Introduction

The solution/framework helps support proactive and reactive logging and monitoring use cases for all customer assets residing on AWS. This helps

- Fast searches over large log volumes
- Scalable cloud-based log management
- Helps Investigate production issues faster
- Easy integration with various AWS service to provide easy monitoring of entire stack.

Design

Log Ingestion and Processing

Framework leverages **EMR/Spark/Spark Streaming to create a unified ETL approach** for extracting log data from various log sources both in batch and streaming mode, transform and persist the data into sinks (Elastic Search).

- **Log Processing** :- Since it leverages Spark and the framework being built is config driven, the number or type of log data sources can easily be expanded in future to include any streaming/batch log data source (For e.g Kafka,HDFS, DB etc) and log data sink (Search, NoSQl, Sql) with the change of few config parameters.
- **Scalability-** The solution leverages EMR (managed Hadoop) for running the spark log ingestion and processing jobs. EMR supports auto scaling and can process TBs/PBS of log data with ease. Each spark Job leverages a bunch of executors (JVMs) in a node to process the data.
- **Flexibility/Future Proof** – By leveraging open source Spark APIs, you get access to all spark Connectors for various sources and sinks thereby proving the flexibility to expand the solution in future.
- **Pricing** :- EMR supports auto scaling and can help save on cost.

- **Error handling-** The solution sticks to one unified approach for processing and ingesting all kind of log data sources/sinks and has less moving parts thereby reducing the failures**. Error handling is built in the framework to ensure data is not lost**. This is achieved via checkpointing and custom error handling code.
- **Orchestration/Automation**- Orchestration/Automation of the jobs is implemented using Cloudformation/Stepfunctions/Lambda.
- **Metadata**- Log related config files leverage DynamoDB.

Log Storage,Analytics and Monitoring

It leverages **AWS managed ELK stack** for Storing, analyzing, and correlating application and infrastructure log data to find and fix issues faster and improve application performance. It is a **fully managed service** that helps monitor, and troubleshoot your applications The service provides support for open source Elasticsearch APIs, managed Kibana, integration with Logstash and other AWS services, and built-in alerting and SQL querying.

- **Flexibility-**
  - Supports any **unstructured data** and it is schema on write.
- **Security-**
  - Provides support for **Index, doc, and field** security
- **Visualization-**
  - **Kibana** offers **intuitive charts and reports that you can use to interactively navigate through large amounts of log data stored in ES**.
  - **Using Kibana's pre-built aggregations and filters**, you can run a variety of analytics like histograms, top-N queries, and trends with just a few clicks.
  - You can rapidly create **dashboards** that pull together charts, maps, and filters to display the full picture of your data. All you need is a browser to view and explore the data.
- **Machine Leaning-**
  - **Easy to detect anomalies hiding in Elasticsearch data** and explore the properties that significantly influence them

with unsupervised machine learning features. Helps you achieve actionable insights.

- **Operations-**
  - Amazon Elasticsearch Service, you get the ELK stack you need, without the operational overhead.
- **Pricing-**
  - Amazon Elasticsearch Service lets you pay only for what you use – **there are no upfront costs or usage requirements**
- **Alerting-**
  - Easy to build alerts that trigger custom actions

Benefits

- The architecture/framework is **future proof** and can evolve easily with change in requirements by simply modifying the metadata in config files.
- By leveraging **EMR/Spark Open source APIs for ingesting and processing logs**, it gives you access to numerous set of connectors to various data sources/sinks, thereby proving the **flexibility** to use the same approach/solution for multiple log sources and destinations.
- By leveraging Managed ELK stack you can easily store, analyze, and correlate application and infrastructure log data to find and fix issues faster and improve application performance **without any operational overhead**.
- The architecture sticks to the practices of building a **Unified Logging Platform** for processing log data and has fewer moving components , thereby reducing the chances of failure and the maintenance overhead.

FAQ.

What are the benefits of using this Framework?

This framework enables aggregating logs from applications and services deployed and running in AWS into one centralized location (Elastic Search). This solution when deployed would enable
> Fast searches over large log volumes

> Helps Investigate production issues faster

What AWS services are used to achieve this solution?

DynamoDB (config files related to log groups are persisted in DynamoDB), EMR (Solution runs on EMR and leverages Spark), Cloudformation & Lambda/Stepfunctions (For Automation & Orchestration)

What libraries are used in the framework ?

Framework is built using Java , Spark and Spring libraries.

Can the Solution be customized ?

Yes, the solution is extensible and can be customized. Please refer to help.md for customizing and extending the solution.

Is the Solution Scalable ?

The solution runs on EMR/Spark (managed Hadoop). EMR supports auto scaling and can process TBs/PBs of log data with ease. Each Job leverages a bunch of executors (JVMs) in a node to process the data.

What service is used for Log Storage ?

Solution uses AWS managed ELK stack for Storing, Analyzing, and correlating application and infrastructure log data to find and fix issues faster and improve application performance. It is a fully managed service that helps monitor, and troubleshoot your applications. The service provides support for open source Elasticsearch APIs, managed Kibana, integration with Logstash and other AWS services, and built-in alerting and SQL querying.

Is the Solution Secure ?

The solution leverages AWS EMR/Elasticsearch/DynamoDB/Lambda. These Services support IAM, Security Groups, SSH, Data Encryption/SSL/TLS to achieve the desired Security. Elastic Search supports Index, doc, and field security too.

Is there support for Log Analytics/Visualization ?

Logs are persisted in AWS Managed ELK. Kibana offers intuitive charts and reports that can be used to interactively navigate through large amounts of log data stored in ES. Using Kibana's pre-built aggregations and filters, a variety of analytics like histograms, top-N queries, and trends with just a few clicks. You can rapidly create dashboards that pull together charts, maps, and filters to display the full picture of your data. All you need is a browser to view and explore the data.

Are there any Operational Overhead ?

Solution is designed/automated to leverage Managed Services provided by AWS such as EMR/ElasticSearch thereby reducing operational overhead.

Does the solution automate the process of managing ES indices ?

The solution automates the process of managing indices by leveraging ultrawarm storage and indices rollover.

How do I Search for logs ?

Solution leverages Amazon Elasticsearch Service for storing the log data. There are several common methods for searching documents in Amazon Elasticsearch Service (Amazon ES), including URI searches and request body searches. Amazon ES offers additional functionality that improves the search experience, such as custom packages, SQL support, and asynchronous search.