

# How to run a full node (Nitro)

info There is no protocol level incentive to run an Arbitrum full node. If you're interested in accessing an Arbitrum chain, but you don't want to set up your own node, see our [Node Providers](#) to get RPC access to fully-managed nodes hosted by a third party provider.

## Minimum hardware configuration

Minimum hardware configuration required to setup a Nitro full node (not archival):

- RAM
  - : 8-16 GB
- CPU
  - : 2-4 core CPU (For AWS: t3 xLarge)
- Storage (last updated on January 2024)
  - : \* Arbitrum One: 480GB SSD for a pruned node, growing at ~40GB per month
  - - Arbitrum Nova: 112 GB SSD for a pruned node, growing at ~20GB per month

info The minimum storage requirements will change over time as the Nitro chain grows. It is recommended to use more than the minimum requirements to run a robust full node.

## Prerequisites

Only use released versions Even though there are alpha and beta versions of the [Arbitrum Nitro software](#), only release versions should be used when running your node. Running alpha or beta versions is not supported, and might lead to unexpected behaviors. \* Latest Docker Image: `offchainlabs/nitro-node:v2.3.2-064fa11` \* Database snapshot (required for Arbitrum One, optional for other chains) \* Use the parameter `--init.url` \* \* on first startup to initialize the Nitro database (you can find a list of snapshots [here](#) \* \* ). Example: `--init.url="https://snapshot.arbitrum.foundation/arb1/nitro-pruned.tar" * * . * * When running more than one node, it's easier to manually download the image of the snapshot and host it locally for your nodes. You can then use --init.url="file:///path/to/snapshot/in/container/snapshot-file.tar" * * to use it. * * This parameter is required * * when initializing an Arbitrum One node because the chain has classic * * blocks. For the rest of chains, this parameter is optional. * * This parameter is ignored if the database already exists.`

## Required parameters

- L1 RPC URL \* Use the parameter `--parent-chain.connection.url=`
  - - for execution layer.
  - - If the chain is running [ArbOS 20](#)
  - - , additionally use the parameter `--parent-chain.blob-client.beacon-url=`
  - - for consensus layer. You can find a list of beacon chain RPC providers [here](#)
  - - .
  - - It must provide a standard layer 1 node RPC endpoint that you run yourself or from a node provider.
  - - Note: this parameter was called `--l1.url`
  - - in versions prior to v2.1.0
  - - Note: 8545 is usually the default port for the execution layer. For the Beacon endpoint port, you should connect to the correct port that is set on your parent chain's consensus client.
- L2 chain id or name \* Use the parameter `--chain.id=`
  - - to set the L2 chain from its chain id. See [RPC endpoints and providers](#)
  - - for a list of Arbitrum chains and their respective L2 chain ids.
  - - Alternatively, you can use the parameter `--chain.name=`
  - - to set the L2 chain from its name (options are: `arb1`
  - - , `nova`
  - - and `sepolia-rollup`
  - - )
  - - Note: this parameter was called `--l2.chain-id` and only accepted chain ids in versions prior to v2.1.0

## Important ports

- RPC:8547
- Sequencer Feed:9642
- WebSocket:8548
- - WS port8548
- - needs extra args to be opened. Please use these flags:\* --ws.port=8548
- - - --ws.addr=0.0.0.0
- - - --ws.origins=\*

## Putting it all together

- When running docker image, an external volume should be mounted to persist the database across restarts. The mount point inside the docker image should be /home/user/.arbitrum
- Here is an example of how to run nitro-node:
  - - Note that is important that /some/local/dir/arbitrum
  - - already exists, otherwise the directory might be created with root
  - - as owner, and the docker container won't be able to write to it
- docker
- run --rm -it -v /some/local/dir/arbitrum:/home/user/.arbitrum -p
- 0.0
- .0.0:8547:8547 -p
- 0.0
- .0.0:8548:8548 offchainlabs/nitro-node:v2.3.2-064fa11 --parent-chain.connection.url https://l1-node:8545 --chain.id
- =
- <
- L2ChainId
- 
- --http.api
- =
- net,web3,eth --http.corsdomain
- =
- - --http.addr
- =
- 0.0
- .0.0 --http.vhosts
- =
- \*
- - Note that if you are running L1 node on localhost, you may need to add --network host
- - right after docker run
- - to use docker host-based networking
- - When shutting down docker image, it is important to allow for a graceful shutdown so that the current state can be saved to disk. Here is an example of how to do a graceful shutdown of all docker images currently running
- docker
- stop --time
- =
- 300
- (
- docker
- ps
- -aq
- )

