

ElasticsearchAdjuster - Smart Elasticsearch Deployments

Degree programme : BSc in Computer Science | Specialisation : Data Engineering
Thesis advisor : Prof. Dr. Mascha Kurpicz-Briki
Expert : Dr. Joachim Wolfgang Kaltz (Camptocamp SA)

Elasticsearch (ES), a distributed search engine, is often deployed with configurations based on best practices and assumptions about future workload. As circumstances regularly change during operation, the cluster setup and configuration must be continuously optimised. This results in frequent manual adjustments and analysis. In this thesis, a Kubernetes (K8S) based environment was set up and a solution - the ElasticsearchAdjuster - was developed to automate this.

Used technologies

Kubernetes (K8S), a container orchestration platform, was set up to host the ecosystem. Different K8S-distributions as well as a suitable monitoring solution were evaluated. The choice fell on MicroK8S and Prometheus. Prometheus ingests metrics from all main components of the system and can send alarms. Notifications for the operator are sent via the chat platform Signal. ESRally, the official benchmark solution for Elasticsearch was used for performance evaluations. For the setup, knowledge about operating Elasticsearch on K8S from the pre-project “Smart Elasticsearch Deployments” was applied.

Approach

Setting up an environment meant defining a base configuration for the K8S cluster, deploying the needed ES instances and attaching them to Prometheus. This was executed on virtual servers provided by BFH and represented the foundation for our work on the ElasticsearchAdjuster, short Elad.

In a further step, a list of different critical situations on which to act was defined, called triggers. These triggers not only cover Elasticsearch, but also the underlying infrastructure. This might include problems with resources or specific components that may have a negative impact on Elasticsearch.

Result

ElasticsearchAdjuster was developed to handle problems during operation by enhancing notification and performing automatic configuration adjustments. For this purpose, several metric based rules were defined in Prometheus that trigger an alarm in the event of deviation. This alarm will then, depending on the possible reaction, be handled by Elad for automatic resolution or directly end in a Signal message for an operator. If the alarm is sent to Elad it will, depending on the nature of the problem, either change configuration directly or validate it using a benchmark test. This is carried out on a separate benchmark cluster. If the new configuration succeeds considering performance, it is applied on the productive environment. If Elad cannot find a solution for the situation or configuration changes are not possible due to other constraints, it sends a notification via Signal.

Conclusion

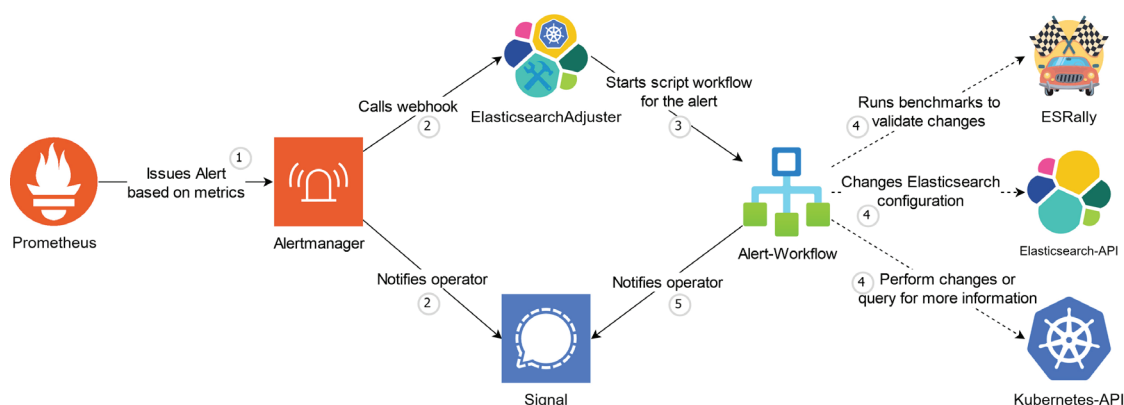
A measurable positive effect could be observed in situations where Elad is automatically adjusting configurations and solving problems. In other cases, Elad provides more information for operators by enriching alarming messages and reporting issues pre-emptively where possible. Still, more development work is needed to benefit from this case study.



Marc Häsler



Yaron Timon Walter



How critical situations are handled by the ElasticsearchAdjuster ecosystem