

# proxyd

proxyd is an important RPC request router and proxy used within the OP Stack infrastructure. It enables operators to efficiently route and manage RPC requests across multiple backend services, ensuring performance, fault tolerance, and security.

## Key features

- RPC method whitelisting
- Backend request routing
- Automatic retries for failed backend requests
- Consensus tracking (latest, safe, and finalized blocks)
- Request/response rewriting to enforce consensus
- Load balancing across backend services
- Caching of immutable responses
- Metrics for request latency, error rates, and backend health

## How it works

To start using proxyd , follow these steps:

### Build the binary

:

- Run the following command to build the proxyd binary:
- `make`
- `proxyd`
- This will build the proxyd binary. No additional dependencies are required.

### Configure proxyd

:

- Create a configuration file to define your proxy backends and routing rules.
- Refer to [example.config.toml \(opens in a new tab\)](#)
- for a full list of options with commentary.

### Start the service

:

Once the configuration file is ready, start the proxyd service using the following command:

```
proxyd
```

```
< path-to-config.toml
```

## Consensus awareness

Version 4.0.0 and later include consensus awareness to minimize chain reorganizations.

Set `consensus_aware` to `true` in the configuration to enable:

- Polling backends for consensus data (latest block, safe block, peer count, etc.).
- Resolving consensus groups based on healthiest backends
- Enforcing consensus state across client requests

## Caching and metrics

### Cacheable methods

Certain immutable methods, such as `eth_chainId` and `eth_getBlockByHash` , can be cached using Redis to optimize performance.

## Metrics

Extensive metrics are available to monitor request latency, error rates, backend health, and more. These can be configured via `metrics.port` and `metrics.host` in the configuration file.

## Next steps

- Read about the [OP Stack chain architecture](#)
- .
- Find out how you can support [snap sync](#)
- .
- on your chain.
- Find out how you can utilize [blob space](#)
- to reduce the transaction fee cost on your chain.

[proxyd Intro to OP Stack hacks](#)