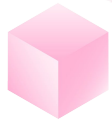


DEPLOYMENT GUIDE

1 / Introduction	3
2 / LogDNA Architecture	4
3 / Deploying LogDNA	6
4 / Getting Started with LogDNA	10
5 / Next Steps	13



1/ Introduction

Introduction

This guide is designed for IT System administrators to walk you through the process of deploying LogDNA to your environment. This guide is not intended for end users and does not go into depth about how to use the application. At the end of this guide, you will have a fully functional LogDNA instance capable of ingesting, centralizing and storing logs from your applications, services and platforms.

What is LogDNA?

LogDNA is highly scalable log management solution that indexes, aggregates, and analyzes log data. It supports logs originating from any source including applications, cloud systems, orchestration platforms, bare metal servers, and more. Its modular architecture allows it to grow to handle any log volume, whether you're logging 100 GB of data or 100 TB per day.

LogDNA also provides several tools to help you manage your logs. The fully featured web UI provides a streamlined interface for live-tailing, searching, graphing, and alerting on log data. You have complete control over your logs including the ability to create custom parsing rules, export logs for long-term storage, and more.

How LogDNA Provides Observability

Observability is the measure of how well a system is working based on its outputs. It allows you to infer the internal state of a system—such as its performance and overall stability—without directly monitoring each and every behavior that takes place. Logs provide crucial insights into the health of your applications and infrastructure. Logs contain detailed contextual information about events that have occurred, and how those events impact other services. This includes normal operational events, unexpected events, and errors.

Compared to metrics and traces, logs are much more flexible in the information that they can provide. Logs are customizable and capable of storing almost any form of data, from numeric measurements to error messages. In addition, logs provide an immutable record of activity over time, making it easier to troubleshoot and audit your systems. Organizations that leverage logs have a significant advantage in their ability to quickly identify, respond to, and resolve problems in their infrastructure.



2/ Overview of the LogDNA Architecture

LogDNA uses a microservice-based architecture to split different tasks into discrete and scalable units. These microservices can be organized into two roles: log ingestion, and log retrieval.

Logs sent from your log sources to LogDNA are received by one of many ingestion endpoints. These endpoints route each log to a message queue using a proprietary message bus. The message queues manage the delivery of logs to a number of worker pools, which process logs for use in various LogDNA services such as parsing, indexing, live tail, graphs, and alerts. Other microservices provide features such as hosting API endpoints, providing security and authentication, and maintaining log data stores.





2/ Overview of the LogDNA Architecture

Ingestion Services

Ingestion services provide the backend components necessary to ingest, parse, index, and store logs. They receive logs sent by log generating components, as well as fulfill requests for log data sent from other LogDNA services.

Logs arrive into the LogDNA infrastructure through ingestion endpoints, which route logs from your applications and systems to a proprietary message brokering service called Buzzsaw. Buzzsaw is a highly optimized and highly scalable brokering service designed specifically for log data.

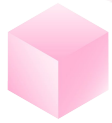
Buzzsaw routes logs to worker pools, which consist of microservices performing specific actions such as parsing, generating graphs, and running alerts. It also sends logs to a cluster of Elasticsearch nodes, which provide LogDNA's search and filtering functionality. Buzzsaw acts as a buffer between ingestion sources, worker pools, and Elasticsearch nodes, ensuring that performance problems in any one service doesn't impact the performance of other services.

Retrieval Services

Retrieval services provide the ability to access, view, and export log data. They act as the gateway between users and the LogDNA ingestion services and are responsible for actions such as responding to user requests, querying Elasticsearch for saved log data, and live tailing logs.

The primary services provided by retrieval services are the user interfaces, which include the LogDNA web UI and REST API. Actions performed in the LogDNA web application, command line interface (CLI), or REST API are received by these services and proxied to the relevant infrastructure services before returning the results to the user. These services also perform user authentication and enforce access controls.

Other retrieval services include features that export log data automatically, such as alerts and archives.



