Switching to Elasticsearch for Storing Metrics

In this lesson, we will see whether we can use Elasticsearch for storing metrics or not.

WE'LL COVER THE FOLLOWING

Using elastic.co to store our metrics

Now that we had **Elasticsearch** running in our cluster and knowing that it can handle almost any data type, a logical question could be whether we can use it to store our metrics besides logs. If you explore elastic.co, you'll see that metrics are indeed something they advertise. If it could replace `Prometheus`, it would undoubtedly be beneficial to have a single tool that can handle not only logs but also metrics. On top of that, we could ditch `Grafana` and keep **Kibana** as a single UI for both data types.

Right tool for the job

Nevertheless, I would strongly advise against using **Elasticsearch** for metrics. It is a general-purpose free-text no-SQL database. That means that it can handle almost any data but, at the same time, it does not excel at any specific format. `Prometheus`, on the other hand, is designed to store time-series data which are the preferred way of exposing metrics. As such, it is more limited in what it does. But, it handles metrics much better than **Elasticsearch**. I believe that using the right tool for the job is better than having a single tool that does too many things, and if you believe the same, `Prometheus is a much better choice for metrics`.

When compared to **Elasticsearch**, and focused only on metrics, `Prometheus'` requires much fewer resources (as you already noticed), it is faster, and it has a much better query language. That shouldn't come as a surprise given that both tools are great, but only `Prometheus` is designed to work exclusively with metrics. The increased cost of maintaining an additional tool is well paid off by having a better (and more focused) solution.

Did I mention that notifications generated through `Prometheus` and `Alertmanager` are better than those through **Elasticsearch**?

`Prometheus` integration with Kubernetes

There's one more important thing to note. `Prometheus` integration with Kubernetes is way better than what **Elasticsearch** offers. That is not a surprise since `Prometheus` is based on the same cloud-native principles as Kubernetes and both belong to Cloud Native Computing Foundation. **Elasticsearch**, on the other hand, comes from a more traditional background.

> 🔍 *Elasticsearch* is excellent, but it does too much. Its lack of focus makes it inferior to Prometheus for storing and querying metrics, as well as sending alerts based on such data.

Using `Prometheus` for logs

If replacing `Prometheus` with **Elasticsearch** is not a good idea, can we invert the question? Can we use `Prometheus` for logs? The answer is a definite no. As already stated, `Prometheus` is focused only on metrics. If you do adopt it, you need a different tool for storing logs. That can be **Elasticsearch**, **Papertrail**, or any other solution that fits your needs.

Using Kibana for metrics

How about **Kibana**? Can we ditch it in favor of `Grafana`? The answer is yes, but don't do that. While we could create a table in `Grafana` and attach it to **Elasticsearch** as a data source, its capability to display and filter logs is inferior. On the other hand, `Grafana` is much more flexible than **Kibana** for displaying graphs based on metrics. So, the answer is similar to the

Elasticsearch vs. `Prometheus` dilemma. Keep `Grafana` for metrics and use **Kibana** for logs, if you chose to store them in **Elasticsearch**.

Should you add **Elasticsearch** as yet another data source in `Grafana`? If you took previous recommendations, the answer is most likely no. There is not much value in presenting logs as graphs. Even the predefined graph available in **Kibana's** *Explore* section is, a waste of space. There is no point in showing how many log entries we have in total, nor even how many are error entries. We use metrics for that.

> 🔍 Logs themselves are too expensive to parse, and most of the time they do not provide enough data to act as metrics.

Q

Elasticsearch is designed to store time-series data which are the preferred way of exposing metrics.

COMPLETED 0%

1 of 1

We saw several tools in action, but we did not yet discuss what we truly need from a centralized logging solution. We'll explore that in the next lesson.