Data Analytics Helping Real-time Decision Making

Customer: The fastest-growing cinema business in the Middle East

Summary

The customer is the fastest-growing cinema business in the Middle East wanted to manage the logs from multiple environments by setting up centralized logging and visualization, this was done by implementing the EKK(Amazon Elasticsearch, Amazon Kinesis and Kibana) solution in their AWS environment.

About Customer

The customer is a cinema arm of a leading shopping mall, retail and leisure pioneer across the Middle East and North Africa. They are the Middle East's most innovative and customer-focused exhibitor, and the fastest and rapidly growing cinema business in the MENA region.

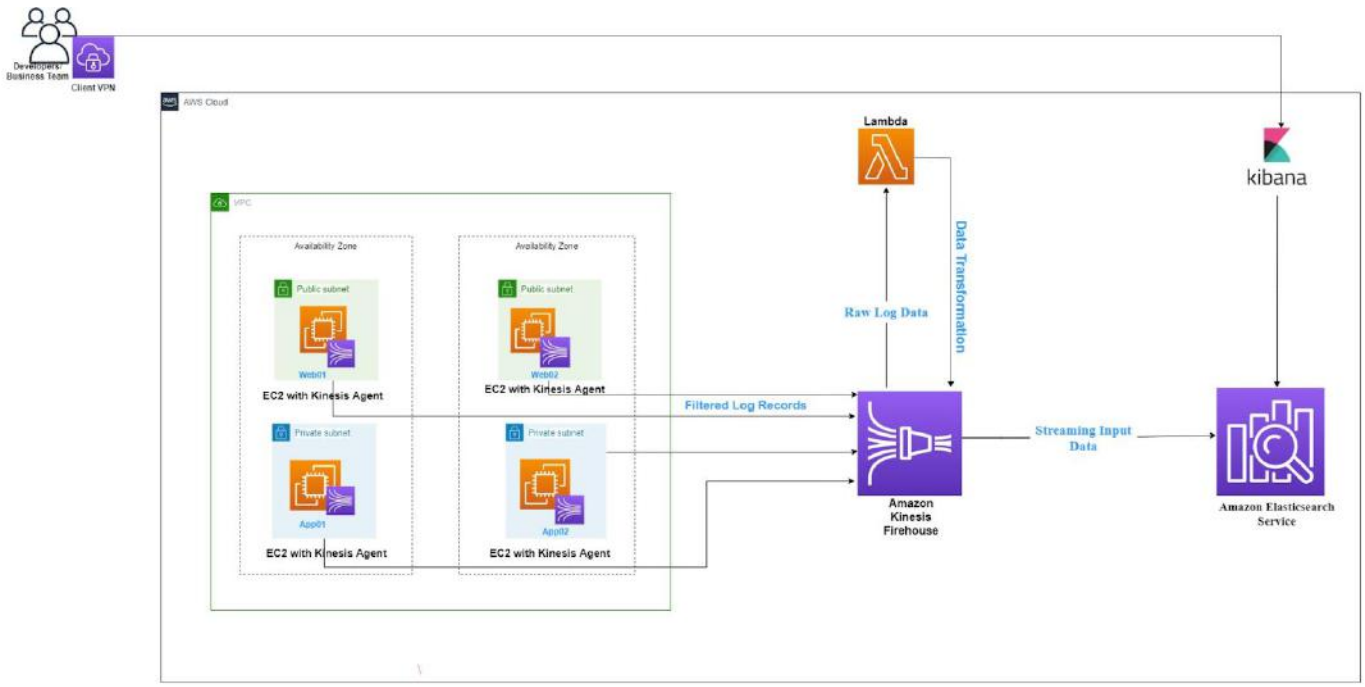
Problem Statement

The customer's applications generate huge amounts of logs from multiple servers, if any error occurs in the application it is difficult for the development team to get the logs or view the logs in real-time to troubleshoot the issue. They do not have a centralized location to visualize logs and get notified if any errors occur.

