Consensus node

This guide covers how to set up a consensus node on Celestia. Consensus nodes allow you to sync the entire blockchain history in the Celestia consensus layer.

Minimum hardware requirements

The following minimum hardware requirements are recommended for running a consensus node:

- Memory:16 GB RAM
- CPU:Quad-Core
- Disk:2 TB SSD Storage
- Bandwidth:1 Gbps for Download/1 Gbps for Upload

Set up a consensus node

The following tutorial is done on an Ubuntu Linux 20.04 (LTS) x64 instance machine.

Set up the dependencies

Follow the instructions oninstalling dependencies .

Install celestia-app

Follow the tutorial oninstallingcelestia-app .

Set up the P2P networks

To initialize the network, pick a "node-name" that describes your node. Keep in mind that this might change if a new testnet is deployed.

Mainnet Beta

Mocha

Arabica bash celestia-appd

init

"node-name"

--chain-id

celestia celestia-appd

init

"node-name"

--chain-id

celestia bash celestia-appd

init

"node-name"

--chain-id

mocha-4 celestia-appd

init

"node-name"

--chain-id

mocha-4 bash celestia-appd

init

"node-name"

--chain-id

arabica-11 celestia-appd

init

"node-name"

--chain-id

arabica-11 Download thegenesis.json file:

Mainnet Beta

Mocha

Arabica bash celestia-appd

download-genesis

celestia celestia-appd

download-genesis

celestia bash celestia-appd

download-genesis

mocha-4 celestia-appd

download-genesis

mocha-4 bash celestia-appd

download-genesis

arabica-11 celestia-appd

download-genesis

arabica-11 Set seeds in theHOME/.celestia-app/config/config.toml file:

Mainnet Beta

Mocha

Arabica bash SEEDS = (curl

-sL https://raw.githubusercontent.com/celestiaorg/networks/master/celestia/seeds.txt |

tr '\n' ',') echo SEEDS sed

-i.bak

-e

"s/^seeds =./seeds = \" SEEDS \" /" HOME /.celestia-app/config/config.toml SEEDS = (curl

-sL https://raw.githubusercontent.com/celestiaorg/networks/master/celestia/seeds.txt |

tr '\n' ',') echo SEEDS sed

-i.bak

-e

"s/^seeds =./seeds = \" SEEDS \" /" HOME /.celestia-app/config/config.toml bash SEEDS = (curl

-sL https://raw.githubusercontent.com/celestiaorg/networks/master/mocha-4/seeds.txt |

tr '\n' ',') echo SEEDS sed

-i.bak

-e

"s/^seeds =./seeds = \" SEEDS \" /" HOME /.celestia-app/config/config.toml SEEDS = (curl

-sL https://raw.githubusercontent.com/celestiaorg/networks/master/mocha-4/seeds.txt |

tr '\n' ',') echo SEEDS sed

-i.bak

-e

"s/^seeds =./seeds = \" SEEDS \" /" HOME /.celestia-app/config/config.toml bash

For Arabica, you can set seeds manually in the

HOME/.celestia-app/config/config.toml file:

Comma separated list of seed nodes to connect to

seeds

=

""

For Arabica, you can set seeds manually in the

HOME/.celestia-app/config/config.toml file:

Comma separated list of seed nodes to connect to

seeds

=

"" Optional: Set persistent peers Optionally, you can set persistent peers in yourconfig.toml file. If you set persistent peers, your node willalways try to connect to these peers. This is useful when running a local devnet, for example, when you would always want to connect to the same local nodes in your devnet. In production, setting persistent peers is advised only if you are running asentry node .

You can get the persistent peers from the@cosmos/chain-registry repository (for Mainnet Beta) or@celestiaorg/networks repository repo (for Mocha and Arabica) with the following commands:

