Install EFK stack in K8's cluster using helm

Objective:

Monitoring, alerting and log aggregation are essential for the smooth functioning of a production grade Kubernetes cluster.

Application and system logs help us understand what is happening inside a cluster. The logs are particularly useful for debugging problems and monitoring cluster activity.

In a microservices architecture, a single business operation might trigger a chain of downstream microservice calls, which can be pretty challenging to debug. Things, however, can be easier when the logs of all microservices are centralized and each log event contains details that allow us to trace the interactions between the applications.

We can use Elastic Stack along with Docker to collect, process, store, index and visualize logs of microservices.

What is Elastic Stack?

Elastic Stack is a group of open source applications from Elastic designed to take data from any source and in any format and then search, analyze, and visualize that data in real time. It was formerly known as *ELK Stack*, in which the letters in the name stood for the applications in the group: *Elasticsearch*, *Logstash* and *Kibana*. A fourth application, *Beats*, was subsequently added to the stack.

Elasticsearch is a real-time, distributed storage, JSON-based search, and analytics engine designed for horizontal scalability, maximum reliability, and easy management. It can be used for many purposes, but one context where it excels is indexing streams of semi-structured data, such as logs or decoded network packets.

Kibana is an open source analytics and visualization platform designed to work with Elasticsearch. Kibana can be used to search, view, and interact with data stored in Elasticsearch indices, allowing advanced data analysis and visualizing data in a variety of charts, tables, and maps.

Beats are open source data shippers that can be installed as agents on servers to send operational data directly to Elasticsearch or via Logstash, where it can be further processed and enhanced. There's a number of Beats for different purposes:

- Filebeat: Log files
- Metricbeat: Metrics
- Packetbeat: Network data
- Heartbeat: Uptime monitoring

As we intend to ship log files, Filebeat will be our choice.

Logstash is a powerful tool that integrates with a wide variety of deployments. It offers a large selection of plugins to help you parse, enrich, transform, and buffer data from a variety of sources. If the data requires additional processing that is not available in Beats, then Logstash can be added to the deployment.

Putting the pieces together

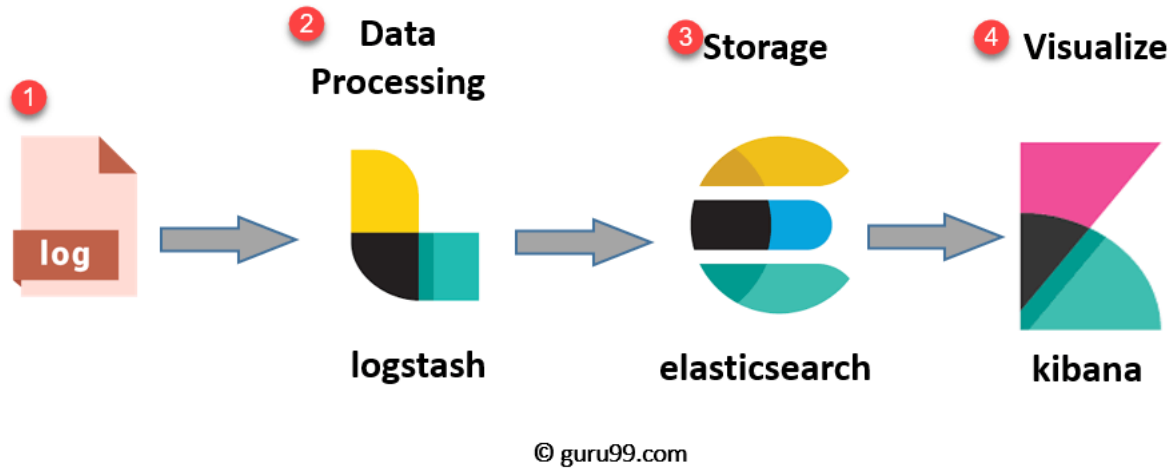
The following illustration shows how the components of Elastic Stack interact with each other:

In a few words:

- Filebeat collects data from the log files and sends it to Logststash.
- Logstash enhances the data and sends it to Elasticsearch.
- Elasticsearch stores and indexes the data.
- Kibana displays the data stored in Elasticsearch.

ELK Stack Architecture

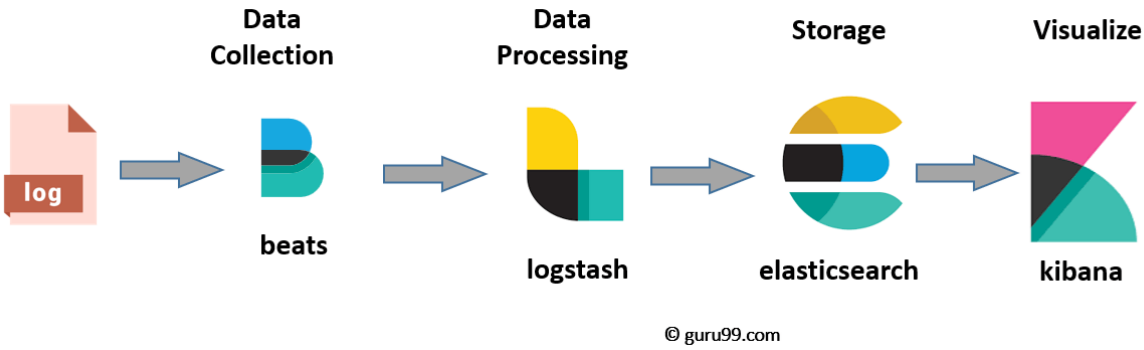
Here is the simple architecture of ELK stack

