what-youre-going-to-build)

Run OP Node

This guide will walk you through the process of building a node from source, focusing on the op-node and op-geth implementations. These steps are essential if you want to run a node on a specific architecture or inspect the source code of the node you're running.

What you're going to Build

1. Rollup Node (op-node
2.)
3.
    - Responsible for deriving L2 block payloads from L1 data and passing those payloads to the Execution Client.
4.
    - Analogous to a consensus client in Ethereum.
5. Execution Client (op-geth
6.)
7.
    - Executes the block payloads it receives from the Rollup Node.
8.
    - Exposes the standard JSON-RPC API used by Ethereum developers.

Software Dependencies

git ^2 git --version go ^1.22.6 go version node ^20 node --version just ^1.34 just --version foundry ^0.2.0 forge --version make ^4 make --version

Folder Structure

This guide supports running nodes for arbitrary Gelato RaaS chains, both testnets, and mainnets. Feel free to use any folder structure that suits your needs. For the purposes of this documentation, we will use the following structure below: ```

Copy op-rollup-node/ ├── config/ | ├── testnet/ | | ├── genesis.json | | └── rollup.json | ├── mainnet/ | | ├── genesis.json | | └── rollup.json ├── op-geth/ | ├── init-geth.sh | ├── jwt.txt | ├── .env | └── ... ├── optimism/ | ├── jwt.txt | ├── .env | ├── op-node | └── ... └── jwt.txt

```

Create a JWT file

To communicate with op-node and enable the Engine API, you'll also need to generate a JWT secret file and enable Geth's authenticated RPC endpoint.

To generate the JWT secret, run the following:

```

Copy cd /path/to/op-rollup-node openssl rand -hex 32 > jwt.txt

```

Obtaingenesis.json androllup.json
```

1. Log in to your Dashboard.
2. Downloadrollup.json
3. (rollup config) andgenesis.json
4. (genesis config).
5. Place them in yourconfig/testnet
6. in theop-rollup-node
7. directory
8.

?

Build the Rollup Node

- Clone the Optimism Monorepo
- :
- ```
- Copy
- gitclonehttps://github.com/ethereum-optimism/optimism.git
- cdoptimism
- ```
- Check Out the Required Release Tag
- :
- ```
- Copy
- gitcheckout

- **We use the latest official release tagged in the GitHub repo https://github.com/ethereum-optimism/optimism/releases**

- **final command**

- gitcheckoutddc37daa49558c2fb5c1a92e694eeb7de5942e00
- ```
- Build op-node
- :
- ```
- Copy
- makebuild
- ```
-

Build the Execution Client

- Clone op-geth
- :
- ```
- Copy
- gitclonehttps://github.com/ethereum-optimism/op-geth.git
- cdop-geth
- ```
- Check Out the Required Release Tag
- :
- ```
- Copy
- gitcheckout

- **We use the latest official release tagged in the GitHub repo https://github.com/ethereum-optimism/op-geth/releases**

- gitcheckout7c2819836018bfe0ca07c4e4955754834ffad4e0
- ```
- Build op-geth
- :
- ```
- Copy
- makegeth

- ```
-

Runningop-geth

Create a Data directory in yourop-geth

```

Copy cd /path/to/op-rollup-node/op-geth mkdir data-geth

```

Create a .env File in op-geth Directory

```

Copy cd /path/to/op-rollup-node/op-geth nano .env

```

```

Copy GENESIS_FILE_PATH="" SNAPSHOT_FILE_PATH="" GETH_DATADIR="" GETH_AUTHRPC_ADDR="0.0.0.0" GETH_AUTHRPC_PORT="" GETH_AUTHRPC_JWTSECRET="" GETH_AUTHRPC_VHOSTS="" *GETH_METRICS="true" GETH_METRICS_EXPENSIVE="false" GETH_METRICS_ADDR="0.0.0.0" GETH_METRICS_PORT="" GETH_GCMODE="archive" GETH_SYNCMODE="full" GETH_MAXPEERS="0" GETH_NODISCOVER="true" GETH_HTTP="true" GETH_HTTP_ADDR="0.0.0.0" GETH_HTTP_PORT="" GETH_HTTP_VHOSTS="*" GETH_HTTP_CORSDOMAIN="" GETH_HTTP_API="web3,debug,eth,txpool,net,engine" GETH_WS="true" GETH_WS_ADDR="0.0.0.0" GETH_WS_PORT="" GETH_WS_ORIGINS="*" GETH_WS_API="debug,eth,txpool,net,engine" GETH_RPC_ALLOW_UNPROTECTED_TXS="false"

