Log Analytics using Amazon Web Services(AWS)

Abhilash Reddy Mandadi Dheeraj Manchukonda Sujay Umesh

Department of Computer Science Department of Computer Science Department of information

amandadi@albany.edu dmanchukonda@albany.edu sumesh@albany.edu

Abstract:

Log analytics is a common big data use case that allows you to analyze log data from websites, mobile devices, servers, sensors, and more for a wide variety of applications such as digital marketing, application monitoring, fraud detection, ad tech, gaming, and IoT. Amazon Kinesis Firehose is a fully managed service for delivering real-time streaming data to destinations such as Amazon Simple Storage Service (Amazon S3), Amazon Redshift, or Amazon Elasticsearch Service (Amazon ES). Firehose is part of the Amazon Kinesis streaming data platform, along with Amazon Kinesis Streams and Amazon Kinesis Analytics

In this project, we used Amazon Web Services to build an end-to-end log analytics solution that collects, processes, and loads both batch data and streaming data, and makes the processed data available to the users in analytics systems which are already using and in near real-time. The solution is highly reliable, cost-effective, scales automatically to varying data volumes, and requires almost no IT administration.

1. Introduction:

One of the major benefits to using Amazon Kinesis Analytics is that an entire analysis infrastructure can be created with a serverless architecture. The system created will implement Amazon Kinesis Firehose, Amazon Kinesis Analytics, and Amazon Elasticsearch Service (Amazon ES). Each of these services is designed for seamless integration with one another. The Architecture will be described in the later section.

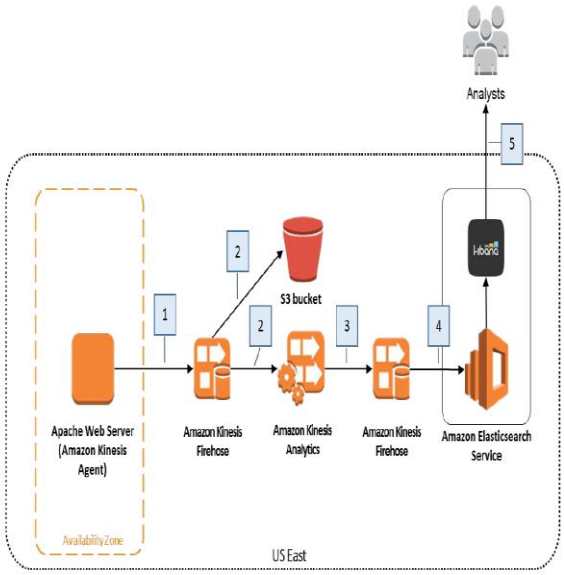
The web server in this example will be an Amazon Elastic Compute Cloud (EC2) instance. We will install the Amazon Kinesis Agent on this Linux instance, and the agent will continuously forward log records to an Amazon Kinesis Firehose delivery stream. Amazon Kinesis Firehose will write each log record to Amazon Simple Storage Service (Amazon S3) for durable storage of the raw log data, and the Amazon Kinesis Analytics application will continuously run an SQL statement against the streaming input data. The Amazon Kinesis Analytics application will create an aggregated data set every minute and output that data to a second Firehose delivery stream. This Firehose delivery stream will write the aggregated data to an Amazon ES domain. Finally, we will create a view of the streaming data using Kibana to visualize the output of our system.

Amazon Kinesis is a fully managed, cloud-based service for real-time data processing over large, distributed data streams.

1.1 Architecture:

One of the major benefits to using Amazon Kinesis Analytics is that an entire analysis infrastructure can be created with a serverless architecture. The system created in this project implements Amazon Kinesis Firehose, Amazon Kinesis Analytics, and Amazon Elasticsearch Service (Amazon ES). Each of these services is designed for

seamless integration with one another. The architecture is depicted below.



2. Terms and Definitions:

Amazon EC2

Amazon EC2 provides the virtual application servers, known as instances, to run your web application on the platform you choose. EC2 allows us to configure and scale our compute capacity easily to meet changing requirements and demand. It is integrated into Amazon's computing environment, allowing us to leverage the AWS suite of services.

Amazon S3

Amazon S3 provides secure, durable, and highly-scalable cloud storage for the objects that make up our application. Examples of objects we can store include source code, logs, images, videos, and other artifacts that are created when we deploy your application. Amazon S3 makes it is easy to use object storage with a simple web interface to store and retrieve our files from anywhere on the web, meaning that our website will be reliably available to the visitors.

