# Node Base Configuration

⚠ Always runop-node andop-geth in a one-to-one configuration. Don't run multipleop-geth instances behind oneop-node , or vice versa. To configure your node, you will need to do the following:

1. Configureop-node
2. to point to the correct L1,op-geth
3. , and L2 network.
4. Initializeop-geth
5. with the correct network parameters.
6. Configureop-geth
7. to properly communicate with the Rollup Node.
8. Optionally, configure Legacy Geth.

## Configuringop-geth

Although the Docker image for the Execution Engine is calledop-geth , the actual binary is still calledgeth in order to minimize differences betweenop-geth andgo-ethereum . You can see the differencehere(opens in a new tab) . op-geth stores its state in a database that requires initialization. Depending on the network you're running, initialization is done one of three ways:

1. With Network Flags:
2. This initializes the genesis information and chain configuration from thesuperchain-registry(opens in a new tab)
3. .
4. With a Genesis File:
5. This is for deployments that are not migrated from a legacy network (i.e. OP Sepolia). In this case, you'll use a genesis file and initialize the data directory viageth init
6. .
7. With a Data Directory:
8. This is used for networks that are migrated from a legacy network. This currentlyonly
9. includes OP Mainnet. In this case, you'll download a preconfigured data directory and extract it. No further initialization is necessary in this case, because the data directory contains the network's genesis information. This method can be bypassed if you utilizesnap sync
10. .

Regardless of howop-geth is initialized, you'll need to ensure that you have sufficient disk space available to store the network's data. As of this writing, the OP Mainnet data directory is ~1.6TB for a full node and ~5TB for an archival node.

### Initializeop-geth

Instructions for each initialization method are below. If you're spinning up an OP Mainnet, use theInitialization via Data Directory path. If you're spinning up an OP Sepolia node, use theInitialization via Network Flags path.

#### Initialization via Network Flags

To initializeop-geth with the network flags, you simply need to set the--op-network=and--network= onop-node . To see the latest support networks, you can consult the--help output for theop-network option.

#### Initialization via Genesis File

op-geth uses JSON files to encode a network's genesis information. For networks that are initialized in this way, you'll receive a URL to the genesis JSON. You'll need to download the genesis JSON, then run the following command to initialize the data directory:

# !/bin/sh

# FILE

/ DATADIR /genesis.json OP_GETH_GENESIS_URL =<< insert op-geth url to the genesis file

if [ ! -s FILE ]; then apk add curl curl OP_GETH_GENESIS_URL -o FILE geth init --datadir /db FILE else echo "Genesis file already exists. Skipping initialization." fi

#### Initialization via Data Directory

To initializeop-geth with a preconfigured data directory, simply download and extract the data directory to a place of your choosing. The data directory is exported as a tar file. An example command to do this is below:

curl

-o

< path

to

data

director y

-sL

< URL

to

data

director y

tar

-xvf

< path

to

data

director y

## Configuration

Onceop-geth is initialized, it can be configured via CLI flags.op-geth accepts all thestandardgo-ethereum flags(opens in a new tab) as well as a few extra flags that are specific to Optimism. These flags are:

- --rollup.historicalrpc
- : Enables the historical RPC endpoint. This endpoint is used to fetch historical execution data from Legacy Geth. This flag is only necessary for upgraded networks.
- --rollup.sequencerhttp
- : HTTP endpoint of the sequencer.op-geth
- will routeeth_sendRawTransaction
- calls to this URL. Bedrock does not currently have a public mempool, so this is required if you want your node to support transaction submission. Consult the documentation for the network you are participating in to get the correct URL.
- --rollup.disabletxpoolgossip
- : Disables transaction pool gossiping. While not required, it's useful to set this totrue
- since transaction pool gossip is currently unsupported.

To communicate withop-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:

openssl

rand

-hex

32

jwt.txt Then, specify the following flags:

- --authrpc.addr
- : Sets the addressop-geth

- 's authenticated RPC should listen on.
- --authrpc.port
- : Sets the portop-geth
- 's authenticated RPC should listen on. The default value is8551
- .
- --authrpc.jwtsecret
- : Sets the path to a JWT secret file you generated above.

### Recommended Flags forop-geth

Configuration

You may also want to specify the following flags based on your configuration:

- --authrpc.vhosts
- : Whitelists which hosts (as defined in theHost
- header) are allowed to access the authenticated RPC endpoint. This is useful if you're runningop-geth
- on containerized infrastructure. The default value islocalhost
- .
- --http.vhosts
- : Whitelists which hosts (as defined in theHost
- header) are allowed to access the unauthenticated RPC endpoint. This is useful if you're runningop-geth
- on containerized infrastructure. The default value islocalhost
- .
- --http
- ,--http.addr
- , and--http.port
- : Enables the unauthenticated RPC endpoint, configures its address, and configures its port. You'll almost certainly want to specify these, since they will enable Geth's JSON-RPC endpoint.
- --ws
- ,--ws.addr
- , and--ws.port
- : Enables the WebSocket API.
- --verbosity
- : Configures Geth's log level. This is a number between 0 and 5, with 5 being the most verbose. Defaults to 3.

