

# Deploy an OP Stack devnet to Celestia

In order to deploy a devnet to Celestia, you will need to have a modified version of optimism-bedrock . Refer to the [steps to install dependencies and the modified version of OP Stack](#) for your environment setup.

## Pick your deployment type

Using Celestia and OP stack, you have the option to either run a light node of your own or a local-celestia-devnet , both of which will give you a local devnet to test things out with.

### Using a local devnet

If you'd like to use the local-celestia-devnet , you're in luck! This is the default for the OP Stack + Celestia repository. Head to the [previous page](#) to get started.

### Using a light node

This is a beta integration and we are working on resolving [open issues](#) .

In order to allow your light node to post and retrieve data without errors, you will need to change `UseShareExchange` to `false` in:

Mainnet Beta

Mocha

Arabica `bash HOME/.celestia-light/config.toml` `HOME/.celestia-light/config.toml` `bash HOME/.celestia-light-mocha-4/config.toml` `HOME/.celestia-light-mocha-4/config.toml` `bash HOME/.celestia-light-arabica-11/config.toml` `HOME/.celestia-light-arabica-11/config.toml` If you choose to use your own node store, the light node must be fully synced and funded for you to be able to submit and retrieve `PayForBlobs` to a Celestia network.

If it is not synced, you will run into [errors similar to this](#) .

Visit the [Arabica](#) or [Mocha](#) pages to visit their faucets.

In order to mount existing data, you must have a node store that is in this directory:

Mainnet Beta

Mocha

Arabica `bash HOME/.celestia-light` `HOME/.celestia-light` `bash HOME/.celestia-light-mocha-4` `HOME/.celestia-light-mocha-4` `bash HOME/.celestia-light-arabica-11` `HOME/.celestia-light-arabica-11` This is the default location of the node store when you initialize and run a new Celestia node.

By default, the node will run with the account named `my_celes_key` .

If you have your own setup you'd like to try, you can always edit `optimism/ops-bedrock/docker-compose.yml` to work with your setup.

### Using a RaaS provider

If you'd like to use a Rollups as a Service (RaaS) provider, you can do so by going to the RaaS category in the menu.

## Build the devnet

Build TypeScript definitions for TS dependencies:

```
bash cd HOME cd
```

```
optimism make cd HOME cd
```

optimism make Set environment variables to start network:

```
bash export SEQUENCER_BATCH_INBOX_ADDRESS = 0xff00000000000000000000000000000000 export L2OO_ADDRESS = 0x70997970C51812dc3A010C7d01b50e0d17dc79C8 export SEQUENCER_BATCH_INBOX_ADDRESS = 0xff00000000000000000000000000000000 export L2OO_ADDRESS = 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
```

## Start the devnet

First, make sure your light node is synced and funded. It must not be running for this example to work.

This example is for Mainnet Beta. You can modify the `da:` section of your `HOME/optimism/ops-bedrock/docker-compose.yml` for your specific use, similarly to the example below:

This setup will use `celestia-da`, which is `celestia-node` with a DA server on port 26650.

For the `P2P_NETWORK` variable, you'll need to supply the network of choice, either `celestia`, `mocha`, or `arabica`. Using `celestia`, the volume path will be just `celestia-light` instead of `celestia-light-`. You will also need to provide a `core.ip` RPC URL for the network you are using.

```
yml da : image : ghcr.io/rollkit/local-celestia-devnet:v0.12.1 image : ghcr.io/rollkit/celestia-da:v0.12.9 command :
```

```
celestia-da light start --p2p.network=--da.grpc.namespace=000008e5f679bf7116cb --
da.grpc.listen=0.0.0.0:26650 --core.ip --gateway environment : - NODE_TYPE=light - P2P_NETWORK= ports : -
"26650:26650" - "26658:26658" - "26659:26659" volumes : - HOME/.celestia-light-:/home/celestia/.celestia-
light-/ healthcheck : test : [ "CMD" , "curl" , "-f" , "http://localhost:26659/header/1" ] interval : 10s timeout : 5s
retries : 5 start_period : 30s da : image : ghcr.io/rollkit/local-celestia-devnet:v0.12.1 image : ghcr.io/rollkit/celestia-
da:v0.12.9 command :
```

```
celestia-da light start --p2p.network=--da.grpc.namespace=000008e5f679bf7116cb --
da.grpc.listen=0.0.0.0:26650 --core.ip --gateway environment : - NODE_TYPE=light - P2P_NETWORK= ports : -
"26650:26650" - "26658:26658" - "26659:26659" volumes : - HOME/.celestia-light-:/home/celestia/.celestia-
light-/ healthcheck : test : [ "CMD" , "curl" , "-f" , "http://localhost:26659/header/1" ] interval : 10s timeout : 5s
retries : 5 start_period : 30s Now start the devnet:
```

```
bash make
```

```
devnet-up make
```

```
devnet-up
```

## View the logs of the devnet

If you'd like to view the logs of the devnet, run the following command from the root of the Optimism directory:

```
bash make
```

```
devnet-logs make
```

```
devnet-logs
```

## Stop the devnet

To safely stop the devnet, run the following command:

```
bash make
```

```
devnet-down make
```

```
devnet-down
```

## Clean the devnet

To remove all data from the devnet, run the following command:

```
bash make
```

```
devnet-clean make
```

```
devnet-clean
```

## Deploying to an L1 (or L2)

If you'd like to deploy to an EVM L1 or L2, reference the [OP stack deployment guide](#). [\[ \[ Edit this page on GitHub \]](#) Last updated: [Previous page Deploy an OP Stack devnet](#) [Next page Rollkit](#) [\[](#)