Jason Wilder's Blog

Software developer and architect interested in scalability, performance and distributed systems.

Centralized Logging

Jan 3, 2012 · 5 minute read · 12 Comments

logging fluentd logstash architecture

Logs are a critical part of any system, they give you insight into what a system is doing as well what happened. Virtually every process running on a system generates logs in some form or another. Usually, these logs are written to files on local disks. When your system grows to multiple hosts, managing the logs and accessing them can get complicated. Searching for a particular error across hundreds of log files on hundreds of servers is difficult without good tools. A common approach to this problem is to setup a centralized logging solution so that multiple logs can be aggregated in a central location.

So what are your options?

File Replication

A simple approach is to setup file replication of your logs to a central server on a cron schedule. Usually rsync and cron are used since they are simple and straightforward to setup. This solution can work for a while but it doesn't provide timely access to log data. It also doesn't aggregate the logs and only co-locates them.

Syslog

Another option that you probably already have installed is syslog. Most people use rsyslog or syslog-ng which are two syslog implementations. These daemons allow processes to send log messages to them and the syslog configuration determines how the are stored. In a centralized logging setup, a central syslog daemon is setup on your network and the client logging dameons are setup to forward messages to the central daemon. A good write-up of this kind of setup can be found at: Centralized Logging Use Rsyslog

Syslog is great because just about everything uses it and you likely already have it installed on your system. With a central syslog server, you will likely need to figure out how to scale the server and make it highly-available.

- **syslog-ng**
- **rsyslog**

Distributed Log Collectors

A new class of solutions that have come about have been designed for high-volume and high-throughput log and event collection. Most of these solutions are more general purpose event streaming and processing systems and logging is just one use case that can be solved using them. All of these have their specific features and differences but their architectures are fairly similar. They generally consist of logging clients and/or agents on each specific host. The agents forward logs to a cluster of collectors which in turn forward the messages to a scalable storage tier. The idea is that the collection tier is horizontally scalable to grow with the increase number of logging hosts and messages. Similarly, the storage tier is also intended to scale horizontally to grow with increased volume. This is gross simplification of all of these tools but they are a step beyond traditional syslog options.

- **Scribe** - Scribe is scalable and reliable log aggregation server used and released by Facebook as open source. Scribe is written in C++ and uses Thrift for the protocol encoding. Since it uses thrift, virtually any language can work with it.

- **Flume** - Flume is an Apache project for collecting, aggregating, and moving large amounts of log data. It stores all this data on HDFS.

- **logstash** - logstash lets you ship, parse and index logs from any source. It works by defining inputs (files, syslog, etc.), filters (grep, split, multiline, etc..) and outputs (elasticsearch, mongodb, etc..). It also provides a UI for accessing and searching your logs. See Getting Started

- **Chukwa** - Chukwa is another Apache project that collects logs onto HDFS.

- **fluentd** - Fluentd is similar to logstash in that there are inputs and outputs for a large variety of sources and destination. Some of it’s design tenets are easy installation and small footprint. It doesn't provide any storage tier itself but allows you to easily configure where your logs should be collected.

- **kafka** - Kafka was developed at LinkedIn for their activity stream processing and is now an Apache incubator project. Although Kafka could be used for log collection this is not it's primary use case. Setup requires Zookeeper to manage the cluster state.

- **Graylog2** - Graylog2 provides a UI for searching and analyzing logs. Logs are stored in MongoDB and/or elasticsearch. Graylog2 also provides the GELF logging format to overcome some issues with syslog message: 1024 byte limit and unstructured log messages. If you are logging long stacktraces, you may want to look into GELF.

- **splunk** - Splunk is commercial product that has been around for several years. It provides a whole host of features for not only collecting logs but also analyzing and viewing them.

Update: I wrote a post comparing Fluentd vs Logstash.

Hosted Logging Services

There are also several hosted "logging as a service" providers as well. The benefit of them is that you only need to configure your syslog forwarders or agents and they manage the collection, storage and access to the logs. All of the infrastructure that you have to setup and maintain is handled by them, freeing you up to focus on your application. Each service provide a simple setup (usuallysyslog forwarding based), an API and a UI to support search and analysis.

- **loggly**
- **papertrail**
- **logentries**

I go into more detail how all of these fit together in Centralized Logging Architecture.

Tweet Like Share 9 people like this. Be the first of your friends. G+1 7

12 Comments Jason Wilder's Blog 1 Login

Recommend 7 Share Sort by Best

Join the discussion…

Kurt • 2 years ago

Check out Sumo Logic for SaaS based logging solutions.

3 ^ | ⌄ • Reply • Share ›

Nick ➦ Kurt • a year ago

Best dashboards, easiest to as well manage in my opinion.

1 ^ | ⌄ • Reply • Share ›

coematrix • a year ago

https://insightextensions.code... seems to support this with their router service. I use their solution and they are in the works of support cloud logging as well.

1 ^ | ⌄ • Reply • Share ›

Ramashish Baranwal • 2 years ago

Useful article.
The link to Fluentd vs Logstash is however broken.

1 ^ | ⌄ • Reply • Share ›

Jason Wilder Mod ➦ Ramashish Baranwal • 2 years ago

Thanks for pointing that out. Fixed!

^ | ⌄ • Reply • Share ›

Rakesh Sankar • 10 months ago

Thank you writing this article - pretty useful.

^ | ⌄ • Reply • Share ›

Jim Sherman • a year ago

Nice article.
We had good experience using Stackify (www.stackify.com). I like how it combines logs statements with errors and their context so when something happens you see all the relevant logs and info of what happened at that time. Something that many of the tools you mentioned here lacked (or at least didn't have it when we tested them)

^ | ⌄ • Reply • Share ›

Jason Raymond ➦ Jim Sherman • a year ago

Ditto on Stackify, I use their error and log management solution as well. I like also being able to monitor both error rates and their frequencies (as oppose to getting million errors on the same issue) and being able to monitor and being notified for a spike in logs of specific string

^ | ⌄ • Reply • Share ›

ROCKY MAJUMDAR • a year ago

Nice article Jason.

⌃ | ⌄ • Reply • Share ›

markhu • 2 years ago

LogStash's use of Redis has me intrigued about Redis vs. MemCache and/or RabbitMQ, not to mention Storm.

⌃ | ⌄ • Reply • Share ›

John Wheeler • 2 years ago

still useful article thanks!

⌃ | ⌄ • Reply • Share ›

Logfreak • 3 years ago

I have had great success with nxlog for centralizing windows and unix logs. It has great support for handling various log formats that some of the mentioned tools cannot cope with easily.

⌃ | ⌄ • Reply • Share ›

ALSO ON **JASON WILDER'S BLOG**

WHAT'S THIS?

Fluentd vs Logstash

14 comments • 2 years ago

Jordan Sissel — It's also worth noting that your use case (simply ship logs to s3) does a poor job of showcasing

Squashing Docker Images

6 comments • a year ago

Jason Wilder — You can change the temp dir used by setting TMPDIR. The README has an example:

Docker Log Management Using Fluentd

Docker Service Discovery

15 comments • a year ago