Choosing a Centralized Logging Solution

In this lesson, we will see which solution to choose as a centralized logging solution.

WE'LL COVER THE FOLLOWING

Database for logs

The first thing we need to do is to find a place where we'll store logs. Given that we want to have the ability to filter log entries, storing them in files should be discarded from the start. What we need is a database, of sorts. It is more important that it is fast than transactional, so we are most likely looking into a solution that is an in-memory database. But, before we take a look at the choices, we should discuss the location of our database. Should we run it inside our cluster, or should we use a service? Instead of making that decision right away, we'll explore both options, before we make a choice.

Logging-as-a-service solutions

There are two major groups of logging-as-a-service types. If we are running our cluster with one of the Cloud providers, an obvious choice might be to use a logging solution they provide. EKS has **AWS CloudWatch**, GKE has **GCP Stackdriver**, and AKS has **Azure Log Analytics**. If you choose to use one of the Cloud vendors, that might make a lot of sense. Why bother with setting up your own solution of looking for a third-party service, if everything is already set up and waiting for you? We'll explore them soon.

Outside hosting vendors

Outside hosting vendors

Since my mission is to provide instructions that work for (almost) anyone, we'll also explore one of the logging-as-a-service solutions found outside hosting vendors. But, which one should we select? There are too many solutions in the market. We could, for example, choose Splunk or DataDog. Both are excellent choices, and both are much more than only logging solutions. We can use them to collect metrics (like with `Prometheus`). They provide dashboards (like `Grafana`), and a few other things. Later on, we'll discuss whether we should combine logs and metrics in a single tool. For now, our focus is only on logging, and that's the main reason we'll skip Splunk, DataDog, and similar comprehensive tools that offer much more than what we need. That does not mean that you should discard them, but rather this chapter tries to maintain the focus on logging.

There are many logging services available, with Scalyr, logdna, and sumo logic being only a few. We won't go through all of them since it would take much more time and space than I feel is useful. Given that most services are very similar when logging is involved, I'll skip a detailed comparison and jump straight into Papertrail, my favorite logging service. Please bear in mind that we'll use it only as an example. I will assume that you'll check at least a few others and make your own choice based on your needs.

Self-hosted logging solution

Logging-as-a-service might not be a good fit for all. Some might prefer a self-hosted solution, while others might not even be allowed to send data outside their clusters. In those cases, a self-hosted logging solution is likely the only choice. Even if you are not restrained to your own cluster, there might be other reasons to keep it inside, latency being only one of many. We'll explore a self-hosted solution as well, so let us pick one. Which one will it be?

Traditional databases

Given that we need a place to store our logs, we might look into traditional databases. However, most of them would not fit our needs. Transactional databases like **MySQL** require fixed schemas so we can discard them immediately. **NoSQL** is a better fit so we might choose something like **MongoDB**. But, that would be a poor choice since we require the ability to

perform very fast free-text searches. For that, we probably need an in-memory database. **MongoDB** is not one of those. We could use Splunk Enterprise, but this course is dedicated to free (mostly open-source) solutions. The only exception we made so far is with Cloud providers, and I intend to keep it that way.

The few requirements we mentioned (fast, free-text, in-memory, and a free solution) limit our potential candidates to only a few. Solr is one of those, but its usage has been dropping, and it is rarely used today (for a good reason). The solution that sticks out from a tiny crowd is Elasticsearch. If you have a preference for a different on-prem solution, consider the examples we'll go through as a set of practices that you should be able to apply to other centralized logging solutions.

All in all, we'll explore an example of independent logging-as-a-service offering (**Papertrail**), we'll explore the solutions provided with Cloud hosting vendors (**AWS CloudWatch**, **GCP Stackdriver**, and **Azure Log Analytics**), and we'll try to set up **ElasticSearch** with a few friends. Those should provide enough examples for you to choose which type of solution fits your use case the best.

Q

Transactional databases like **NoSQL** require fixed schemas so we can discard them immediately.

COMPLETED 0%

1 of 1

But, before we explore the tools where we will store logs, we need to figure out how to collect them and ship them to their final destination. Let's see that in the next lesson.