
chaind Documentation

Release 0.0.1

Kyokan, LLC

Dec 11, 2018

Contents:

1	Getting started	3
2	History of chaind	5
2.1	Installation	5
2.2	Configuration	6

Building reliable blockchain infrastructure is a notoriously hard problem. As a result, most blockchain companies don't run their own infrastructure, instead choosing to outsource it to a third-party. `chaind` makes deploying your own infrastructure simpler and cheaper by acting as a proxy between the Internet and your blockchain nodes. Out of the box, `chaind` provides the following features:

- Automatic failover to any blockchain node with an open RPC endpoint
- Intelligent request caching that takes chain reorgs into account
- RPC-aware request logging

CHAPTER 1

Getting started

The simplest way to get started with `chaind` is to install one of our pre-built packages using your OS's package manager. For more information, head on over to [Installation and Configuration](#).

History of `chaind`

`chaind` began in late 2018 as an internal project at Kyokan. Since then, `chaind` has served millions of RPC request on low-cost hardware.

2.1 Installation

2.1.1 From Package

The fastest way to install `chaind` is to install it via your OS's package manager. We currently have Debian packages available, with RPMs coming soon. To install from our Debian repository, follow these steps:

1. Add Kyokan's signing key

First, you'll need to instruct your package manager to trust Kyokan's PGP key. Our PGP key is stored on MIT's keyserver, and has the fingerprint `A27E D5CE 49FA EB7D E0AA 50D4 23D7 FC00 2105 25`.

You'll only need to perform this step once per machine. Run the command below to use our key:

```
sudo apt-key adv --keyserver pgp.mit.edu --recv-keys 21052518
```

You should receive output that looks like this:

```
Executing: /tmp/apt-key-gpghome.MZu04hoFiT/gpg.1.sh --keyserver pgp.mit.edu --recv-
↳ keys 21052518
gpg: key 23D7FC0021052518: public key "Kyokan, LLC OSS Signing Key <mslipper@kyokan.
↳ io>" imported
gpg: Total number processed: 1
gpg:             imported: 1
```

2. Add Kyokan's Debian repository to your `sources.list`

Run the following command to add Kyokan's Debian repository to your `sources.list` file:

```
echo "deb https://dl.bintray.com/kyokan/oss-deb any main" | sudo tee -a /etc/apt/  
↪sources.list
```

This also only needs to be done once per machine.

3. Install chaind

Now you're ready to install chaind. To do so, run these commands:

```
sudo apt-get update  
sudo apt-get install chaind
```

2.1.2 From Source

To build and install chaind from source, you'll need to first install the following prerequisites:

1. go version 1.10 or higher
2. make
3. dep
4. git

Now, get the chaind source by running `go get -u github.com/kyokan/chaind` and `cd` into the source directory. You can now build and install chaind with the following commands:

```
make deps  
make build  
make install-global
```

`make install-global` will place the chaind binary in `/usr/bin`. You'll need to create a configuration file manually.

2.1.3 Next Steps

Now, you're ready to configure your chaind instance. See our Configuration page for more information.

2.2 Configuration

2.2.1 Configuration file

chaind is configured via a single `chaind.toml` file. When you install chaind using a package manager, a default `chaind.toml` is placed in `/etc/chaind`. It looks like this:

```
eth_path = "eth"  
rpc_port = 8080  
log_level = "info"  
  
[log_auditor]  
log_file="/var/log/chaind_audit.log"  
  
[redis]  
url="localhost:6379"
```

(continues on next page)

(continued from previous page)

```
[[backend]]
type="ETH"
url="https://mainnet.infura.io/"
name="infura"

[[backend]]
type="ETH"
url="http://localhost:8545/"
name="local"
main=true
```

The only parts of the config file you'll need to change are the `[[backend]]` stanzas, since the default URLs are only examples and won't work out of the box. These stanzas define which blockchain nodes `chaind` will be proxying to. There can be an unlimited number of backends. Below, see a description of each available backend configuration directive:

2.2.2 Backend configuration

Key	Description
<code>type</code>	The type of blockchain node. Currently, can only be <code>ETH</code> , however in the future <code>BTC</code> (and potentially others) will be supported.
<code>url</code>	The URL to the blockchain node. Can be <code>http</code> or <code>https</code> .
<code>name</code>	A name for the backend. Will appear in logs.
<code>main</code>	Optional. Defines whether or not <code>chaind</code> should proxy to this node by default. There can only be one main backend per <code>type</code> . If <code>main</code> isn't specified, the first backend will be chosen as the main.

2.2.3 Server configuration

The following directives are used to configure `chaind` itself:

Key	Description
<code>rpc_port</code>	The port at which to listen for RPC requests.
<code>log_level</code>	<code>chaind</code> 's log level. Can be one of the following: <code>trace</code> , <code>debug</code> , <code>info</code> , <code>warn</code> , <code>error</code> , <code>crit</code> .
<code>[log_auditor].log_file</code>	The location of <code>chaind</code> 's audit log file
<code>[redis].url</code>	URL to an instance of Redis.