Mainnet Beta

Mocha

Arabica bash PERSISTENT_PEERS = (curl

-s https://raw.githubusercontent.com/cosmos/chain-registry/master/celestia/chain.json |

jq

-r '.peers.persistent_peers[].address' |

tr '\n' ',' |

sed 's/,/\n/') echo PERSISTENT_PEERS sed

-i.bak

```
-e

"s/^persistent_peers =./persistent_peers = \" PERSISTENT_PEERS \" /" HOME /.celestia-app/config/config.toml
PERSISTENT_PEERS = ( curl

-s https://raw.githubusercontent.com/cosmos/chain-registry/master/celestia/chain.json |

jq

-r '.peers.persistent_peers[].address' |

tr '\n' ',' |

sed 's/,/\n/') echo PERSISTENT_PEERS sed

-i.bak

-e

"s/^persistent_peers =./persistent_peers = \" PERSISTENT_PEERS \" /" HOME /.celestia-app/config/config.toml bash
PERSISTENT_PEERS = ( curl

-sL https://raw.githubusercontent.com/celestiaorg/networks/master/mocha-4/peers.txt |

tr '\n' ',') echo PERSISTENT_PEERS sed

-i.bak

-e

"s/^persistent_peers =./persistent_peers = \" PERSISTENT_PEERS \" /" HOME /.celestia-app/config/config.toml
PERSISTENT_PEERS = ( curl

-sL https://raw.githubusercontent.com/celestiaorg/networks/master/mocha-4/peers.txt |

tr '\n' ',') echo PERSISTENT_PEERS sed

-i.bak

-e

"s/^persistent_peers =./persistent_peers = \" PERSISTENT_PEERS \" /" HOME /.celestia-app/config/config.toml bash
PERSISTENT_PEERS = ( curl

-sL https://raw.githubusercontent.com/celestiaorg/networks/master/arabica-11/peers.txt |

tr '\n' ',') echo PERSISTENT_PEERS sed

-i.bak

-e

"s/^persistent_peers =./persistent_peers = \" PERSISTENT_PEERS \" /" HOME /.celestia-app/config/config.toml
PERSISTENT_PEERS = ( curl

-sL https://raw.githubusercontent.com/celestiaorg/networks/master/arabica-11/peers.txt |

tr '\n' ',') echo PERSISTENT_PEERS sed

-i.bak

-e

"s/^persistent_peers =./persistent_peers = \" PERSISTENT_PEERS \" /" HOME /.celestia-app/config/config.toml
```

Storage and pruning configurations

Optional: Connect a consensus node to a bridge node

If your consensus node is being connected to a celestia-node bridge node, you will need to enable transaction indexing and retain all block data. This can be achieved with the following settings in yourconfig.toml .

Enable transaction indexing

toml indexer = "kv" indexer = "kv"

Retain all block data

And in yourapp.toml ,min-retain-blocks should remain as the default setting of0 :

toml min-retain-blocks = 0

retain all block data, this is default setting

min-retain-blocks = 0

retain all block data, this is default setting

Query transactions by hash

To query transactions using their hash, transaction indexing must be turned on. Set theindexer to"kv" in yourconfig.toml :

toml indexer = "kv" indexer = "kv"

Optional: Access historical state

If you want to query the historical state — for example, you might want to know the balance of a Celestia wallet at a given height in the past — you should run an archive node withpruning = "nothing" in yourapp.toml . Note that this configuration is resource-intensive and will require significant storage:

toml pruning = "nothing" pruning = "nothing"

Save on storage requirements

If you want to save on storage requirements, consider usingpruning = "everything" in yourapp.toml to prune everything. If you select"everything" or"default" , but still want to keep the block data, you can do so by not changing the default value ofmin-retain-blocks = 0 in yourapp.toml . A value of0 formin-retain-blocks will keep all block data. This will prune snapshots of the state, but it will keep block data:

toml pruning = "everything" min-retain-blocks = 0

this is the default setting

pruning = "everything" min-retain-blocks = 0

this is the default setting

Sync types

Sync mode Time Notes Block sync ~3 weeks Downloads and executes all blocks from genesis to the tip State sync ~1 hour Downloads a snapshot of the state then downloads and executes all blocks after that snapshot to the tip. Quick sync ~5 hours Downloads the data directory from a node. Time depends on your download speed because the data being downloaded can exceed 1 TB for mainnet.

Option 1: Block sync

By default, a consensus node will sync using block sync; which will request, validate and execute every block up to the head of the blockchain. This is the most secure mechanism yet the slowest (taking up to weeks depending on the height of the blockchain).

There are two alternatives for quicker syncing.

Option 2: State sync

State sync uses light client verification to verify state snapshots from peers and then apply them. State sync relies on weak subjectivity; a trusted header (specifically the hash and height) must be provided. This can be found by querying a trusted RPC endpoint (/block). RPC endpoints are also required for retrieving light blocks. These can be found in the docs here under the respective networks or fromthe chain-registry .

