Lido depositor bot

Introduction

Depositor bot is a part ofDeposit Security Module .

The Depositor Bot obtains signed deposit messages from Council Daemons. Once a sufficient number of messages is collected to constitute a quorum, the bot proceeds to initiate a deposit into the designated staking module. This deposit is executed using thedepositBufferedEther function within theDepositSecurityModule smart contract.

Requirements

Hardware

- 1-core CPU
- 2GB RAM

Nodes

- Ethereum EL RPC service
- Onchain databus transport RPC service (Gnosis at the moment)

How to use

Depositor bot performs series of checks before accepting the deposit. One of the most important optimisations it is doing is optimising gas spending. An example of this is fetchingGAS_FEE_PERCENTILE_DAYS_HISTORY_1 days of gas history and checkingGAS_FEE_PERCENTILE_1 bot will send transactions only if current gas price is less or equals to the percentile. AlsoGAS_PRIORITY_FEE_PERCENTILE ,MIN_PRIORITY_FEE ,MAX_PRIORITY_FEE variables are used to calculatemaxFeePerGas andmaxPriorityFeePerGas transaction parameters. The formula is:

priority = min(max(GAS_PRIORITY_FEE_PERCENTILE reward percentile of fee history for the last block, MIN_PRIORITY_FEE,), MAX_PRIORITY_FEE,)

maxFeePerGas = baseFeePerGas * 2 + priority maxPriorityFeePerGas = priority

Envs

Required variables are(mainnet):

Variable Default Description WEB3_RPC_ENDPOINTS - List of EL rpc endpoints that will be used to send requests comma separated (,) WALLET_PRIVATE_KEY - Account private key CREATE_TRANSACTIONS false If true then tx will be send to blockchain LIDO_LOCATOR 0xC1d0b3DE6792Bf6b4b37EccdcC24e45978Cfd2Eb Lido Locator address. Mainnet by default. Other networks can be foundhere DEPOSIT_CONTRACT 0x00000000219ab540356cBB839Cbe05303d7705Fa Ethereum deposit contract address DEPOSIT_MODULES_WHITELIST 1 List of staking module's ids in which the depositor bot will make deposits

---

MESSAGE_TRANSPORTS

Transports used in bot. Set: onchain_transport ONCHAIN_TRANSPORT_RPC_ENDPOINTS - List of databus(Gnosis) rpc endpoints that will be used for reading data bus contract, comma separated (,). ONCHAIN_TRANSPORT_ADDRESS - Data bus contract address. MIN_PRIORITY_FEE 50 mwei Min priority fee that will be used in tx MAX_PRIORITY_FEE 10 gwei Max priority fee that will be used in tx MAX_GAS_FEE 100 gwei Bot will wait for a lower price. Treshold for gas_fee CONTRACT_GAS_LIMIT 15000000 Default transaction gas limit GAS_FEE_PERCENTILE_1 20 Percentile for first recommended fee calculation GAS_FEE_PERCENTILE_DAYS_HISTORY_1 1 Percentile for first recommended calculates from N days of the fee history GAS_PRIORITY_FEE_PERCENTILE 25 Priority transaction will be N percentile from priority fees in last block (min MIN_PRIORITY_FEE - max MAX_PRIORITY_FEE) Optional variables can be foundhere .

Running

Source Code

1. Clone repository and install requirements:git clone git@github.com:lidofinance/depositor-bot.git
2. cd depositor-bot
3. Install requirementspoetry install
4. Run depositor botpoetry run python src/depositor.py
5. Verify in logs that depositor bot is performing validations, you should see logs of a kind:{"name": "bots.depositor", "levelname": "INFO", "funcName": "execute", "lineno": 121, "module": "depositor", "pathname": "/app/src/bots/depositor.py", "timestamp": 1729511569, "msg": "Do deposit to module with id: 1."}
6. {"name": "bots.depositor", "levelname": "INFO", "funcName": "_deposit_to_module", "lineno": 210, "module": "depositor", "pathname": "/app/src/bots/depositor.py", "timestamp": 1729511569, "msg": "Checks failed. Skip deposit."}
7. {"name": "bots.depositor", "levelname": "INFO", "funcName": "_deposit_to_module", "lineno": 194, "module": "depositor", "pathname": "/app/src/bots/depositor.py", "timestamp": 1729511569, "msg": "Calculations deposit recommendations.", "value": false, "is_mellow": false}

If you are facing problems, check what environment variables depositor bot is using, find a log line"msg": "Bot env variables"

Docker

Docker image can be foundhere .

Monitoring

Prometheus metrics will be available on endpointhttp://localhost:{PROMETHEUS_PORT}/metrics . Alertssource code for AlertManager.

Edit this page Previous Reward distribution bot Next Multisig deployment