

# How to Run a Full Node for an Orbit Chain

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

## Prerequisites

- Latest Docker Image: `offchainlabs/nitro-node:v2.3.4-b4cc111`

### Minimum Hardware Configuration

- RAM: 8-16 GB
- CPU: 2-4 core CPU (e.g., AWS t3.xLarge)
- Storage: Depends on the Orbit chain and its traffic over time

### Required Parameters

#### 1. Parent Chain Parameters

The `--parent-chain.connection.url` argument requires a standard RPC endpoint for an EVM node, whether self-hosted or obtained from a node service provider:

`--parent-chain.connection.url=<Parent chain RPC URL>`

#### 2. Child Chain Parameters

In the Arbitrum Orbit context, the child chain is an L2 or an L3 Orbit chain. The required parameters are `chain.info-json` and `chain.name`.

`chain.info-json` A JSON string that contains required information about the Orbit chain.

`--chain.info-json=<Orbit Chain's chain info>`

`chain.name` A mandatory flag that needs to match the chain name used in `--chain.info-json`:

`--chain.name=<Orbit Chain's name>`

3. AnyTrust Chains For AnyTrust chains, add the following flags to the command or configuration:

`--node.data-availability.enable`

`--node.data-availability.rest-aggregator.urls=<A list of DAS REST endpoints>`

Or

`--node.data-availability.rest-aggregator.online-url-list=<A URL that returns a list of the DAS REST endpoints>`

#### 4. Important Ports

Protocol	Port
RPC/http	8547
RPC/websocket	8548
Sequencer Feed	9642

For the RPC/websocket protocol, use the following flags:

```
--ws.port=8548  
--ws.addr=0.0.0.0  
--ws.origins=\*
```

#### 5. Putting it all together

When running a 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`.

Example:

```
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:v2.3.4-b4cc111 --parent-chain.connection.url=<Parent  
chain RPC URL> --chain.id=<OrbitChainId> --chain.name=<My Arbitrum Orbit Chain>  
--http.api=net,web3,eth --http.corsdomain=* --http.addr=0.0.0.0 --http.vhosts=*  
--chain.info-json=<Orbit Chain's chain info>
```

Ensure that `/some/local/dir/arbitrum` already exists; otherwise, the directory might be created with root as the owner, and the Docker container won't be able to write to it.

When using the flag `--chain.info-json=<Orbit Chain's chain info>`, replace `<Orbit Chain's chain info>` with the specific chain info JSON string of the Orbit chain for which you wish to run the node.

Example:

```
--chain.info-json="{\"chain-id\":94692861356,\"parent-chain-id\":421614,\"chain-name\":\"My  
Arbitrum L3  
Chain\", \"chain-config\":{ \"chainId\":94692861356, \"homesteadBlock\":0, \"daoForkBlock\":null, \"d  
aoForkSupport\":true, \"eip150Block\":0, \"eip150Hash\":\"0x00000000000000000000000000000000  
000000000000000000000000000000\", \"eip155Block\":0, \"eip158Block\":0, \"byzantiumBl  
ock\":0, \"constantinopleBlock\":0, \"petersburgBlock\":0, \"istanbulBlock\":0, \"muirGlacierBlock\":0,  
\"berlinBlock\":0, \"londonBlock\":0, \"clique\":{ \"period\":0, \"epoch\":0}, \"arbitrum\":{ \"EnableArbOS  
\":true, \"AllowDebugPrecompiles\":false, \"DataAvailabilityCommittee\":false, \"InitialArbOSVersion  
\":10, \"InitialChainOwner\":\"0xAde4000C87923244f0e95b41f0e45aa3C02f1Bb2\", \"GenesisBloc  
kNum\":0}}, \"rollup\":{ \"bridge\":\"0xde835286442c6446E36992c036EF6261AcD87F6d\", \"inbox\"
```

```
":\0x0592d3861Ea929B5d108d915c36f64EE69418049\","sequencer-inbox\":"0xf9d77199288f00440Ed0f494Adc0005f362c17b1\","rollup\":"0xF5A42aDA664E7c2dFE9DDa4459B927261BF90E09\","validator-utils\":"0xB11EB62DD2B352886A4530A9106fE427844D515f\","validator-wallet-creator\":"0xEb9885B6c0e117D339F47585cC06a2765AaE2E0b\","deployed-at":1764099}}]"
```

## Further Reading

For more detailed instructions and additional configuration options, please refer to the Arbitrum documentation [here](#).