InHOME/.celestia-app/config/config.toml , set

toml rpc_servers = "" trust_height = 0 trust_hash = "" rpc_servers = "" trust_height = 0 trust_hash = "" And also set statesync totrue :

toml

State Sync Configuration Options

[statesync] enable = true

State Sync Configuration Options

[statesync] enable = true To their respective fields. At least two different rpc endpoints should be provided. The more, the greater the chance of detecting any fraudulent behavior.

Once setup, you should be ready to start the node as normal. In the logs, you should see:Discovering snapshots . This may take a few minutes before snapshots are found depending on the network topology.

TIP

If you are looking to quickly sync a consensus node, and do not need historical blocks, you can use the following scripts and state sync. Remember to checkout to the correct version and runmake install before running the scripts:

- Local devnet:https://github.com/celestiaorg/celestia-app/blob/main/scripts/single-node.sh
- Arabica:https://github.com/celestiaorg/celestia-app/blob/main/scripts/arabica.sh
- Mocha:https://github.com/celestiaorg/celestia-app/blob/main/scripts/mocha.sh
- Mainnet Beta:https://github.com/celestiaorg/celestia-app/blob/main/scripts/mainnet.sh

The public networks will use state sync so they'll get to the tip very quickly, but won't work for your use case if you need historical blocks.

Option 3: Quick sync

Quick sync effectively downloads the entiredata directory from a third-party provider meaning the node has all the application and blockchain state as the node it was copied from.

Run the following command to quick-sync from a snapshot:

Mainnet Beta

Mocha

Arabica bash cd HOME rm

-rf

~/.celestia-app/data mkdir

-p

~/.celestia-app/data SNAP_NAME = (curl

-s https://snaps.qubelabs.io/celestia/ |

\ egrep

-o ">celestia.*tar" |

tr

-d ">") aria2c

-x

16

-s

16

-o

celestia-snap.tar

"https://snaps.qubelabs.io/celestia{ SNAP_NAME }" tar

xf

celestia-snap.tar

-C

~/.celestia-app/data/ cd HOME rm

-rf

~/.celestia-app/data mkdir

-p

~/.celestia-app/data SNAP_NAME = (curl

-s https://snaps.qubelabs.io/celestia/ |

\ egrep

-o ">celestia.*tar" |

tr

-d ">") aria2c

-x

16

-s

16

-o

celestia-snap.tar

"https://snaps.qubelabs.io/celestia{ SNAP_NAME }" tar

xf

celestia-snap.tar

-C

~/.celestia-app/data/ bash cd HOME rm

-rf

~/.celestia-app/data mkdir

-p

~/.celestia-app/data SNAP_NAME = (curl

-s https://snaps.qubelabs.io/celestia/ |

\ egrep

-o ">mocha-4.*tar" |

tr

```
-d ">") aria2c
-x
16
-s
16
-o
celestia-snap.tar
"https://snaps.qubelabs.io/celestia{ SNAP_NAME }" tar
xf
celestia-snap.tar
-C
~/.celestia-app/data/ cd HOME rm
-rf
~/.celestia-app/data mkdir
-p
~/.celestia-app/data SNAP_NAME = ( curl
-s https://snaps.qubelabs.io/celestia/ |
\ egrep
-o ">mocha-4.*tar" |
tr
-d ">") aria2c
-x
16
-s
16
-o
celestia-snap.tar
"https://snaps.qubelabs.io/celestia{ SNAP_NAME }" tar
xf
celestia-snap.tar
-C
~/.celestia-app/data/ bash cd HOME rm
-rf
~/.celestia-app/data mkdir
-p
~/.celestia-app/data SNAP_NAME = ( curl
-s https://snaps.qubelabs.io/celestia/ |
\ egrep
-o ">arabica-11.*tar" |
tr
-d ">") aria2c
-x
16
-s
16
-o
celestia-snap.tar
"https://snaps.qubelabs.io/celestia{ SNAP_NAME }" tar
xf
celestia-snap.tar
-C
~/.celestia-app/data/ cd HOME rm
-rf
~/.celestia-app/data mkdir
-p
~/.celestia-app/data SNAP_NAME = ( curl
-s https://snaps.qubelabs.io/celestia/ |
\ egrep
-o ">arabica-11.*tar" |
tr
-d ">") aria2c
-x
16
-s
16
-o
celestia-snap.tar
"https://snaps.qubelabs.io/celestia{ SNAP_NAME }" tar
xf
celestia-snap.tar
-C
~/.celestia-app/data/
```