3/ Deploying LogDNA

LogDNA has a very flexible deployment model that supports both cloud and on-premise deployments. There are two models for using LogDNA:

- **Software as a Service (SaaS):** LogDNA runs on a cloud infrastructure. This offers the greatest convenience for teams wishing to quickly get started with a scalable logging solution.
- **Self-hosted:** LogDNA runs on your own infrastructure, whether in a cloud environment or on-premise. This offers the greatest control for teams that have strict security, regulatory, or operational requirements.





3/ Deploying on SaaS Infrastructure

Software as a Service (SaaS)

The LogDNA SaaS is the simplest method of deploying LogDNA. In this model, your logs are sent to an enterprise-grade cloud infrastructure managed by LogDNA. You get all the benefits of a SaaS service including high availability, infinite scalability, and zero maintenance.

Single vs. Multi-Tenant Installations

The LogDNA cloud platform uses a multi-tenant model. This means that all customers share the same LogDNA cloud platform resources. LogDNA has several controls in place to prevent unauthorized access to log data including encryption in transit and at rest and strict access controls. LogDNA also offers a single-tenant model, which provides dedicated managed cloud resources to customers requiring greater security or larger ingestion volume. The LogDNA cloud platform is fully compliant with HIPAA, SOC 2, Privacy Shield, GDPR, and PCI DSS.

To learn more about single or multi-tenant SaaS, please [contact the LogDNA team](#).

Prerequisites

Before you begin using LogDNA SaaS, please meet the following prerequisites:

- Create a LogDNA account at <https://logdna.com/sign-up/>. You can create a new LogDNA username and password, or sign in using your Google, GitHub, or Heroku account.
 - Note that this will create a multi-tenant account. If you wish to create a single-tenant account, contact the [LogDNA sales team](#).
- If you have a firewall or other security device in place, allow outbound traffic to <https://api.logdna.com>.

Once your account is created, you can sign into the LogDNA web application by opening your browser to <https://app.logdna.com>.



3/ Deploying LogDNA on Self-Hosted Infrastructure

Self-Hosted Deployments

LogDNA supports self-hosted deployments for organizations wishing to deploy an in-house log management solution. This gives you complete control over your logging infrastructure and data. The process requires no dedicated engineers or consultants and your users can immediately begin viewing log data without extensive training or onboarding.

LogDNA can run on any infrastructure, provided that it can be deployed with Kubernetes. Kubernetes is easy to install, set up and get started.

This section explains the requirements and processes for deploying LogDNA to either a private cloud environment, or to a bare metal environment.

Requirements

Whether you are deploying to a private cloud or to bare metal, your infrastructure must meet the following minimum requirements.

You must have a Kubernetes cluster in order to deploy LogDNA. These requirements are designed for ingesting 250 GB of log data per day with 30 days of retention.

LogDNA requires hot and cold worker nodes.

Each hot worker node in your Kubernetes cluster should meet the following requirements:

- Linux distribution
- 128 GB of RAM
- 48 vCPUs
- SSD or NVME-based storage with 2,000 MB/s throughput
- 10 GB/s network bandwidth

LogDNA recommends deploying at least 3 hot nodes with 1 TB of NVMe storage, and 3 cold nodes with 8 TB of storage. **These requirements may vary based on your specific needs.** For example, if you require a longer retention period, you will need to increase the size of your storage volumes. If you ingest a significant volume of log data per day, you will need to add CPU, RAM, and storage capacity of your worker nodes, or add additional nodes.



3/ Deploying LogDNA on Self-Hosted Infrastructure

A) Private Cloud

Deploying LogDNA to your private cloud lets you leverage the scalability of the cloud for better performance, while also keeping your log data under your control. Since LogDNA runs on Kubernetes, it can be deployed to most major cloud Kubernetes providers including:

- IBM Cloud™ Kubernetes Service (IKS)
- Google Kubernetes Engine (GKS)
- Azure Kubernetes Service (AKS)
- Amazon Elastic Container Service for Kubernetes (EKS)

Your cluster must use a network-based storage solution supporting at least 2,000 MB/s of throughput. In addition, we recommend enabling cluster autoscaling if your cloud provider supports it. LogDNA automatically scales based on log volume, and enabling cluster autoscaling ensures that an increase in log volume doesn't lead to poor performance.

