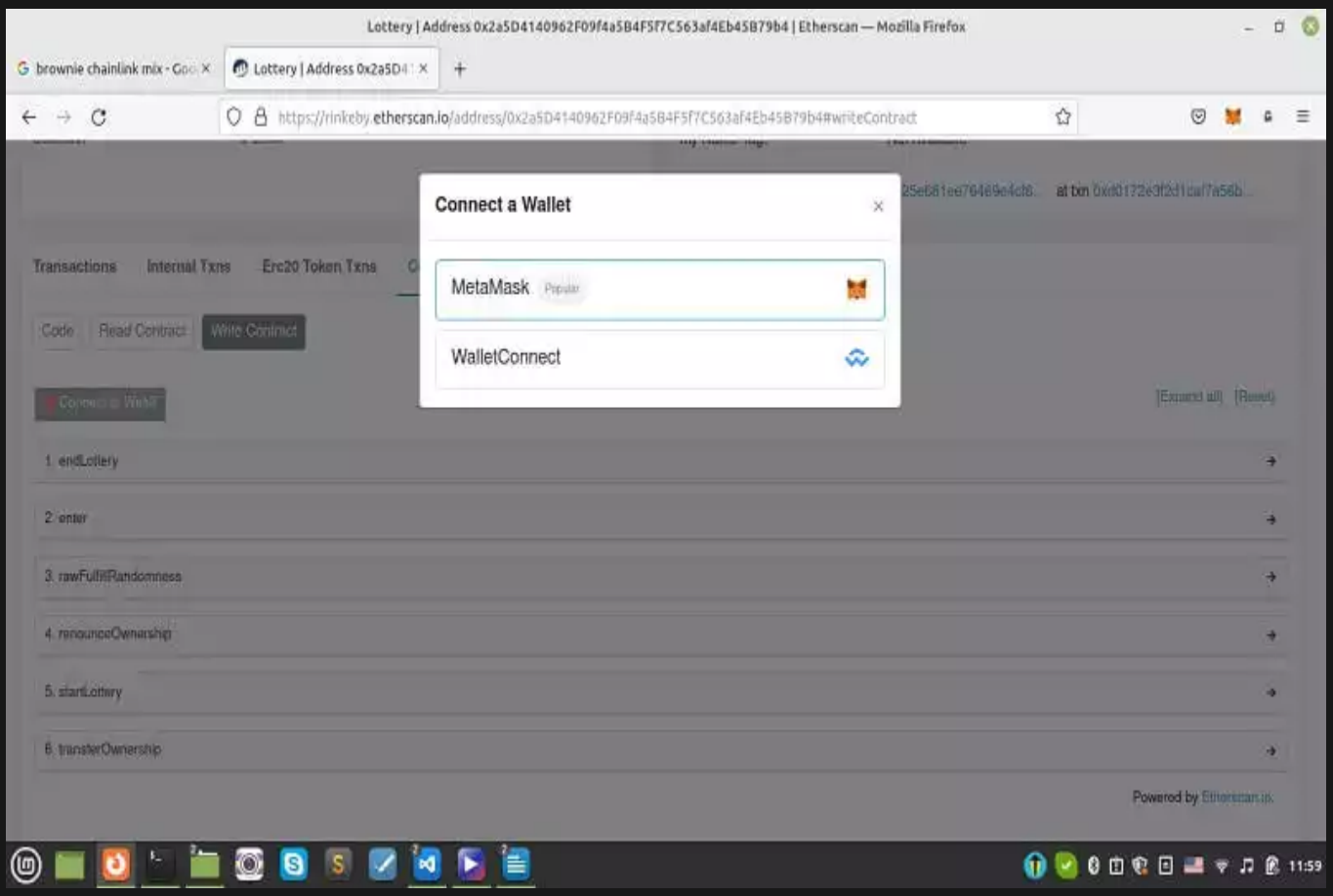


You can also run another lottery using Write Contract section:

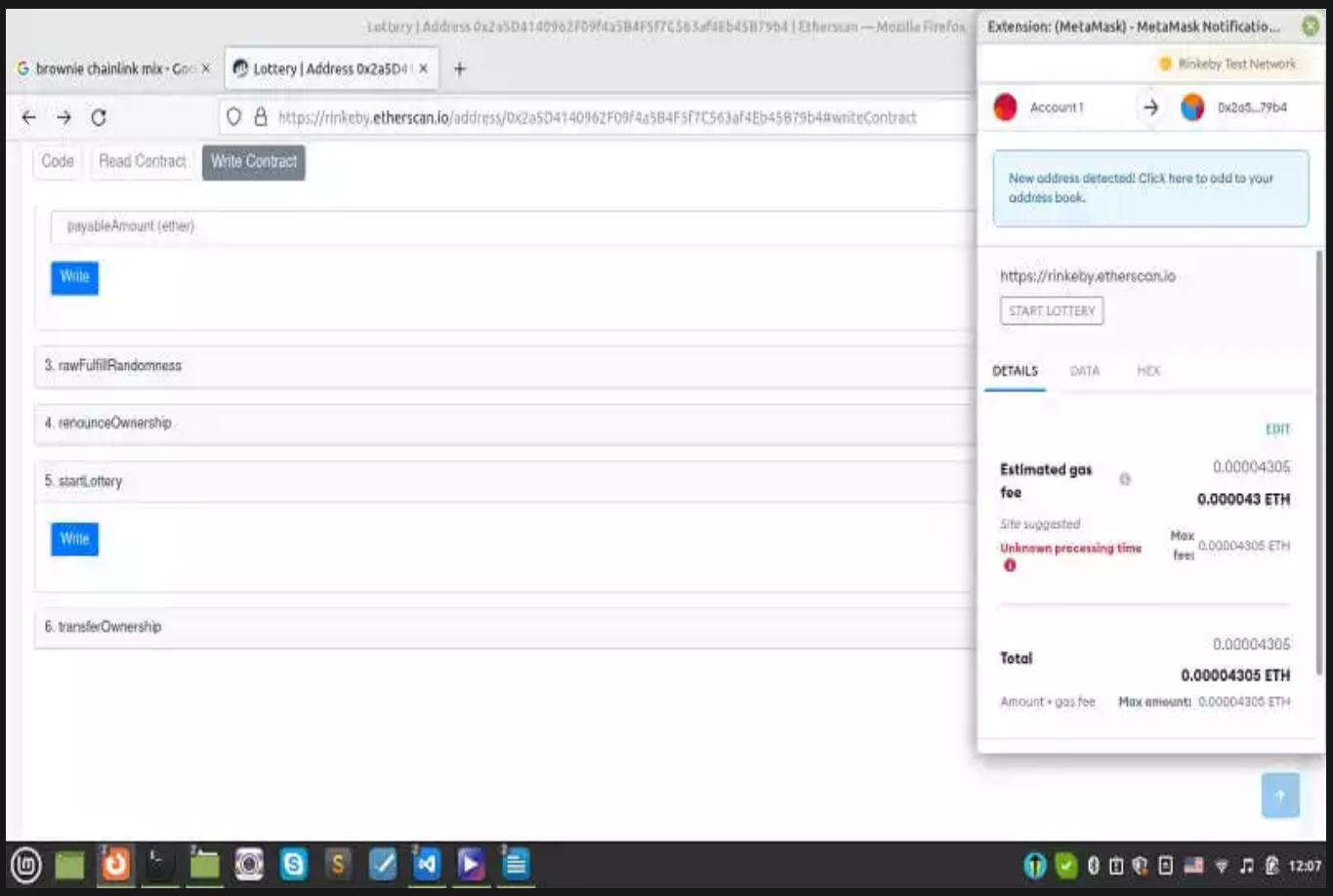


Connecting to Metamask Wallet

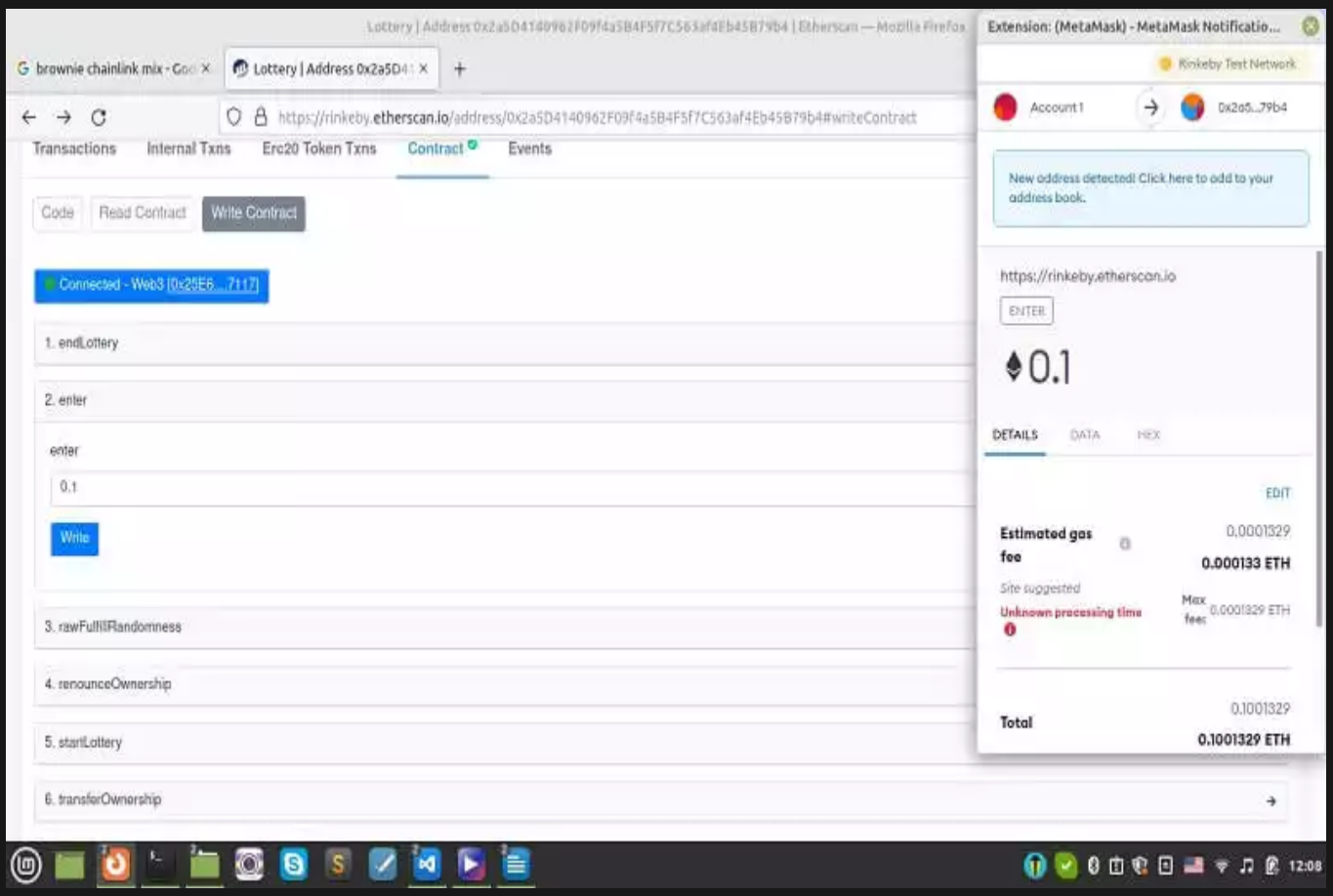
But, before writing anything in there, make