Start the consensus node

If you are running celestia-app v1.x.x:

sh celestia-appd

start celestia-appd

start If you are running celestia-app >= v2.0.0: then you'll want to start the node with a--v2-upgrade-height that is dependent on the network. The--v2-upgrade-height flag is only needed during the v2 upgrade height so after your node has executed the upgrade (e.g. you see the logupgraded from app version 1 to 2), you don't need to provide this flag for futurecelestia-appd start invocations.

Mainnet Beta

Mocha

Arabica sh celestia-appd

start

--v2-upgrade-height

2371495 celestia-appd

start

--v2-upgrade-height

2371495 sh celestia-appd

start

--v2-upgrade-height

2585031 celestia-appd

start

--v2-upgrade-height

2585031 sh celestia-appd

start

--v2-upgrade-height

1751707 celestia-appd

start

--v2-upgrade-height

1751707 Optional: If you would like celestia-app to run as a background process, you can follow the SystemD tutorial .

Extra resources for consensus nodes

Optional: Reset network

This will delete all data folders so we can start fresh:

sh celestia-appd

tendermint

unsafe-reset-all

--home HOME /.celestia-app celestia-appd

tendermint

unsafe-reset-all

--home HOME /.celestia-app

Optional: Configure an RPC endpoint

You can configure your consensus node to be a public RPC endpoint. This allows it to accept connections from data availability nodes and serve requests for the data availability API.

Expose RPC

By default, the RPC service listens onlocalhost which means it can't be accessed from other machines. To make the RPC service available publicly, you need to bind it to a public IP or0.0.0.0 (which means listening on all available network interfaces).

You can do this by editing the config.toml file:

sh sed

-i

's#"tcp://127.0.0.1:26657"#"tcp://0.0.0.0:26657"#g'

~/.celestia-app/config/config.toml sed

-i

's#"tcp://127.0.0.1:26657"#"tcp://0.0.0.0:26657"#g'

~/.celestia-app/config/config.toml This command replaces thelocalhost IP address with0.0.0.0 , making the RPC service listen on all available network interfaces.

Note onexternal-address

Theexternal-address field in the configuration is used when your node is behind a NAT and you need to advertise a different address for peers to dial. Populating this field is not necessary for making the RPC endpoint public.

sh EXTERNAL-ADDRESS = (wget

-qO- eth0.me) sed

-i.bak

-e

"s/^external-address = ""/external-address = " EXTERNAL -ADDRESS:26656"/"

\ HOME /.celestia-app/config/config.toml EXTERNAL-ADDRESS = (wget

-qO- eth0.me) sed

-i.bak

-e

"s/^external-address = ""/external-address = " EXTERNAL -ADDRESS:26656"/"

\ HOME /.celestia-app/config/config.toml

Restart the node

After making these changes, restartcelestia-appd to load the new configurations.

Optional: Transaction indexer configuration options

This section guides you on how to configure yourconfig.toml file incelestia-app to select which transactions to index. Depending on the application's configuration, a node operator may decide which transactions to index.

The available options are:

1. null
2. : This option disables indexing. If you don't need to query transactions, you can choose this option to save space.
3. kv
4. (default): This is the simplest indexer, backed by key-value storage (defaults to levelDB; see DBBackend). Whenkv
5. is chosen,tx.height

6. andtx.hash
7. will always be indexed. This option is suitable for basic queries on transactions.
8. psql
9. : This indexer is backed by PostgreSQL. When psql is chosen,tx.height
10. andtx.hash
11. will always be indexed. This option is suitable for complex queries on transactions.

An example to set the value tokv inconfig.toml is:

toml indexer = "kv" indexer = "kv" Remember to restartcelestia-appd after making changes to the configuration to load the new settings.

Optional: Discard ABCI responses configuration

This section will guide you on how to configure yourconfig.toml file incelestia-app to manage the storage of ABCI responses. ABCI responses are the results of executing transactions and are used for/block_results RPC queries and to reindex events in the command-line tool.