To ensure that LogDNA is deployed successfully to your private cloud, LogDNA will manage the deployment for you. This will require us to access your cloud environment and manage resources in your Kubernetes cluster.

If you have questions about this process, speak to your sales rep or [contact our sales team](#) for more information.

B) On Premise Deployments

Deploying LogDNA to a self-hosted or on-premise environment gives you the greatest degree of control over your data and log management services.

As long as Kubernetes can be installed on your infrastructure, LogDNA can reside to most bare metal infrastructure.

The process for doing so will vary depending on your environment's topology. Here are some considerations for your LogDNA on premise deployment:

- Infrastructure and Data Collection Topology
- Sizing Needs based on retention and predicted volume and traffic patterns
- Logging Sources
- Access Control
- Interdependencies with other teams

[Contact us](#) to speak with an expert who will help you design a solution that best fits your infrastructure.



4/ Getting Started with LogDNA

Once your LogDNA deployment is up and running, you may want to finalize the installation by taking the following steps.

Ingestion Methods

LogDNA supports a number of options for shipping logs from your applications and systems to LogDNA. There are five in total:

1. LogDNA Agent
2. Code Libraries
3. Platforms
4. Syslog
5. REST API

1. LogDNA Agent

The LogDNA agent is the recommended method of sending logs to LogDNA. It installs a service onto a host machine that continuously monitors the machine's log files and sends new logs to LogDNA. It automatically detects common log directories—such as `/var/log` on Linux and Mac and `C:\ProgramData\logs` on Windows—and supports any number of additional directories or files.

The agent is open source, self-updating, and automatically reconnects to LogDNA in case of a connection error.

The agent can also be deployed as a container using Docker or Kubernetes. To see the agent source code, check out the [GitHub repo](#).

To install the host agent, log in to the LogDNA web app and follow the instructions on the [Add a Log Source](#) screen. You can learn more about the agent by reading the [agent documentation](#).

2. Code Libraries

Code libraries let you ship logs directly from your applications to LogDNA. This is useful in environments where you don't have the ability to install the LogDNA agent, such as serverless platforms. LogDNA provides official libraries for Node.JS, Python, and Ruby, and you can find community-contributed libraries for Go, Java, .NET, PHP, and iOS (Objective-C and Swift).

To see the available code libraries and how to install them, see the [Ingestion Methods documentation page](#).

3. Platforms

LogDNA supports several computing platforms and platforms as a service (PaaS) including Heroku, Kubernetes, Docker, Fluentd, Flynn, CloudWatch, Elastic Beanstalk, and Cloud Foundry. Using platform integration lets you log all of your applications without having to deploy code libraries, even if you don't have access to install the LogDNA agent.

For instructions, log into your LogDNA account and follow the instructions on the [Add a Log Source](#) screen.



4/ Getting Started with LogDNA

4. Syslog

You can send logs directly from your devices to LogDNA using syslog. This lets you log devices that don't support the LogDNA agent or libraries, such as switches and routers, printers, and embedded devices. Enabling syslog generates a new syslog endpoint unique to your account, which you can use to route logs from your devices directly to LogDNA.

For instructions, visit the [syslog documentation page](#).

5. REST API

LogDNA provides a REST API endpoint for ingesting JSON-formatted logs. This lets you send customizable log events from systems that don't support either the LogDNA agent or code libraries. For example, if you use a service that supports webhooks (such as a source code repository or build server), you can use the LogDNA REST API to log activities that occur within that service.

For instructions, follow the [ingest documentation page](#).