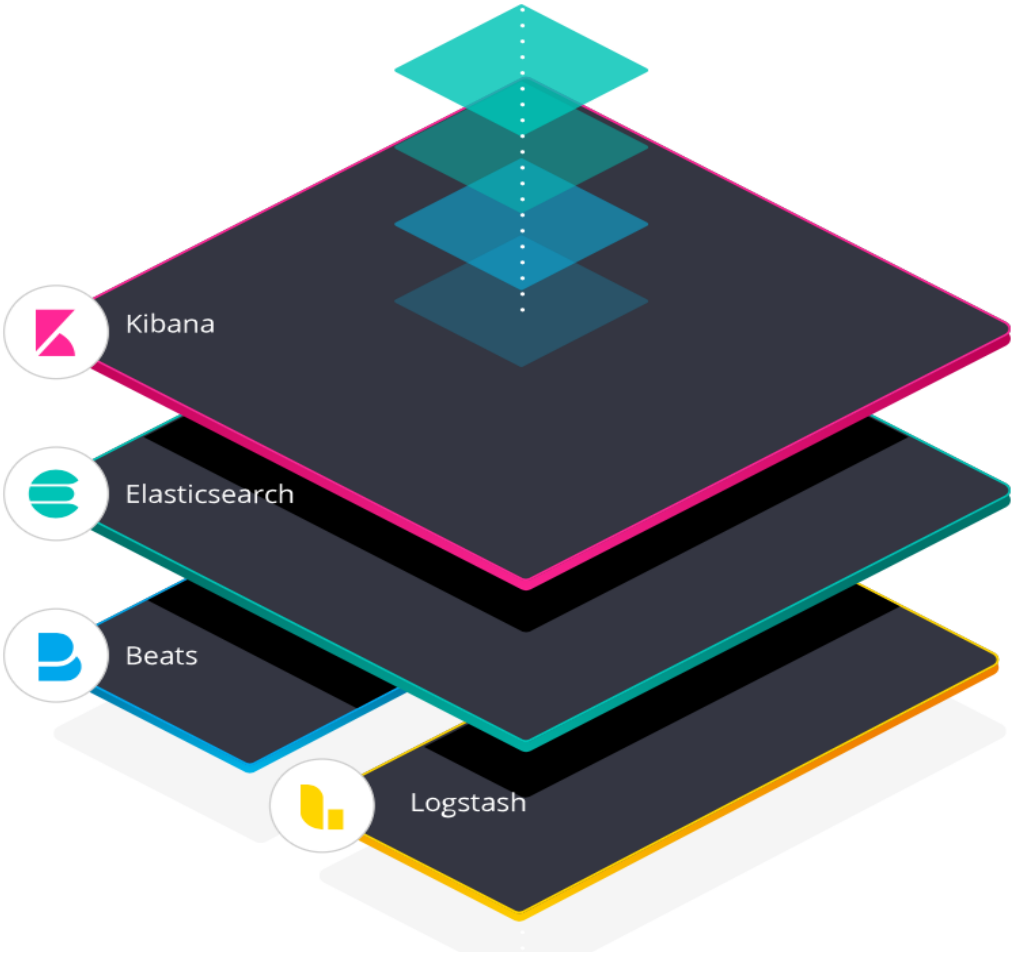


- **Logs:** Server logs that need to be analyzed are identified
- **Logstash:** Collect logs and events data. It even parses and transforms data

- **ElasticSearch:** The transformed data from Logstash is Store, Search, and indexed.
- **Kibana:** Kibana uses Elasticsearch DB to Explore, Visualize, and Share

However, one more component is needed or Data collection called Beats. This led Elastic to rename ELK as the Elastic Stack.





Prerequisite:

1) Kubernetes Cluster With Storage Class Configured(Dynamic Provisioning).

   If Storage Class is not Configured then make persistence enabled as false in helm values files # Which is not suggestable as if pods are terminated and recreated we will lost the data.

2) Kubernetes nodes with minimum 4GB RAM With 2 Core Processer.
3) Server which has kubectl & Helm Configured.

Install EFK(Elastic FileBeat Kibana) Using Helm.

```
$ kubectl create ns efk

$ helm repo add elastic https://helm.elastic.co

$ helm show values elastic/elasticsearch >> elasticsearch.values

# Update replicas & minimumMasterNodes , And Resource requests and limits (Min: Memor
y 2Gi)in elasticsearch.values

$ helm install elasticsearch elastic/elasticsearch -f elasticsearch.values -n efk

$ helm show values elastic/kibana >> kibana.values

# Resource requests and limits (Memory 1Gi) in & Service type as NodePort Or Load Balance
r And Port as 80
   kibana.values

$ helm install kibana elastic/kibana -f kibana.values -n efk

$ helm install filebeat elastic/filebeat -n efk



# Optional
$ helm show values elastic/ metricbeat >> metricbeat.values
# Update hostNetworking as true
$ helm install  metricbeat elastic/metricbeat -f metricbeat.values -n elk
```

Access Kibana Dash Board Using Node IP & NodePort Or Load Balancer based on service type we created. & Create Index Pattern.

Deploy Demo Sample Application.

Note: Make sure your cluster have default storage class configured as we are using PVC with dynamic storage class(Default) to provision PV for mongo DB POD which we are deploying as part of below demo app.

Manifests Link below.

https://raw.githubusercontent.com/MithunTechnologiesDevOps/Kubernates-Manifests/master/SpringBoot-Mongo-DynamicPV.yml

Deploy using below command.

kuebctl apply -f https://raw.githubusercontent.com/MithunTechnologiesDevOps/Kubernates-Manifests/master/SpringBoot-Mongo-DynamicPV.yml

Using Kibana we visualise the logs.