In the ticket booking scenario, by analyzing the logs that are generated by the application, an organization can enable valuable features, such as notifying the developers that error occurred in the application server while customers are booking the ticket. If the application logs can be analyzed and monitored in real-time, developers can be notified immediately to investigate and fix the issues.

Proposed Solution

Powerup built a log analytics solution on AWS using ElasticSearch as the real-time analytics engine. AWS Kinesis firehose pushes the data to ElasticSearch. In some scenarios, the Customer wanted to transform or enhance data streaming before it is delivered to ElasticSearch. Since all the application logs are in an unstructured format in the server, the customer wanted to filter the unstructured data and transform it into JSON before delivering it to Amazon Elasticsearch Service. Logs from Web, App and DB were pushed to Elasticsearch for all the six applications.



[Architecture Diagram]

Architecture Description

Amazon Kinesis Agent

- The Amazon Kinesis Agent is a stand-alone Java software application that offers an easy way to collect and send data to Kinesis Streams and Kinesis Firehose.
- AWS Kinesis Firehose Agent - daemon installed on each EC2 instance that pipes logs to Amazon Kinesis Firehose.
- The agent continuously monitors a set of files and sends new data to your delivery stream. It handles file rotation, checkpointing, and retry upon failures. It delivers all of your data in a reliable, timely, and simple manner.

Amazon Kinesis Firehose

- Amazon Kinesis Firehose is the easiest way to load streaming data into AWS. It can capture, transform, and load streaming data into Amazon Kinesis Analytics, Amazon S3, Amazon Redshift, and Amazon Elasticsearch Service, enabling near real-time analytics with existing business intelligence tools and dashboards that you're already using today.
- Kinesis Data Firehose Stream - endpoint that accepts the incoming log data and forwards to ElasticSearch

Data Transformation

Kinesis Data Firehose can invoke your Lambda function to transform incoming source data and deliver the transformed data to destinations. When you enable Kinesis Data Firehose data

transformation, Kinesis Data Firehose buffers incoming data up to 3 MB by default. Kinesis Data Firehose then invokes the specified Lambda function asynchronously with each buffered batch using the AWS Lambda synchronous invocation model. The transformed data is sent from Lambda to Kinesis Data Firehose. Kinesis Data Firehose then sends it to the destination when the specified destination buffering size or buffering interval is reached, whichever happens first.

ElasticSearch

- Elasticsearch is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.
- Store, analyze, and correlate application and infrastructure log data to find and fix issues faster and improve application performance. You can receive automated alerts if your application is underperforming, enabling you to proactively address any issues.
- Provide a fast, personalized search experience for your applications, websites, and data lake catalogs, allowing users to quickly find relevant data.
- Collect logs and metrics from your servers, routers, switches, and virtualized machines to get comprehensive visibility into your infrastructure, reducing mean time to detect (MTTD) and resolve (MTTR) issues and lowering system downtime.

Kibana

Kibana is an open-source data visualization and exploration tool used for log and time-series analytics, application monitoring, and operational intelligence use cases. It offers powerful and easy-to-use features such as histograms, line graphs, pie charts, heat maps, and built-in geospatial support. Also, it provides tight integration with Elasticsearch, a popular analytics and search engine, which makes Kibana the default choice for visualizing data stored in Elasticsearch.

- 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 easily set up dashboards and reports and share them with others. All you need is a browser to view and explore the data.
- Kibana comes with powerful geospatial capabilities so you can seamlessly layer in geographical information on top of your data and visualize results on maps.

Ingesting data to ElasticSearch using Amazon Kinesis Firehose.

Kinesis Data Firehose is part of the Kinesis streaming data platform, along with Kinesis Data Streams, Kinesis Video Streams, and Amazon Kinesis Data Analytics. With Kinesis Data Firehose, you don't need to write applications or manage resources. You configure your data producers to send data to Kinesis Data Firehose, and it automatically delivers the data to the destination that you specified. You can also configure Kinesis Data Firehose to transform your data before delivering it.

Record

The data of interest that your data producer sends to a Kinesis Data Firehose delivery stream. A record can be as large as 1000 KB.

