Troubleshooting

Network selection

Note: If you do not select a network, the default network will be Mainnet Beta.

sh celestia

< node-typ e

init

--p2p.network

< networ k

celestia

< node-typ e

start

--p2p.network

< networ k

--core.ip

< UR I

celestia

< node-typ e

init

--p2p.network

< networ k

celestia

< node-typ e

start

--p2p.network

< networ k

--core.ip

< UR I

TIP

Refer tothe ports section of this page for information on which ports are required to be open on your machine. NOTE

It is advised before switching networks to reinitialize your node viainit command. This is due to an old config being present. Re-initialisation will reset the config.

Chain ID

When interacting with celestia-node, it is important to take into account the different chain IDs for different networks. For Mainnet Beta, there is no need to declare a chain ID, as the default is celestia,i.e. no--p2p.network string flag is required for Mainnet Beta.

Network Chain ID --p2p.network string Mainnet Beta celestia not required (--p2p.network celestia) Mocha mocha-4 --p2p.network mocha Arabica arabica-11 --p2p.network arabica

Ports

When interacting with a Celestia node, you may need to open ports on your machine to allow communication between nodes, such as bridge nodes. It is essential that specific ports are accessible. Make sure that your firewall allows connections to the correct ports.

If you run a node on a cloud server, make sure that the ports are open on the server's firewall. If you run a node at home, make sure that your router allows connections to the correct ports.

For example, validator ports 9090 and 26657 need to be accessible by the bridge, and port 2121 is required for P2P connections for all node types.

The following ports are used by Celestia nodes:

Port Protocol Address Description Enabled by default on node Flag 2121 TCP/UDP localhost P2P true N/A 26658 HTTP localhost RPC true --rpc.port string 26659 HTTP localhost REST Gateway false --gateway.port string WARNING

The gateway endpoints have been deprecated and will be removed in the future. If you would like to use them anyway, you canfind more details on GitHub .

Changing the location of your node store

Background

Anenhancement has been made in v0.14.0+ to automate the detection of the running node, eliminating the need to manually specify the--node.store flag for each RPC request.

Assumptions:

- The presence of a lock signifies a running node.
- Networks are ordered as Mainnet Beta, Mocha, Arabica, private, custom.
- Node types are ordered as bridge, full, and light.
- Each network has only one running node type.
- Multiple nodes of the same network and type are prohibited (resulting in anError: node: store is in use
-).

Key Points:

- Authentication token and other flags maintain their previous behavior and take precedence.
- Address and port details are fetched from the configuration.
- skipAuth
- allows bypassing authentication for trusted setups and follows Unix daemon conventions.
- Non-default node store and cel-key configurations still require specific flags in the configuration settings.

Demonstration

In this section, we'll guide you through starting your node using a node store in a different location than you originally started with.

First, stop your node safely usingcontrol + C .

Then, init your node again with a new node store:

bash celestia

< node-typ e

init

--node.store

/home/user/celestia- < node-typ e

-location/

\ --p2p.network

mocha celestia

< node-typ e

init

--node.store

/home/user/celestia- < node-typ e

-location/

\ --p2p.network

mocha Next, start your node:

bash celestia

full

start

--core.ip

rpc-mocha.pops.one

--p2p.network

mocha

\ --node.store

/home/user/celestia- < node-typ e

-location/ celestia

full

start

--core.ip

rpc-mocha.pops.one

--p2p.network

mocha

\ --node.store

/home/user/celestia- < node-typ e

-location/ If you choose to change the location of your node store, you will need to execute each command on your node with the following flag:

bash --node.store

/home/user/celestia- < node-typ e

-location/ --node.store

/home/user/celestia- < node-typ e

-location/ When usingcel-key , the process is different. To show the keys you should add--keyring-dir like this example:

bash ./cel-key

list

--p2p.network

mocha

--node.type

full

\ --keyring-dir

/home/user/celestia- < node-typ e

-location/keys/ ./cel-key

list

--p2p.network

mocha

--node.type

full

\ --keyring-dir

/home/user/celestia- < node-typ e

-location/keys/

Examples

Mainnet Beta full and Mocha light

This example uses a Mainnet Beta full node and a Mocha light node. When making the request:

bash ❯

celestia

blob

get

1318129

0x42690c204d39600fddd3

0 MFhYKQUi2BU+U1jxPzG7QY2BVV1lb3kiU+zAK7nUiY= { "result" :

"RPC client error: sendRequest failed: http status 401 Unauthorized unmarshaling response: EOF" } ❯

celestia

blob

get

1318129

0x42690c204d39600fddd3

0 MFhYKQUi2BU+U1jxPzG7QY2BVV1lb3kiU+zAK7nUiY= { "result" :

"RPC client error: sendRequest failed: http status 401 Unauthorized unmarshaling response: EOF" } The request will go to the Mainnet Beta node, and a 401 will show in this node's logs. Note that a 401 is expected because this blob was posted to Mocha and neither the namespace nor the blob exist on Mainnet Beta.