## Note on permissions

- The Docker image is configured to run as non-root UID 1000. This means if you are running in Linux or OSX and you are getting permission errors when trying to run the docker image, run this command to allow all users to update the persistent folders mkdir
- /data/arbitrum
- chmod

- -fR
- 777
- /data/arbitrum

## Watchtower mode

- By default, the full node will run in Watchtower mode. This means that the node watches the on-chain assertions and if it disagrees with them, it will log an error containing the stringfound incorrect assertion in watchtower mode
- .
- Watchtower mode adds a small amount of execution and memory overhead. You can deactivate this mode by using the parameter--node.staker.enable=false
- .

## Pruning

- Pruning a full node refers to the process of removing older, unnecessary data from the local copy of the blockchain that the node maintains, in order to save disk space and slightly improve the efficiency of the node. Pruning will remove all states from blocks older than the latest 128.
- You can activate pruning by using the parameter--init.prune
- and using "full" or "validator" as the value (depending on the type of node you are running). Keep in mind that this process will happen upon starting the node and it will not serve RPC requests while pruning.

## Optional parameters

We show here a list of the parameters that are most commonly used when running a node. You can also use the flag--help for a full comprehensive list of the available parameters.

Flag Description --execution.rpc.classic-redirect= Redirects archive requests for pre-nitro blocks to this RPC of an Arbitrum Classic node with archive database. Only for Arbitrum One. --http.api Offered APIs over the HTTP-RPC interface. Default:net,web3,eth,arb . Adddebug for tracing. --http.corsdomain Accepts cross origin requests from these comma-separated domains (browser enforced). --http.vhosts Accepts requests from these comma-separated virtual hostnames (server enforced). Default:localhost . Accepts\* . --http.addr Address to bind RPC to. May require0.0.0.0 for Docker networking. --execution.caching.archive Retains past block state. For archive nodes. --node.feed.input.url= Default:wss://arbitrum.io/feed . ⚠ One feed relay per datacenter is advised. See[feed relay guide](#) . --execution.forwarding-target= Defaults to the L2 Sequencer RPC based on provided L1 and L2 chain IDs. --execution.rpc.evm-timeout Default:5s . Timeout foreth\_call . (0 == no timeout). --execution.rpc.gas-cap Default:50000000 . Gas cap foreth\_call /estimateGas . (0 = no cap). --execution.rpc.tx-fee-cap Default:1 . Transaction fee cap (in ether) for RPC APIs. (0 = no cap). --ipc.path Filename for IPC socket/pipeline within datadir. Not supported on macOS. Note the path is within the Docker container. --init.prune Prunes before starting the node. Can be "full" or "validator". --init.url="" (Non-Orbit Nitro nodes only) URL to download the genesis database from. Required only for the first startup of an Arbitrum One node. Reference to[snapshots](#) and[archive node guide](#) . --init.download-path="/path/to/dir" (Non-Orbit Nitro nodes only) Temporarily saves the downloaded database snapshot. Defaults to/tmp/ . Used with--init.url . --node.batch-poster.post-4844-blobs Boolean. Default:false . Used to enable or disable the posting of transaction data using Blobs to L1 Ethereum. If using calldata is more expensive and if the parent chain supports EIP4844 blobs, the batch poster will use blobs when this flag is set tottrue . Can betrue orfalse . --node.batch-poster.ignore-blob-price Boolean. Default:false . If the parent chain supports EIP4844 blobs andignore-blob-price is set tottrue , the batch poster will use EIP4844 blobs even if using calldata is cheaper. Can betrue orfalse . [Edit this page](#) Last updatedonMar 22, 2024 [Previous Quickstart: Run a node](#)[Next How to run a local dev node](#)