```

Initialize op-geth with the Genesis File

Feel free to customize the base configurations provided in the Optimism documentation to suit your specific requirements. While we will use the recommended configurations for this guide, you can explore and add additional flags as needed. Detailed information about execution layer configurations can be foundhere . Createinit-geth.sh :

```

Copy cd /path/to/op-rollup-node/op-geth nano init-geth.sh

```

```

Copy

# !/bin/bash

# Set environment variables

source.env

# Create data directory if it doesn't exist

mkdir-pDATADIR_PATH

# Initialize geth with the genesis file

echo"Initializing geth with genesis file..." ./build/bin/geth--datadir=DATADIR_PATHinitGENESIS_PATH
```

Generate JWT secret if it doesn't exist

if[!-f"JWT_SECRET_PATH"];then echo"Generating JWT secret..." opensslrand-hex32>JWT_SECRET_PATH fi

```

Run the geth Node

```

Copy cd/path/to/op-rollup-node/op-geth

Load environment variables from the .env file in the root directory

source.env

Initialize geth

./init-geth.sh

Run the geth command with the environment variables

./build/bin/geth

```

Testing the Running Geth Instance

After starting your geth instance, you can use the following bash script to test if geth is running and to confirm the chain ID:

```

Copy curl-XPOSThttp://127.0.0.1:8545\ -H"Content-Type: application/json"\ --data'{"method":"eth_chainId","params":[],"id":1,"jsonrpc":"2.0"}'

You should see a response similar to this

{ "jsonrpc":"2.0", "id":1, "result":"0x2a88"##This is your chain id in hex, in this case its 10888. }

```

Runningop-node

We will utilize the base configurations provided in the Optimism documentation for the consensus layer. However, you can adjust and expand these configurations to fit your specific requirements. For a comprehensive understanding of all available configurations, refer to the detailed documentation on consensus layer configurationshere . Create a .env File in optimism Directory

```

Copy OP_NODE_L1_ETH_RPC="" OP_NODE_L1_RPC_KIND="standard" OP_NODE_L2_ENGINE_AUTH="" OP_NODE_ROLLUP_LOAD_PROTOCOL_VERSIONS="true" OP_NODE_ROLLUP_HALT="major" OP_NODE_ROLLUP_CONFIG="" OP_NODE_SEQUENCER_ENABLED="false" OP_NODE_SEQUENCER_L1_CONFS="5" OP_NODE_VERIFIER_L1_CONFS="4" OP_NODE_LOG_FORMAT="json" OP_NODE_LOG_LEVEL="info" OP_NODE_P2P_DISABLE="false" OP_NODE_P2P_LISTEN_IP="0.0.0.0" OP_NODE_P2P_LISTEN_TCP_PORT="" OP_NODE_P2P_LISTEN_UDP_PORT="" OP_NODE_P2P_PEER_SCORING="none" OP_NODE_P2P_PEER_BANNING="false" OP_NODE_P2P_PEER_BANNING_DURATION="0h1m0s" OP_NODE_P2P_BOOTNODES="" OP_NODE_P2P_ADVERTISE_TCP="" OP_NODE_P2P_ADVERTISE_UDP="" OP_NODE_P2P_ADVERTISE_IP="" OP_NODE_P2P_SYNC_REQ_RESP="true" OP_NODE_P2P_STATIC="" OP_NODE_P2P_PRIV_RAW="" OP_NODE_RPC_ADDR="0.0.0.0" OP_NODE_RPC_PORT="" OP_NODE_RPC_ENABLE_ADMIN="true" OP_NODE_SNAPSHOT_LOG="" OP_NODE_METRICS_ENABLED="true" OP_NODE_METRICS_ADDR="0.0.0.0" OP_NODE_METRICS_PORT="" OP_NODE_PPROF_ENABLED="true" OP_NODE_L1_BEACON=""
```
```

OP_NODE_L2_ENGINE_RPC=""

```

Ensure thatop-node P2P ports ( and ) are accessible externally via. This allows the node to communicate with other peers on the network.

Add the sequencer node's multiaddr to in your configuration. This helps establish a direct connection with the sequencer, ensuring smooth operation and synchronization.

Run the op-node

```

Copy cd/path/to/op-rollup-node/optimism

Load environment variables from the .env file in the root directory

source.env

Run the op-node command with the environment variables

./op-node/bin/op-node

```

Previous OP Stack Next Polygon CDK Last updated14 days ago
```