EFK

When running multiple services and applications on a Kubernetes cluster, a centralized, cluster-level logging stack can help you quickly sort through and analyze the heavy volume of log data produced by your Pods. One popular centralized logging solution is the Elasticsearch, Fluentd, and Kibana (EFK) stack.

Elasticsearch is a real-time, distributed, and scalable search engine which allows for full-text and structured search, as well as analytics. It is commonly used to index and search through large volumes of log data but can also be used to search many different kinds of documents.

Elasticsearch is commonly deployed alongside Kibana, a powerful data visualization frontend and dashboard for Elasticsearch. Kibana allows you to explore your Elasticsearch log data through a web interface and build dashboards and queries to quickly answer questions and gain insight into your Kubernetes applications.

We use Fluentd to collect, transform, and ship log data to the Elasticsearch backend. Fluentd is a popular open-source data collector that we’ll set up on our Kubernetes nodes to tail container log files, filter and transform the log data, and deliver it to the Elasticsearch cluster, where it will be indexed and stored.

EFK Architecture

Diagram

Description automatically generated

EKF components get deployed as follows,

* Fluentd: - Deployed as daemonset as it need to collect the container logs from all the nodes. It connects to the Elasticsearch service endpoint to forward the logs.
* Elasticsearch: - Deployed as statefulset as it holds the log data. We also expose the service endpoint for Fluentd and Kibana to connect to it.
* Kibana: - Deployed as deployment and connects to elasticsearch service endpoint.

Diagram

Description automatically generated