How to run a validator

Some Arbitrum nodes will choose to act as validators. This means that they watch the progress of the rollup protocol and perhaps also participate in that protocol to advance the state of the chain securely. Not all nodes will choose to do this. Because the rollup protocol doesn't decide what the chain will do but merely confirms the correct behavior that is fully determined by the inbox messages, a node can ignore the rollup protocol and simply compute for itself the correct behavior. Here we describe different strategies that validators follow and provide instructions on how to run them.

Validation strategies

- Currently, the ability to post assertions on-chain for mainnet Arbitrum chains is allowlisted.
- Here's a full list of validation strategies:* Defensive
-
    - (allowlist required)* Post stake and create challenge if local state disagrees with on-chain assertion (wallet required, will only post stake on-chain if bad assertion found)
-
    - StakeLatest
-
    - (allowlist required)* Stay staked on latest assertion and challenge any bad assertions found (wallet required, always staked, uses some gas every time new assertion created)
-
    - ResolveNodes
-
    - (allowlist required)* Stay staked on latest assertion, resolve any unconfirmed assertions and challenge any bad assertions found (wallet required, always staked, uses some gas every time unconfirmed assertion resolved or new assertion created)
-
    - MakeNodes
-
    - (allowlist required)* Continuously create new assertions, challenging any bad assertions found (wallet required, always staked, most expensive node to run)
-
    -
        - Note that if there is more than oneMakeNodes
-
    -
        - validator running, they might all try to create a new assertion at same time. In that case, only one will be successful, and the others will have still spent gas on reverted calls that didn't do anything.
- There's one more validation strategy that is not allowlisted and is available for all types of node:Watchtower
- . A node inWatchtower
- mode will immediately log an error if an on-chain assertion deviates from the locally computed chain state. It doesn't require a wallet, as it never takes any action on-chain. This strategy is enabled by default in all nodes (full and archive).

Running a Watchtower validator

- By default, all nodes (full and archive) will run inWatchtower
- mode.
- If a deviation is detected, a node running in Watchtower mode will log an error containing the stringfound incorrect assertion in watchtower mode
- To verify that the Watchtower mode is enabled, this line should appear in the logs:INFO
- [
- 09-28
- |
- 18
- :43:49.367
-]
- running as validator
- txSender
- =
- nil
- actingAsWallet
- =
- nil
- whitelisted
- =
- false
- strategy

- =
- Watchtower
-
    - strategy
-
    - should beWatchtower
-
    - The log linevalidation succeeded
-
    - shows that the L2 block validator is working
-
    - The log linefound correct assertion
-
    - shows that the L1 validator is working
- Watchtower mode adds a small amount of execution and memory overhead. You can deactivate this mode by using the parameter--node.staker.enable=false
- .

Creating a wallet for an allowlisted validator

- Watchtower validators never need a wallet, because they never post on-chain
- Defensive validators need a wallet configured, but the wallet does not need to be funded until it logs that an assertion has been found
- All other validators require a funded wallet to immediately post stake, as well as additional funds that will be spent at regular intervals
- Here is an example of how to tell Nitro to create validator wallet for Arbitrum One and exit:docker
- run --rm -it -v /some/local/dir/arbitrum:/home/user/.arbitrum offchainlabs/nitro-node:v2.3.3-6a1c1a7 --parent-chain.connection.url
- =
- https://l1-mainnet-node:8545 --chain.id
- =
- 42161
- --node.staker.enable --parent-chain.wallet.only-create-key --parent-chain.wallet.password
- =
- "SOME SECURE PASSWORD"
- Wallet file will be created under the mounted directory inside thearb1/wallet/
- directory for Arb1, ornova/wallet/
- directory for Nova. Be sure to backup the wallet, it will be the only way to withdraw stake when desired

Running an allowlisted defensive validator

- A defensive validator requires that a wallet has already been created using the above steps
- Defensive validator wallets do not need to be funded initially
- If a defensive validator detects a deviation, it will logbringing defensive validator online because of incorrect assertion
- , and wait for funds to be added to wallet so stake can be posted and a dispute created
- Here is an example of how to run an allowlisted defensive validator for Arbitrum One:docker
- run --rm -it -v /some/local/dir/arbitrum:/home/user/.arbitrum offchainlabs/nitro-node:v2.3.3-6a1c1a7 --parent-chain.connection.url
- =
- https://l1-mainnet-node:8545 --chain.id
- =
- 42161
- --node.staker.enable --node.staker.strategy
- =
- Defensive --parent-chain.wallet.password
- =
- "SOME SECURE PASSWORD"
- To verify validator is working, this log line shows the wallet is setup correctly:INFO
- [
- 09-28
- |
- 18
- :43:49.367
-]
- running as validator
- txSender
- =
- 0x

- ..
- .
- actingAsWallet
- =
- 0x
- ..
- .
- whitelisted
- =
- true
- strategy
- =
- Defensive
-
  - whitelisted
-
  - should betrue
-
  - after your wallet has been added to the allowlist
-
  - strategy
-
  - should beDefensive
-
  - txSender
-
  - andactingAsWallet
-
  - should both be present and notnil
-
  - The log linevalidation succeeded
-
  - shows that the L2 block validator is working
-
  - The log linefound correct assertion
-
  - shows that the L1 validator is working[Edit this page](#) Last updatedonApr 24, 2024[Previous How to run an archive node](#) [Next How to run a Classic node](#)