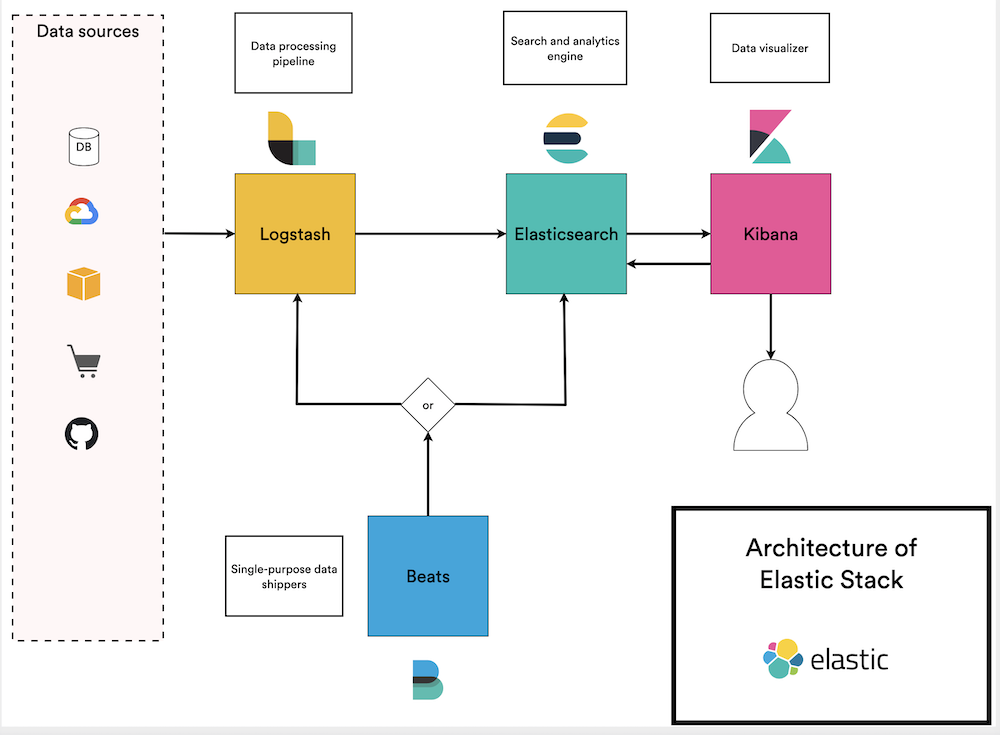
**CAPTURING LOGS USING ELK STACK**

\*501

DESIGN DIAGRAM FOR ELK STACK

**ELK Stack Components**:

**Elastic Search:**

This component is used to index, store and extract your data. Elastic search offers full text search, real-time analytics, scalability and high availability, making it a terrific solution for all your data-extraction needs.

**Logstash:**

No matter what type of logs you’re currently managing, they can be processed by Logstash. This component is a workhorse, able to collect and parse logs and then send them to Elastic search for indexing.

**Kibana**:

This front-end dashboard allows you to visualize your data using pie charts, graphs, scatter plots, maps and more. With Kibana, it’s easy to spot emerging patterns and trends in big data sets that would otherwise be tedious to plow through.

**Approach:**

The best approach is to create an application and deploy it as a microservice which accepts logs from all the frontend and backend microservices and then it sends those logs to logstash. Nodejs fits this scenario very well cause we will have heavy writes here.

Why to create a centralized application to collect logs?

1. By isolating it as a separate service, we can scale it very well on demand
2. Better Security.

In case of collecting frontend logs, we can configure our frontend applications to send logs to logstash directly but we need to configure them with logstatsh ip address or public url and port. It is better not to share those details publicly.

1. Reusability.

We can use this centralized microservice to collect logs from multiple projects across our organization.

**Costs Involved:**

Only costs involved in this infrastructure are compute resources, storage and bandwidth pricing.

Note: All Components in ELK Stack are completely open source. There are no other costs like licensing or any special fees.

**Efforts Required:**

**For SRE Team:**

SRE Team will have heavy work regarding initial setup and also future maintenance and scaling of different components of ELK Stack.

**For Dev Team:**

Need to put efforts to create a centralized logging app. Apart from that, remaining work is pretty much same as we connect to any third party logging service.

**Pros And Cons**:

**Pros:**

1. Completely Opensource solution
2. Centralized Logging Capabilities
3. Real-time log Analysis & Visualization (Kibana)
4. Supports multiple datasources
5. Official Clients in almost all major Programming Languages

**Cons:**

1. High cost of ownership
2. Complex Management Requirements(SRE Team)
3. Stability and uptime issues as the data volume grows
4. Need for heavy compute and storage resources (cause ELK stack stores all the log content in index to provide full text search capabilities).
5. As the data volume grows, our infrastructure costs also increase heavily.