Data producer

Producers send records to Kinesis Data Firehose delivery streams. For example, a web server that sends log data to a delivery stream is a data producer. You can also configure your Kinesis Data Firehose delivery stream to automatically read data from an existing Kinesis data stream, and load it into destinations.

Writing Logs to Kinesis Data Firehose Using Kinesis Agent

- Amazon Kinesis Agent is a standalone Java software application that offers an easy way to collect and send data to Kinesis Data Firehose. The agent continuously monitors a set of files and sends new data to your Kinesis Data Firehose delivery stream.
- The agent handles file rotation, checkpointing, and retry upon failures. It delivers all of your data in a reliable, timely, and simple manner. It also emits Amazon CloudWatch metrics to help you better monitor and troubleshoot the streaming process.
- The Kinesis Agent has been installed on all the production server environments such as web servers, log servers, and application servers. After installing the agent, we need to configure it by specifying the log files to monitor and the delivery stream for the data. After the agent is configured, it durably collects data from those log files and reliably sends it to the delivery stream.
- Since the data in the servers are unstructured and the customer wanted to send the specific format of data to ElasticSearch and visualize it on Kibana. So we configured an agent to preprocess the data and deliver the preprocessed data to AWS Kinesis Firehose. Preprocessed configuration used in the Kinesis Agent

MatchPattern

- Since the data in the logs are unstructured and needed to filter some specific records from the data. So we used the match pattern to send the record to filter the data and send it to Kinesis Firehose.
- The agent has configured in a way to capture the unstructured data using regular expression and send it to the AWS Kinesis Firehose.

An Example How we filtered the data and sent it to the kinesis firehose.

- *LOGTOJSON configuration with Match Pattern*

Sample Kinesis agent configuration:

```
{
   "optionName": "LOGTOJSON",
   "logFormat": "COMMONAPACHELOG",
   "matchPattern": "^([\\d.]+) (\\S+) (\\S+) \\[([\\w:/]+\\s[+\\-]\\d{4})\\] \"(.+?)\" (\\d{3})",
   "customFieldNames": ["host", "ident", "authuser", "datetime", "request", "response"]
}
```

The record in the server before conversion:

```
100.189.189.89 - - [27/Oct/2000:09:27:09 -0400] "GET /java/javaResources.html HTTP/1.0"
200
```

After conversion:

```
{
"Host":"100.189.189.89",
"Ident":null,
"Authuser":null,
"datetime":"27/Oct/2000:09:27:09 -0400",
"request":"GET /java/javaResources.html HTTP/1.0",
"Response":"200"
}
```

The record in the server has been converted to JSON format. The Match pattern only captures the data in the data according to regular expression and sends the data to AWS Kinesis Firehose. AWS Kinesis Firehose sends the data to Elasticsearch and can be visualized on the Kibana.

Business Benefits

- Powerup Team successfully implemented the real-time centralized log analytics solution using AWS kinesis firehose and ElasticSearch.
  - ✓ Kinesis agent was used to filtering the applications and kinesis firehose streams the logs to Elasticsearch.
  - ✓ Separate indexes were created for all 6 applications in Elasticsearch based on access log and error log.
  - ✓ A Total of 20 dashboards were created in Kibana based on error types, for example, 4xx error, 5xx error, cron failure, auth failure.
  - ✓ Created Alerts were sent to the developers using AWS SNS. when the configured thresholds, so that developers can take immediate actions on the errors generated on the application and server.
  - ✓ Developer log analysis time has greatly decreased from a couple of hours to a few minutes.

- The EKK setup implemented for the customer is a total log-analysis platform for search, analyses and visualization of log-generated data from different machines and perform centralized logging to help identify any server and application-related issues across multiple servers in the customer environment and correlate the logs in a particular time frame.
- The data analysis and visualization of EKK setup have benefited the management and respective stakeholders to view the business reports from various application streams which led to easy business decision making.

Cloud platform

AWS.

Technologies used

Lambda, Kibana, EC2, Kinesis.