How to run a full node for an Orbit chain

This how-to provides step-by-step instructions for running an Orbit node on your local machine.

Minimum hardware configuration

- RAM
- : 8-16 GB
- CPU
- : 2-4 core CPU (For AWS: t3 xLarge)
- Storage
- : It depends on the Orbit chain and its traffic during the time.

Prerequisites

- Latest Docker Image:offchainlabs/nitro-node:v2.3.2-064fa11

Required parameters

- Parent chain RPC URL* Use the parameter--parent-chain.connection.url=
-
    - It must provide a standard RPC endpoint for an EVM node, whether self-hosted or obtained from a node service provider* Would be set to appropriate Arbitrum One, Arbitrum Nova, or Arbitrum Sepolia RPC URL
-
    - Note:Arbitrum RPC endpoints
-
    - rate limit connections, So you will need to run a local node for the parent chain or rely on third-party RPC providers
- L3 Orbit chain info and chain name* Providing the flag --chain.info-json=is mandatory. The chain info json string contains vital information about the Orbit chain, which is necessary for the node to commence operations. An example of the chain info json string is available in the subsequent sections.
-
    - --chain.name=
-
    - must be provided, and it must match the chain name that is used in--chain.info-json
-
    - content.
-
    - For an Anytrust chain, it is essential to provide--node.data-availability.enable
-
    - .

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 a node for an Orbit chain:
-
    - Note that is important that/some/local/dir/arbitrum
-
    - already exists, otherwise the directory might be created withroot
-
    - 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
- =
- <
- Parent chain RPC URL
-
- --chain.id
- =
- <
- L3ChainId
-
- --chain.name
- =
- <
- My Arbitrum L3 Chain
-
- --http.api
- =
- net,web3,eth --http.corsdomain
- =
-
    - --http.addr
- =
- 0.0
- .0.0 --http.vhosts
- =
-
    - --chain.info-json
- =
- <
- Orbit Chain's chain info
-
-
    - Please note that when using the flag--chain.info-json=
-

- ○ , it is essential to replace with the specific chain info

- ○ json string of the Orbit chain for which you wish to run the node. For your convenience, an example of an chain info json string is provided below:
- --chain.info-json="[{\"chain-id\":94692861356,\"parent-chain-id\":421614,\"chain-name\":\"My Arbitrum L3 Chain\",\"chain-config\": {\"chainId\":94692861356,\"homesteadBlock\":0,\"daoForkBlock\":null,\"daoForkSupport\":true,\"eip150Block\":0,\"eip150Hash\":\"0x00000000000000000000000000000000000000000000000000000000 {\"period\":0,\"epoch\":0},\"arbitrum\": {\"EnableArbOS\":true,\"AllowDebugPrecompiles\":false,\"DataAvailabilityCommittee\":false,\"InitialArbOSVersion\":10,\"InitialChainOwner\":\"0xAde4000C87923244f0e95b41f0e45aa3C02f1Bb2\",\" {\"bridge\":\"0xde835286442c6446E36992c036EFe261AcD87F6d\",\"inbox\":\"0x0592d3861Ea929B5d108d915c36f64EE69418049\",\"sequencer-inbox\":\"0xf9d77199288f00440Ed0f494Adc0005f362c17b1\",\"rollup\":\"0xF5A42aDA664E7c2dFE9DDa4459B927261BF90E09\",\"validator-utils\":\"0xB11EB62DD2B352886A4530A9106fE427844D515f\",\"validator-wallet-creator\":\"0xEb9885B6c0e117D339F47585cC06a2765AaE2E0b\",\"deployed-at\":1764099}}]"

- ○ Note for Anytrust chains, it is essential to add --node.data-availability.enable

- ○ , and --node.data-availability.rest-aggregator.urls=

- ○ or --node.data-availability.rest-aggregator.online-url-list=

- ○ to the command or configuration.
- 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 Edit this page Last updated on Mar 7, 2024 Previous Quickstart: Launch an Orbit chain Next How to customize your Orbit chain's deployment configuration