Thediscard_abci_responses option allows you to control whether these responses are persisted in the state store:

- false
- (default): ABCI responses are stored in the state store. This ensures that ABCI responses are available for/block_results
- RPC queries and for reindexing events. However, it can consume a significant amount of disk space.
- true
- : ABCI responses are not stored in the state store. This can save a considerable amount of disk space, but/block_results
- RPC queries and event reindexing will not be available.

An example to set the value to false inconfig.toml is:

toml discard_abci_responses = false discard_abci_responses = false Remember to restartcelestia-appd after making changes to the configuration to load the new settings.

FAQ

+2/3 committed an invalid block: wrong Block.Header.Version

If you encounter an error like:

bash 2024-04-25

14 :48:24

6 :48PM

ERR

CONSENSUS

FAILURE!!!

err="+2/3 committed an invalid block: wrong Block.Header.Version. Expected {11 1}, got {11 2}"

module=consensus

stack="goroutine 214 [running]:\nruntime/debug.Stack()\n\t/usr/local/go/src/runtime/debug/stack.go:24 +0x64\ngithub.com/tendermint/tendermint/consensus.
(*State).receiveRoutine.func2()\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:746 +0x44\npanic({0x1b91180?, 0x400153b240?})\n\t/usr/local/go/src/runtime/panic.go:770 +0x124\ngithub.com/tendermint/tendermint/consensus.(*State).finalizeCommit(0x400065ea88, 0x3)\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:1637 +0xd30\ngithub.com/tendermint/tendermint/consensus.(*State).tryFinalizeCommit(0x400065ea88, 0x3)\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:1606 +0x26c\ngithub.com/tendermint/tendermint/consensus.(*State).handleCompleteProposal(0x400065ea88, 0x3)\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:2001 +0x2d8\ngithub.com/tendermint/tendermint/consensus.(*State).handleMsg(0x400065ea88, {{0x2b30a00, 0x400143e048},*

*{0x40002a61b0, 0x28}})\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:856 +0x1c8\ngithub.com/tendermint/tendermint/consensus.(*State).receiveRoutine(0x400065ea88, 0x0)\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:782 +0x2c4\ncreated by github.com/tendermint/tendermint/consensus.(*State).OnStart in goroutine 169\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:391 +0x110\n" 2024-04-25

14 :48:24

6 :48PM

ERR

CONSENSUS

FAILURE!!!

err="+2/3 committed an invalid block: wrong Block.Header.Version. Expected {11 1}, got {11 2}"

module=consensus

stack="goroutine 214 [running]:\nruntime/debug.Stack()\n\t/usr/local/go/src/runtime/debug/stack.go:24 +0x64\ngithub.com/tendermint/tendermint/consensus.
(*State).receiveRoutine.func2()\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:746 +0x44\npanic({0x1b91180?, 0x400153b240?})\n\t/usr/local/go/src/runtime/panic.go:770 +0x124\ngithub.com/tendermint/tendermint/consensus.(*State).finalizeCommit(0x400065ea88, 0x3)\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:1637 +0xd30\ngithub.com/tendermint/tendermint/consensus.(*State).tryFinalizeCommit(0x400065ea88, 0x3)\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:1606 +0x26c\ngithub.com/tendermint/tendermint/consensus.(*State).handleCompleteProposal(0x400065ea88, 0x3)\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:2001 +0x2d8\ngithub.com/tendermint/tendermint/consensus.(*State).handleMsg(0x400065ea88, {{0x2b30a00, 0x400143e048}, {0x40002a61b0, 0x28}})\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:856 +0x1c8\ngithub.com/tendermint/tendermint/consensus.(*State).receiveRoutine(0x400065ea88, 0x0)\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:782 +0x2c4\ncreated by github.com/tendermint/tendermint/consensus.(*State).OnStart in goroutine 169\n\t/go/pkg/mod/github.com/celestiaorg/[email protected] /consensus/state.go:391 +0x110\n" then it is likely that the network has upgraded to a new app version but your consensus node was not prepared for the upgrade. To fix this, you'll need to:

1. Remove DBs from your CELESTIA_HOME directory via:celestia-appd tendermint reset-state
2. .
3. Remove thedata/application.db
4. inside your CELESTIA_HOME directory.
5. Download the latest binary for your network.
6. Restart your consensus node with the relevant--v2-upgrade-height
7. for the network you're running on. [[Edit this page on GitHub] Last updated: Previous page Bridge node Next page Validator node []