

Rollup Operations

This guide reviews the basics of rollup operations, such as how to start your rollup, stop your rollup, and add nodes.

Stopping Your Rollup

An orderly shutdown is done in the reverse order to the order in which components were started:

To stop the batcher, use this command:

```
curl
-d
'{"id":0,"jsonrpc":"2.0","method":"admin_stopBatcher","params":[]}' -H
"Content-Type: application/json"
http://localhost:8548
|
```

jq This way the batcher knows to save any data it has cached to L1. Wait until you seeBatch Submitter stopped in batcher's output before you stop the process.

Stopop-node

This component is stateless, so you can just stop the process.

Stopop-geth

Make sure you useCTRL-C to avoid database corruption.

Starting Your Rollup

To restart the blockchain, use the same order of components you did when you initialized it.

Startup-geth

Startup-node

Startup-batcher

Ifop-batcher is still running and you just stopped it using RPC, you can start it with this command:

```
curl
-d
'{"id":0,"jsonrpc":"2.0","method":"admin_startBatcher","params":[]}' -H
"Content-Type: application/json"
http://localhost:8548
|
```

jq Synchronization takes time

op-batcher might have warning messages similar to:

```
WARN [03-21|14:13:55.248] Error calculating L2 block range err="failed to get sync status: Post \"http://localhost:8547\": context deadline exceeded"
WARN [03-21|14:13:57.328] Error calculating L2 block range err="failed to get sync status: Post \"http://localhost:8547\": context deadline exceeded"
This means thatop-node is not yet synchronized up to the present time.
Just wait until it is.
```

Adding Nodes

To add nodes to the rollup, you need to initialize `op-node` and `op-geth`, similar to what you did for the first node. You should not add an `op-batcher` because there should be only one.

Configure the OS and prerequisites as you did for the first node

Build the Optimism monorepo and `op-geth`

as you did for the first node

Copy from the first node these files:

```
~/op-geth/genesis.json ~ /optimism/op-node/rollup.json
```

Create a new `jwt.txt`

file as a shared secret:

```
cd
```

```
~/op-geth openssl
```

```
rand
```

```
-hex
```

```
32
```

```
jwt.txt cp
```

```
jwt.txt
```

```
~/optimism/op-node
```

Initialize the new `op-geth`:

```
cd
```

```
~/op-geth ./build/bin/geth
```

```
init
```

```
--datadir=./datadir
```

```
./genesis.json
```

Turn on peer to peer synchronization to enable L2 nodes to synchronize directly

If you do it this way, you won't have to wait until the transactions are written to L1. If you already have peer to peer synchronization, add the new node to the `--p2p.static` list so it can synchronize.

Startup `geth`

(using the same command line you used on the initial node)

Startup `node`

(using the same command line you used on the initial node)

[Using Snap Sync Key Management](#)