Setting up your node as a background process with SystemD

SystemD is a daemon service useful for running applications as background processes.

Consensus nodes

If you are running a validator or consensus node, here are the steps to setting upcelestia-appd as a background process.

Start the celestia-app with SystemD

SystemD is a daemon service useful for running applications as background processes.

Create Celestia-App systemd file:

sh sudo

tee

<< EOF

> /dev/null /etc/systemd/system/celestia-appd.service [Unit] Description=celestia-appd Cosmos daemon After=network-online.target

[Service] User= USER ExecStart=(which celestia-appd) start Restart=on-failure RestartSec=3 LimitNOFILE=65535

[Install] WantedBy=multi-user.target EOF sudo

tee

<< EOF

> /dev/null /etc/systemd/system/celestia-appd.service [Unit] Description=celestia-appd Cosmos daemon After=network-online.target

[Service] User= USER ExecStart=(which celestia-appd) start Restart=on-failure RestartSec=3 LimitNOFILE=65535

[Install] WantedBy=multi-user.target EOF If the file was created successfully you will be able to see its content:

sh cat

/etc/systemd/system/celestia-appd.service cat

/etc/systemd/system/celestia-appd.service Enable and startcelestia-appd daemon:

sh sudo

systemctl

enable

celestia-appd sudo

systemctl

start

celestia-appd sudo

systemctl

enable

celestia-appd sudo

systemctl

start

celestia-appd Check if daemon has been started correctly:

sh sudo

systemctl

status

celestia-appd sudo

systemctl

status

celestia-appd Check daemon logs in real time:

sh sudo

journalctl

-u

celestia-appd.service

-f sudo

journalctl

-u

celestia-appd.service

-f To check if your node is in sync before going forward:

sh curl

-s

localhost:26657/status

|

jq

.result

|

jq

.sync_info curl

-s

localhost:26657/status

|

jq

.result

|

jq

.sync_info Make sure that you have"catching_up": false , otherwise leave it running until it is in sync.

Data availability nodes

Celestia full storage node

Create Celestia full storage node systemd file:

sh sudo

tee

<< EOF

/dev/null /etc/systemd/system/celestia-full.service [Unit] Description=celestia-full Cosmos daemon After=network-online.target

[Service] User= USER ExecStart=(which celestia) full start Restart=on-failure RestartSec=3 LimitNOFILE=1400000

[Install] WantedBy=multi-user.target EOF sudo

tee

<< EOF

/dev/null /etc/systemd/system/celestia-full.service [Unit] Description=celestia-full Cosmos daemon After=network-online.target

[Service] User= USER ExecStart=(which celestia) full start Restart=on-failure RestartSec=3 LimitNOFILE=1400000

[Install] WantedBy=multi-user.target EOF If the file was created successfully you will be able to see its content:

sh cat

/etc/systemd/system/celestia-full.service cat

/etc/systemd/system/celestia-full.service Enable and start celestia-full daemon:

sh sudo

systemctl

enable

celestia-full sudo

systemctl

start

celestia-full && sudo

journalctl

-u

\ celestia-full.service -f sudo

systemctl

enable

celestia-full sudo

systemctl

start

celestia-full && sudo

journalctl

-u

\ celestia-full.service -f You should be seeing logs coming through of the full storage node syncing.

Celestia bridge node

Create Celestia Bridge systemd file:

sh sudo

tee

<< EOF

> /dev/null /etc/systemd/system/celestia-bridge.service [Unit] Description=celestia-bridge Cosmos daemon After=network-online.target

[Service] User= USER ExecStart=(which celestia) bridge start Restart=on-failure RestartSec=3 LimitNOFILE=1400000

[Install] WantedBy=multi-user.target EOF sudo

tee

<< EOF

> /dev/null /etc/systemd/system/celestia-bridge.service [Unit] Description=celestia-bridge Cosmos daemon After=network-online.target

[Service] User= USER ExecStart=(which celestia) bridge start Restart=on-failure RestartSec=3 LimitNOFILE=1400000

[Install] WantedBy=multi-user.target EOF If the file was created successfully you will be able to see its content:

sh cat

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

/etc/systemd/system/celestia-bridge.service Enable and start celestia-bridge daemon:

sh sudo

systemctl

enable

celestia-bridge sudo

systemctl

start

celestia-bridge && sudo

journalctl

-u

\ celestia-bridge.service -f sudo

systemctl

enable

celestia-bridge sudo

systemctl

start

celestia-bridge && sudo

journalctl

-u

\ celestia-bridge.service -f Now, the Celestia bridge node will start syncing headers and storing blocks fromcelestia-app .

NOTE

At startup, we can see themultiaddress from Celestia bridge node. This isneeded for future light node connections and communication between Celestia Bridge Nodes Example:

sh NODE_IP =< UR I

>] /ip4NODE_IP/tcp/2121/p2p/12D3KooWD5wCBJXKQuDjhXFjTFMrZoysGVLtVht5hMoVbSLCbV22 NODE_IP =< UR I

] /ip4NODE_IP/tcp/2121/p2p/12D3KooWD5wCBJXKQuDjhXFjTFMrZoysGVLtVht5hMoVbSLCbV22 You should be seeing logs coming through of the bridge node syncing.

Celestia light node

Start the light node as daemon process in the background

sh sudo

tee

<< EOF

/dev/null /etc/systemd/system/celestia-lightd.service [Unit] Description=celestia-lightd light node After=network-online.target

[Service] User= USER ExecStart=(which celestia) light start --core.ipRestart=on-failure RestartSec=3

[Install] WantedBy=multi-user.target EOF sudo

tee

<< EOF

/dev/null /etc/systemd/system/celestia-lightd.service [Unit] Description=celestia-lightd light node After=network-online.target

[Service] User= USER ExecStart=(which celestia) light start --core.ipRestart=on-failure RestartSec=3

[Install] WantedBy=multi-user.target EOF If the file was created successfully you will be able to see its content:

sh cat

/etc/systemd/system/celestia-lightd.service cat

/etc/systemd/system/celestia-lightd.service Enable and start celestia-lightd daemon:

sh sudo

systemctl

enable

celestia-lightd sudo

systemctl

start

celestia-lightd sudo

systemctl

enable

celestia-lightd sudo

systemctl

start

celestia-lightd Check if daemon has been started correctly:

sh sudo

systemctl

status

celestia-lightd sudo

systemctl

status

celestia-lightd Check daemon logs in real time:

sh sudo

journalctl

-u

celestia-lightd.service

-f sudo

journalctl

-u

celestia-lightd.service

-f Now, the Celestia light node will start syncing headers. After sync is finished, light node will do Data Availability Sampling (DAS) from the bridge node. [][Edit this page on GitHub] Last updated: Previous page Create a vesting account Next page Network upgrade process []