Mocha full and Arabica light

This example uses a Mocha full node and an Arabica light node. When making the request:

bash ❯

celestia

blob

get

1318129

0x42690c204d39600fddd3

0 MFhYKQUi2BU+U1jxPzG7QY2BVV1lb3kiU+zAK7nUiY= { "result" :

{ "namespace" :

"AAAAAAAAAAAAAAAAAAAAAAAAAEJpDCBNOWAP3dM=", "data" :

"0x676d", "share_version" :

0 , "commitment" :

"0MFhYKQUi2BU+U1jxPzG7QY2BVV1lb3kiU+zAK7nUiY=", "index" :

23 } } ❯

celestia

blob

get

1318129

0x42690c204d39600fddd3

0 MFhYKQUi2BU+U1jxPzG7QY2BVV1lb3kiU+zAK7nUiY= { "result" :

{ "namespace" :

"AAAAAAAAAAAAAAAAAAAAAAAAAEJpDCBNOWAP3dM=", "data" :

"0x676d", "share_version" :

0 , "commitment" :

"0MFhYKQUi2BU+U1jxPzG7QY2BVV1lb3kiU+zAK7nUiY=", "index" :

23 } } The request will go to the Mocha full node, and result shown as expected.

Using a custom rpc.config address

When using a custom RPC config address0.0.0.1 and port25231 , the CLI accurately routes to the custom address and port, where no node is running. It fails as expected:

bash ❯

celestia

blob

get

1318129

0x42690c204d39600fddd3

0 MFhYKQUi2BU+U1jxPzG7QY2BVV1lb3kiU+zAK7nUiY= { "result" :

"RPC client error: sendRequest failed: Post \" http://0.0.0.1:25231 \" : dial tcp 0.0.0.1:25231: connect: no route to host" } ❯

celestia

blob

get

1318129

0x42690c204d39600fddd3

0 MFhYKQUi2BU+U1jxPzG7QY2BVV1lb3kiU+zAK7nUiY= { "result" :

"RPC client error: sendRequest failed: Post \" http://0.0.0.1:25231 \" : dial tcp 0.0.0.1:25231: connect: no route to host" }

Resetting your config

If you an encounter an error, it is likely that an old config file is present:

sh Error:

nodebuilder/share:

interval

must

be

positive ; nodebuilder/core:

invalid

IP

addr

given:

or

Error:

nodebuilder/share:

interval

must

be

positive Error:

nodebuilder/share:

interval

must

be

positive ; nodebuilder/core:

invalid

IP

addr

given:

or

Error:

nodebuilder/share:

interval

must

be

positive You can re-initialize your node's config with the following commands:

TIP

Save your config so custom values are not lost. Run the following command to update your config:

bash celestia

< node-typ e

config-update

--p2p.network

< networ k

celestia

< node-typ e

config-update

--p2p.network

< networ k

This will pull in any new values from new configuration and merge them into the existing configuration.

TIP

After using theconfig-update command, it is encouraged to double-check that your custom values are preserved. Then, to start your node again:

bash celestia

< node-typ e

start

--p2p.network

< networ k

celestia

< node-typ e

start

--p2p.network

< networ k

Clearing the data store

Forbridge, full, and light nodes , remove the data store with this command:

bash celestia

< node-typ e

unsafe-reset-store

--p2p.network

< networ k

celestia

< node-typ e

unsafe-reset-store

--p2p.network

< networ k

bash celestia

light

unsafe-reset-store

--p2p.network

mocha celestia

light

unsafe-reset-store

--p2p.network

mocha

FATAL headers given to the heightSub are in the wrong order

If you observe a FATAL log line like:

bash FATAL

header/store

store/heightsub.go:87

PLEASE

FILE

A

BUG

REPORT:

headers

given

to

the

heightSub

are

in

the

wrong

order" FATAL

header/store

store/heightsub.go:87

PLEASE

FILE

A

BUG

REPORT:

headers

given

to

the

heightSub

are

in

the

wrong

order" then it is possible the celestia-nodedata/ directory contains headers from a previous instance of the network that you are currently trying to run against. One resolution strategy is to delete the existing celestia-node config for the target network and re-initialize it:

sh

rm -rf ~/.celestia--

rm

-rf

~/.celestia-bridge-private

celestia init --p2p.network

celestia

bridge

init

--p2p.network

private

rm -rf ~/.celestia--

rm

-rf

~/.celestia-bridge-private

celestia init --p2p.network

celestia

bridge

init

--p2p.network

private

Error: "too many open files"

When running a Celestia bridge node, you may encounter an error in the logs similar to this:

bash Error

while

creating

log

file

in

valueLog.open

error:

while

opening

file:

/opt/celestia/.celestia-bridge/data/003442.vlog

error:

open

/opt/celestia/.celestia-bridge/data/003442.vlog:

too

many

open

files Error

while

creating

log

file

in

valueLog.open

error:

while

opening

file:

/opt/celestia/.celestia-bridge/data/003442.vlog

error:

open

/opt/celestia/.celestia-bridge/data/003442.vlog:

too

many

open

files This error indicates that the Celestia application is trying to open more files than the operating system's limit allows. To

fix this, you will need to edit the Celestia bridge service file to increase the number of file descriptors that the service can open.

1. Open the service file for editing:

bash nano

/etc/systemd/system/celestia-bridge.service nano

/etc/systemd/system/celestia-bridge.service 1. Modify theLimitNOFILE 2. parameter:

In the service file, find theLimitNOFILE parameter under the[Service] section and set its value to1400000 . It should look like this:

ini [Service] ... LimitNOFILE =1400000 ... [Service] ... LimitNOFILE =1400000 ... NOTE

Be cautious when increasing file descriptor limits. Setting this value too high might affect system performance. Ensure the value is appropriate for your system's capabilities. 1. Reload daemon and restart bridge service:

bash sudo

systemctl

daemon-reload sudo

systemctl

daemon-reload bash sudo

systemctl

restart

celestia-bridge sudo

systemctl

restart

celestia-bridge [][Edit this page on GitHub] Last updated: Previous page Syncing a light node from a trusted hash Next page Metrics, visualization, and alerts []