sure you connect to your wallet by clicking on Connect to Web3 button:



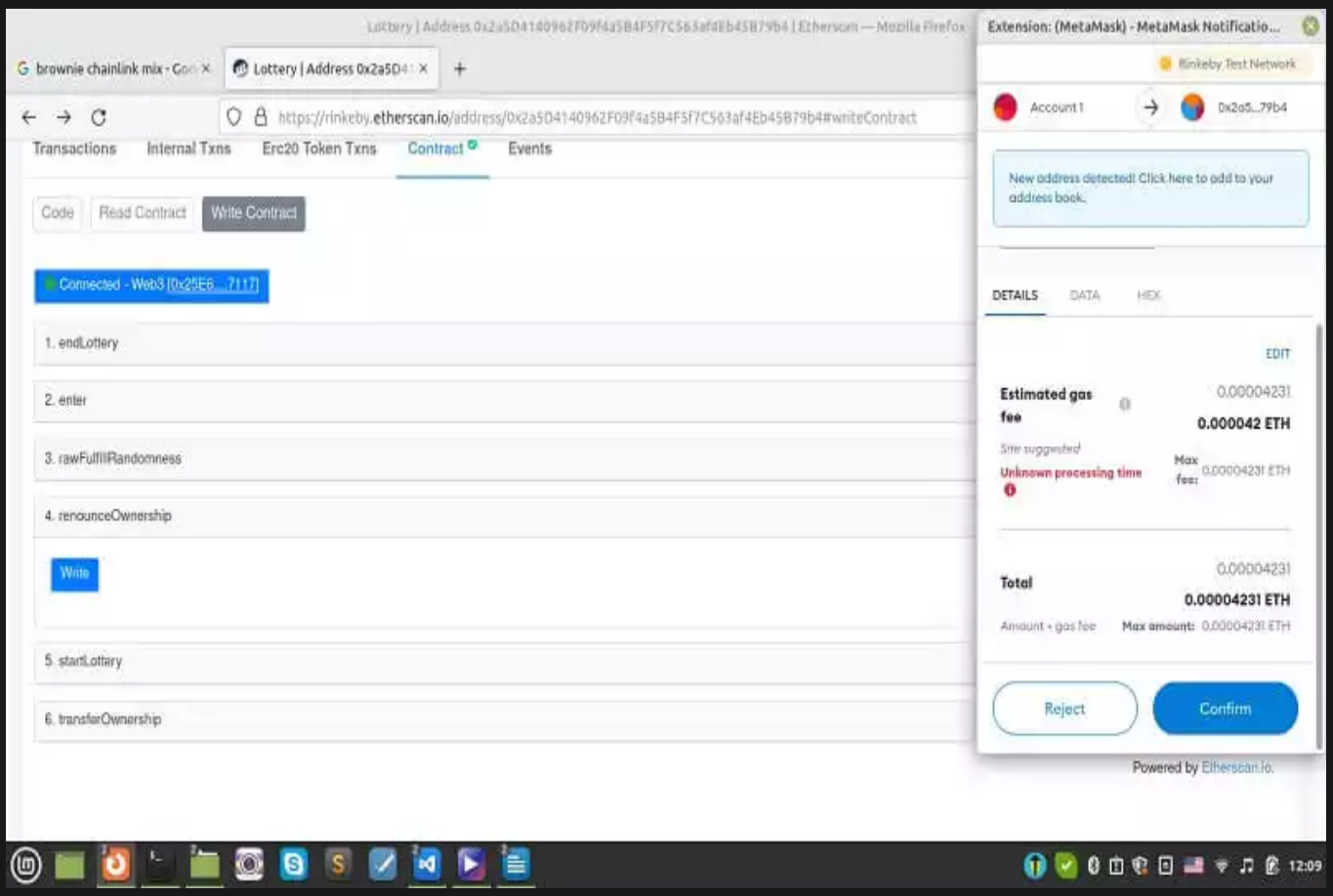
Now you can start the lottery:



Enter the lottery:



And execute other functions:



And that's it. You have written a lottery smart contract, deployed it on a test net, Rinkeby chain, and other networks, and learned how to interact with it using Rinkeby Etherscan!

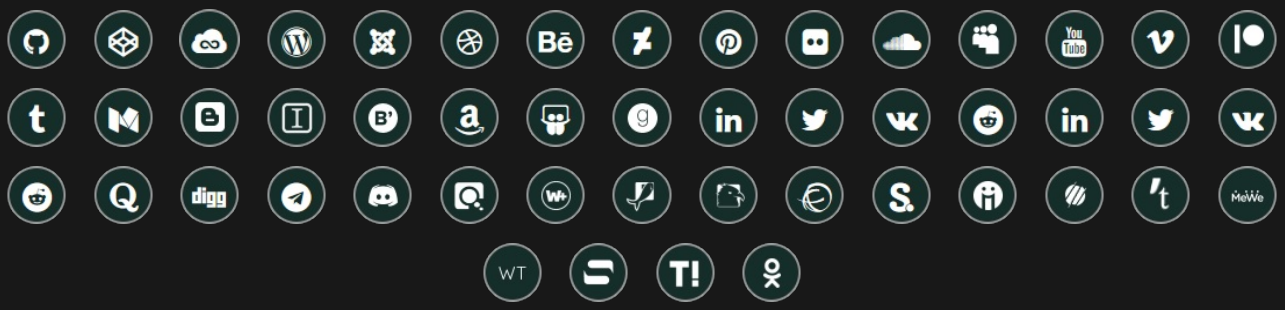
Lottery Project: Done!

In this tutorial, we have managed to deploy the contract on Rinkeby Testnet and interact with it from the address of the deployment on the Rinkeby Etherscan. To make this interaction possible from Etherscan, we connected the Metamask account of the admin of the contract to this website and applied all the stages of the lottery from the contract section.

Join Arashtad Community

Follow Arashtad on Social Media

We provide variety of content, products, services, tools, tutorials, etc. Each social profile according to its features and purpose can cover only one or few parts of our updates. We can not upload our videos on SoundCloud or provide our eBooks on Youtube. So, for not missing any high quality original content that we provide on various social networks, make sure you follow us on as many social networks as you're active in. You can find out Arashtad's profiles on different social media services.



Get Even Closer!

Did you know that only one universal Arashtad account makes you able to log into all Arashtad network at once? Creating an Arashtad account is free. Why not to try it? Also, we have regular updates on our newsletter and feed entries. Use all these benefitial free features to get more involved with the community and enjoy the many products, services, tools, tutorials, etc. that we provide frequently.

SIGN UP | NEWSLETTER | RSS FEED