

Setting up a Celestia validator node

This tutorial will guide you through setting up a validator node on Celestia. Validator nodes allow you to participate in consensus in the Celestia network.

Hardware requirements

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

- Memory:8 GB RAM
- CPU:6 cores
- Disk:500 GB SSD Storage
- Bandwidth:1 Gbps for Download/1 Gbps for Upload

Setting up a validator node

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

First, follow the instructions on[setting up a full consensus node](#).

Wallet

Follow[the tutorial on creating a wallet](#).

Delegate stake to a validator

Create an environment variable for the address:

```
bash VALIDATOR_WALLET =< validator-wallet-name
```

```
VALIDATOR_WALLET =< validator-wallet-name
```

If you want to delegate more stake to any validator, including your own you will need the celestia address of the validator in question. You can run the command below to get the celestia address of your local validator wallet in case you want to delegate more to it:

```
bash celestia-appd
```

```
keys
```

```
show VALIDATOR_WALLET --bech
```

```
val
```

```
-a celestia-appd
```

```
keys
```

```
show VALIDATOR_WALLET --bech
```

```
val
```

-a After entering the wallet passphrase you should see a similar output:

```
bash Enter
```

```
keyring
```

```
passphrase: celestiaoper1q3v5cugc8cdpud87u4zwy0a74uxkk6u43cv6hd Enter
```

```
keyring
```

passphrase: celestiaoper1q3v5cugc8cdpud87u4zwy0a74uxkk6u43cv6hd To delegate tokens to the celestia validator, as an example you can run:

```
bash celestia-appd
```

```
tx
```

```
staking
```

delegate

```
\ celestiavalidator1q3v5cugc8cdpud87u4zwy0a74uxkk6u4q4gx4p 1000000 utia
```

```
\ --from=VALIDATOR_WALLET --chain-id=mocha-4
```

```
\ --fees=21000utia celestia-appd
```

tx

staking

delegate

```
\ celestiavalidator1q3v5cugc8cdpud87u4zwy0a74uxkk6u4q4gx4p 1000000 utia
```

```
\ --from=VALIDATOR_WALLET --chain-id=mocha-4
```

```
\ --fees=21000utia
```

 If successful, you should see a similar output as:

```
console code: 0 codespace: "" data: "" gas_used: "0" gas_wanted: "0" height: "0" info: "" logs: [] raw_log: '[]' timestamp: "" tx:
null txhash: code: 0 codespace: "" data: "" gas_used: "0" gas_wanted: "0" height: "0" info: "" logs: [] raw_log: '[]' timestamp: ""
tx: null txhash: You can check if the TX hash went through using the block explorer by inputting the txhash ID that was
returned.
```

Optional: Deploy the celestia-node

Running a bridge node is critical to the Celestia network as it enables the data availability and consensus nodes to communicate with one another. It is recommended to support the data availability network, but is not required for celestia-app.

If you are not running a bridge node, you can skip to [run a validator node](#).

This section describes part 2 of Celestia validator node setup: running a Celestia bridge node daemon.

Install celestia-node

You can [follow the tutorial for installing celestia-node](#)

Initialize the bridge node

Run the following:

```
bash celestia
```

```
bridge
```

```
init
```

```
--core.ip
```

```
< URL
```

```
celestia
```

```
bridge
```

```
init
```

```
--core.ip
```

```
< URL
```

```
TIP
```

Refer to [the ports section of the celestia-node troubleshooting page](#) for information on which ports are required to be open on your machine. If you need a list of RPC endpoints to connect to, you can find the [list on the Mocha testnet page](#) or [list on the Arabica devnet page](#).

Run the bridge node

Run the following:

```
bash celestia
```

```
bridge
```

```
start celestia
```

```
bridge
```

```
start
```

Optional: start the bridge node with SystemD

Follow [the tutorial on setting up the bridge node as a background process with SystemD](#).

You have successfully set up a bridge node that is syncing with the network.

Run the validator node

In order to start your validator node, run the following:

```
bash celestia-appd
```

```
start celestia-appd
```

start After completing all the necessary steps, you are now ready to run a validator! In order to create your validator onchain, follow the instructions below. Keep in mind that these steps are necessary ONLY if you want to participate in the consensus.

Pick a moniker name of your choice! This is the validator name that will show up on public dashboards and explorers. `VALIDATOR_WALLET` must be the same you defined previously. Parameter `--min-self-delegation=1000000` defines the amount of tokens that are self delegated from your validator wallet.

Now, connect to the network of your choice.

You have the following option of connecting to list of networks shown below:

Continuing the validator tutorial, here are the steps to connect your validator to Mocha:

```
bash MONIKER = "your_moniker" VALIDATOR_WALLET = "validator"
```

```
celestia-appd
```

```
tx
```

```
staking
```

```
create-validator
```

```
\ --amount=1000000utia
```

```
\ --pubkey= ( celestia-appd tendermint show-validator)
```

```
\ --moniker= MONIKER
```

```
\ --chain-id=mocha-4
```

```
\ --commission-rate=0.1
```

```
\ --commission-max-rate=0.2
```

```
\ --commission-max-change-rate=0.01
```

```
\ --min-self-delegation=1000000
```

```
\ --from= VALIDATOR_WALLET
```

```
\ --keyring-backend=test
```

```
\ --fees=21000utia
```

```
\ --gas=220000 MONIKER = "your_moniker" VALIDATOR_WALLET = "validator"
```

celestia-appd

tx

staking

create-validator

\ --amount=1000000utia

\ --pubkey= (celestia-appd tendermint show-validator)

\ --moniker= MONIKER

\ --chain-id=mocha-4

\ --commission-rate=0.1

\ --commission-max-rate=0.2

\ --commission-max-change-rate=0.01

\ --min-self-delegation=1000000

\ --from= VALIDATOR_WALLET

\ --keyring-backend=test

\ --fees=21000utia

\ --gas=220000 You will be prompted to confirm the transaction:

console confirm transaction before signing and broadcasting [y/N]: y confirm transaction before signing and broadcasting [y/N]: y Inputtingy should provide an output similar to:

console code: 0 codespace: "" data: "" gas_used: "0" gas_wanted: "0" height: "0" info: "" logs: [] raw_log: '[]' timestamp: "" tx: null txhash: code: 0 codespace: "" data: "" gas_used: "0" gas_wanted: "0" height: "0" info: "" logs: [] raw_log: '[]' timestamp: "" tx: null txhash: You should now be able to see your validator from[a block explorer](#)

Submit your validator information

After starting your node, please submit your node as a seed and peer to the[networks repository](#) .

Optional: Transaction indexer configuration options

Follow the instructions under[transaction indexer configuration options](#) to configure yourconfig.toml file to select which transactions to index.

Additional resources

For additional resources, refer to[the extra resources for consensus nodessection of the full consensus node page](#). [\[\[Edit this page on GitHub\]](#) Last updated: [Previous page Full consensus node](#) [Next page IBC relaying guide](#) [\[](#)