### Working Base Configuration

A valid command that runsop-geth and enables RPC over HTTP and WebSockets looks like:

geth \ --ws \ --ws.port=8546 \ --ws.addr=localhost \ --ws.origins= "" \ *--http* \ *--http.port=8545* \ *--http.addr=localhost* \ *--http.vhosts= ""* \ --http.corsdomain= "" \ *--authrpc.addr=localhost* \ *--authrpc.jwtsecret=/var/secrets/jwt.txt* \ *--authrpc.port=8551* \ *--authrpc.vhosts= ""* \ --datadir=/data \ --verbosity=3 \ --rollup.disabletxpoolgossip=true \ --rollup.sequencerhttp=https://mainnet-sequencer.optimism.io/ \ --op-network=op-mainnet ConsultGeth's documentation(opens in a new tab) for more information on customizingop-geth 's behavior.

# Configuringop-node

op-node is a standalone, statically linked binary. It stores no state, and requires no initialization. It consumes configuration parameters either via the command line or environment variables. For some networks, the Rollup Node also requires a configuration file (calledrollup.json or the "rollup config") that configures network-specific genesis parameters. For official networks like OP Sepolia and OP Mainnet, the genesis config is hardcoded in theop-node software and can be specified via a--network flag.

Following theEcotone upgrade node operators must set an L1 beacon value to retrieveblobs from a Beacon node.

⚠ Theop-node RPC should not be exposed publicly. If left exposed, it could accidentally expose admin controls to the public internet.

### Working Base Configuration

A minimal valid configuration that runsop-node looks like:

op-node

--l1= < ethereum

mainnet

RPC

ur l

\ --l2= < op-geth

authenticated

RPC

ur l

\ --network=op-mainnet \ --rpc.addr=127.0.0.1 \ --rpc.port=9545 \ --l2.jwt-secret= < path

to

JWT

secre t

\ --l1.beacon= < http

endpoint

address

of

L1

Beacon-nod e

\ --syncmode=execution-layer You can manually specify a path to a rollup config with the--rollup.config flag. This is used for testnets or internal deployments that are not migrated from a legacy network.

Each of the above flags can also be defined via an environment variable. Runop-node --help to see a list of all available flags and environment variables.

### Configuring Peer-to-Peer Networking

Unlike the previous system, theop-node participates in a peer-to-peer network. This network is used to distribute blocks that have not been submitted to L1 yet. Theop-node will automatically discover and connect to peers using a hardcoded set of bootnodes. You can also manually specify peers to connect to via the--p2p.static flag.

For best results, runop-node with a static IP address that is accessible from the public Internet. For Kubernetes deployments, this can be achieved by configuring a dedicatedIngress with an external IP, and using the--p2p.advertise.ip flag to specify the IP address of the load balancer when advertising IP addresses to peers.

The default port for the peer-to-peer network is9003 . You will need to open this port on your firewall to receive unsubmitted blocks. For your node to be discoverable, this port must be accessible via both TCP and UDP protocols.

## Legacy Geth

If you are running a node for an upgraded network like OP Mainnet (but not OP Sepolia), you will also need to run Legacy Geth in order to serve historical execution traces. Fundamentally, Legacy Geth is our oldl2geth binary running against a preconfigured data directory. To configure Legacy Geth, follow the instructions above for using a preconfigured data directory, then execute the following command:

It is imperative that you specify theUSING_OVM=true environment variable in the command below. Failing to specify this will causel2geth to return invalid execution traces, or panic at startup. USING_OVM = true \ ETH1_SYNC_SERVICE_ENABLE = false \ RPC_API = eth,rollup,net,web3,debug \ RPC_ADDR = 0.0 .0.0 \ RPC_CORS_DOMAIN = * \ RPC_ENABLE = true \ RPC_PORT = 8545 \ RPC_VHOSTS = * \ geth

--datadir

< path

to

data

director y

This command is the minimum required to run Legacy Geth and expose a functioning RPC endpoint. As before,l2geth takes all standardgo-ethereum flags so you can customize the configuration as needed.

As mentioned above, don't forget to specify--rollup.historicalrpc onop-geth to properly route requests for historical execution to Legacy Geth.

Since Legacy Geth is read-only, it is safe to run multiple Legacy Geth nodes behind a load balancer.

### Historical Execution vs. Historical Data Routing

Only requests for historical execution will be routed to Legacy Geth. Everything else will be served byop-geth directly. The termhistorical execution refers to RPC methods that need to execute transactions prior to bedrock (not just read data from the database):

- eth_call
- eth_estimateGas
- debug_traceBlockByNumber
- debug_traceBlockByHash
- debug_traceCall
- debug_traceTransaction

If you do not need these RPC methods for historical data, then you do not need to run Legacy Geth at all.

## Next Steps

- See theop-node configuration
- guide for additional configuration options forop-node
- and the Consensus-Layer.
- Similarly, visit theop-geth configuration
- guide for additional configuration options forop-geth
- and Execution-Layer.
- If you run into any problems, please reach out to oudeveloper support forum(opens in a new tab)
- for help.

Running OP Sepolia from Source Consensus Layer Configuration