Linkerd 1.x Runbook

Version 2018-08-23

Copyright © 2018 Buoyant, Inc. All rights reserved. While Linkerd is open source, this document is a commercial publication. No part of this document may be reproduced or transmitted in any form or by any means (including but not limited to electronic, mechanical, photocopying, recording, or otherwise) without prior written permission of Buoyant, Inc.

Please see the important Disclaimer below.

Table of Contents

Introduction
- What is this document?
- About Buoyant
- Join the community!
- Getting help
- Where do I get Linkerd?
- Versioning
- Experimental and deprecated features

Going to production
- A basic preflight checklist
- Preparing your environment
  - For non-SSD hosts, enable in-memory filesystem for temporary files
  - Provide sufficient system resources for Linkerd
  - Validate your OS and network configuration
- Configuring Linkerd for production environments
  - Tune the JVM's heap size
  - Use a recent JVM
  - Enable GC Logging
  - Tune the number of Finagle workers
  - Tune the bind cache sizes
  - Ensure metrics and tracing sample rates are low
  - Lock down Linkerd's admin UI
  - Disable insecure features at the edge

        Consume Linkerd's metrics
    Monitoring Linkerd's Health
        Service Cache
        Client Cache
        Heap Usage
        File Descriptors
        Connection Reuse

Monitoring Service Health
    Request Volume
    Success Rate
        com.twitter.finagle.service.ResponseClassificationSyntheticException
        com.twitter.finagle.NoBrokersAvailableException
        com.twitter.finagle.ConnectionFailedException
        com.twitter.finagle.ChannelClosedException
        com.twitter.finagle.CancelledRequestException
    Retries
    Latency
    Circuit Breakers
    Connection Reuse
    HTTP Response Codes

Understanding Linkerd's Behavior
    Debugging routing issues
    Debugging service discovery issues

Improving Linkerd's Performance
    Measure performance after warmup requests
    Do not use ApacheBench (ab) or HTTP/1.0 load generators
    Improving Request Throughput
    Improving Tail Latency

Disclaimer

Appendix A: Host, OS, and Network Settings
    Host
    CPU
    Network
    Memory
    DNS
    Clock Skew in Cloud Environments

Appendix B: Which loadbalancer should I use?

Appendix C: Namerd

Copyright © 2018 Buoyant, Inc.

Appendix D: Mesos and DC/OS

Appendix E: Recent Linkerd releases
    Linkerd 1.4.6 (August 13, 2018)

Introduction

What is this document?

Welcome to Buoyant's Linkerd 1.x Runbook! This goal of this doc is to provide concrete, practical advice and information for running Linkerd in production environments. We, the authors and core maintainers of Linkerd, have written this doc based on our experience over the past few years helping companies deploy and operate Linkerd in mission-critical roles.

This is not a configuration guide for Linkerd, nor a deployment guide, nor a contributor's guide. For those topics, please refer to the docs on linkerd.io. Instead, this document is designed to help the operator of Linkerd safely deploy Linkerd into production, and keep it there. If you're on call for Linkerd, this doc is for you!

We think of this as a "living document." Linkerd moves fast. We'll update this doc for every release, and we'll track what we've changed in Appendix E. If you just want to catch up on what's new, skip right there.

About Buoyant

Founded by William Morgan and Oliver Gould in 2015, Buoyant is the original creator of Linkerd (now a CNCF project). Buoyant's mission is to create a world where people can trust the technology they rely on, by giving everyone the power to build innovative, reliable, and secure software.

Today, Buoyant is both the primary sponsor of and contributor to Linkerd, as well as the primary provider of commercial support.

Join the community!

If you're serious about operating Linkerd in production, you should join the open source community channels around it. This is important both for staying aware of important updates and announcements, and also for learning from other users who are doing the same thing.

At a minimum, we recommend you join:

- The linkerd-announce mailing list
- The linkerd-users mailing list

You can also join:

- The linkerd-dev mailing list
- The Linkerd Discourse forums

Copyright © 2018 Buoyant, Inc.

- The Linkerd Slack

Linkerd development happens on GitHub. That's also the best place to submit bug reports and pull requests. Please also star the repo to inflate our vanity metric. :)

Getting help

If you need help with Linkerd, you have a couple of open source channels available to you. Our recommendations are:

- One-off question? Start with the Slack room.
- Complex troubleshooting? Spin up a thread on the Discourse forums.
- Bug report? File a GitHub issue, please.

Of course, open source support is provided by the community on a best-effort basis. And don't forget to help others—this is often the best way to give back!

Where do I get Linkerd?

Linkerd's code is 100% open source and is hosted entirely in the GitHub repo. You may choose to build your own binaries or images from this code.

For every release, Buoyant also creates Linkerd binaries in the Linkerd GitHub project release section, and Docker images, hosted on DockerHub. We consider these the canonical binaries and images for Linkerd.

Versioning

Linkerd releases follows a modified form of semantic versioning. Linkerd version numbers are of the form <**major**>.<**minor**>.<**patch**>. Breaking changes (typically, configuration incompatibilities) are denoted by changes in **minor** version number. Non-breaking changes are denoted by changes in **patch** version number. (Changes in major version number are reserved for significant project changes.)

For example:

- 1.3.6 -> 1.3.7: no breaking changes
- 1.3.7 -> 1.4.0: breaking changes to configuration
- 2.0 -> major rewrite

(Note that Linkerd is currently developed in two parallel lines: 1.x and 2.x. This document is specific to the Linkerd 1.x line of development.)

Experimental and deprecated features

Occasionally, we denote features as *experimental*. This means that, while we feel confident in the viability of the the feature, it hasn't seen enough production use for us to recommend it unreservedly. (We tend to err on the side of being conservative.)

Copyright © 2018 Buoyant, Inc.