Managing Users and Roles

LogDNA provides comprehensive role-based access controls (RBAC) for controlling user access to various resources in LogDNA. Users are not assigned permissions directly, but instead inherit their permissions through roles. LogDNA provides the following roles:

- **Owner:** The owner of the organization. The owner has unrestricted access to all resources, including the ability to add or remove other users.
- **Admin:** Admins have the second highest level of access after owners. Admins also have unrestricted access to resources, and an organization can have multiple admins.
- **Member:** Members are standard users in LogDNA. They have access to a limited number of settings, and their access to logs can be further restricted via groups.

You can also assign members to groups to limit the scope of logs that members have access to. For example, you may want to prevent your QA team from viewing production logs, but without limiting their access to test logs. Groups allow you to filter the logs available to a specific role using [LogDNA's search syntax](#), and assign specific members to that group. When those users sign into LogDNA, they will only see the filtered logs. Sign into the LogDNA web app and navigate to [Settings > Team](#)



4/ Getting Started with LogDNA

Enabling Single Sign-On

LogDNA supports single sign-on (SSO) via the Security Assertion Markup Language (SAML) protocol. SSO lets users log into LogDNA using the same credentials as other SSO-enabled applications in your infrastructure. If you are using a private cloud or on-premise instance of LogDNA, please [contact our support team](#) to enable SAML SSO for your deployment.

To add SSO to LogDNA, follow the instructions on the [SAML SSO documentation](#) page (). This page also includes instructions on configuring OneLogin and Okta.

Creating Alerts

Alerts in LogDNA are tied to specific views. When the number of logs in a view exceed (or fall below) a threshold within a given period of time, the alert fires and LogDNA sends a notification. LogDNA supports several notification channels including:

- Email
- Webhook
- Slack
- Pagerduty
- OpsGenie
- Datadog
- AppOptics
- VictorOps

To learn more about creating and using alerts, refer to our [documentation page on creating views and alerts](#).



4/ Getting Started with LogDNA

Enabling Log Archives

LogDNA retains logs for 7–30 days, depending on your plan. If you need to retain your log data beyond this period, LogDNA provides an archiving feature that automatically compresses and sends your log data to an external storage service. This process occurs automatically on a nightly basis and supports the following services:

- AWS S3
- Azure Blob Storage
- Google Cloud Storage
- OpenStack Object Store (Swift)
- DigitalOcean Spaces
- IBM Cloud™ Object Storage

Each night, LogDNA automatically encrypts, compresses, and sends an archive to your storage service. These logs are stored in JSON format, and encrypted server-side where possible. While LogDNA does not support re-ingesting archives, you can use other tools to parse your log archives including Amazon Athena and Google BigQuery. You can also use JSON parsing tools such as jq to read your log files.

To enable log archiving, sign in to your LogDNA account and navigate to [Settings > Archiving](#). Select your storage service from the drop-down list, then follow the instructions that appear. Click the Save button to enable the archiving process. Note that it may take 24–48 hours for the first archive to appear in your storage service.

To learn more about archiving, visit the [Archiving Log Files documentation page](#).



4/ Getting Started with LogDNA


Next Steps

Congratulations on your new LogDNA deployment! Now that you have a comprehensive log management solution in place, be sure to try LogDNA's additional features such as:

- [Custom log parsing](#)
- [Graphing](#)
- [Custom views](#)
- [Version control system integration](#)
- [Embedded views](#)

To learn more about LogDNA's features, please visit our [documentation site](#).

If you have any questions about your deployment, please [contact us](#) or join the [LogDNA Slack channel](#).



To see how easy it is to get started with LogDNA, sign up for a free 14-day trial and start managing your logs today. Want a logging expert to assess your infrastructure? Contact us at onboarding@logdna.com



DEPLOY NOW

LogDNA is a leading provider of Log Management applications that provide deep insight into development and production environments. With customers like IBM, Instacart and Lime, LogDNA is currently enabling over 2,000 companies with their modern log management platform. LogDNA's flexible deployment options enable teams to ingest, aggregate and view all of their log data regardless of data residency and infrastructure. Their Multi-Cloud solution provides DevOps teams with ability to effortlessly control cost and manage their data volumes.

For more information, visit <https://logdna.com>