Amazon Kinesis Analytics

Amazon Kinesis Analytics is the easiest way to process and analyze streaming data in real-time with ANSI standard SQL. It enables us to read data from Amazon Kinesis Streams and Amazon Kinesis Firehose, and build stream processing queries that filter, transform, and aggregate the data as it arrives. Amazon Kinesis Analytics automatically recognizes standard data formats, parses the data, and suggests a schema, which we can edit using the interactive schema editor. It provides an interactive SQL editor and stream processing templates so we can write sophisticated stream processing queries in just minutes. Amazon Kinesis Analytics runs our queries continuously, and writes the processed results to output destinations such as Amazon Kinesis Streams and Amazon Kinesis Firehose, which can deliver the data to Amazon S3, Amazon Redshift, and Amazon ES. Amazon Kinesis Analytics automatically provisions, deploys, and scales the resources required to run our queries.

Amazon Elasticsearch Service

Amazon ES is a popular open-source search and analytics engine for big data use cases such as log and click stream analysis. Amazon ES manages the capacity, scaling, patching, and administration of Elasticsearch clusters for us while giving the direct access to the Elasticsearch API.

3. Implementation:

Before the start of analyzing Apache access logs with Amazon Kinesis Analytics, we should be having AWS account and start an EC2 Instance

3.1 Starting an EC2 Intance

We are using an EC2 instance as the web server and log producer. Here we are using an existing EC2 instance launched from Amazon Linux AMI or we can also create new EC2 instance from Red Hat Enterprise

Linux . While creating we have choosen Amazon Linux AMI for the operating system. In this project we used t2.micro as it is available free-tier and is sufficient for this project.

We also want to ensure that our EC2 instance has an AWS Identity and Access Management (IAM) role configured with permission to write to Amazon Kinesis Firehose and Amazon CloudWatch. Once we launch the EC2 instance, we need to connect to it via SSH.

3.2 Generating Log files

Because Amazon Kinesis Analytics can analyze streaming data in near real-time so a live stream of Apache access log data is used. Our EC2 instance is not serving HTTP traffic as it is a new instance, so we generate continuous sample log files.

To create a continuous stream of log file data on the EC2 instance we downloaded, installed, and run the Fake Apache Log Generator from Github. This script generates a boatload of fake apache logs very quickly. It is useful for generating fake workloads for data ingest and/or analytics applications.

3.3 Creating Amazon Kinesis Firehose Delivery Stream

In Step 3.2, we created log files on our web server. Before they can be analyzed with Amazon Kinesis Analytics, the log data must first be loaded into AWS. Amazon Kinesis Firehose is a fully managed service for delivering real-time streaming data to destinations such as Amazon Simple Storage Service (Amazon S3), Amazon Redshift, or Amazon Elasticsearch Service (Amazon ES). In this step, we created an Amazon Kinesis Firehose delivery stream to save each log entry in Amazon S3 and to provide the log data to the Amazon Kinesis Analytics application.

3.4 Install and Configure the Amazon Kinesis Agent

Now that we have an Amazon Kinesis Firehose delivery stream ready to ingest our data, we configured the EC2 instance to send the data using the Amazon Kinesis Agent software. The agent is a stand-alone Java software application that offers an easy way to collect and send data to 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 our data in a reliable, timely, and simple manner. It also emits Amazon CloudWatch metrics to help us better monitor and troubleshoot the streaming process.
The Amazon Kinesis Agent can pre-process records from monitored files before sending them to our delivery stream. It has native support for Apache access log files, which we created in Step 3.2. When configured, the agent parse log files in the Apache Common Log format and convert each line in the file to JSON format before sending to our Firehose delivery stream.For converting tojso

```
{
 "cloudwatch.endpoint": "monitoring.us-east-1.amazonaws.com",
 "cloudwatch.emitMetrics": true,
 "firehose.endpoint": "firehose.us-east-1.amazonaws.com",
 "flows": [
  {
      "filePattern": "full-path-to-log-file",
      "deliveryStream": "name-of-delivery-stream",
      "dataProcessingOptions": [
      {
             "initialPostion": "START_OF_FILE",
             "maxBufferAgeMillis":"2000",
             "optionName": "LOGTOJSON",
             "logFormat": "COMBINEDAPACHELOG"
      }]
  }
 ]
}
```

3.5 Creating Amazon Elasticsearch Service Domain and create a second delivery stream

To create ES domain we need to configure the cluster, instance and storage based on the traffic, availability of our application.

Create second delivery stream for the storage of the data for this part we need to follow same steps as 3.3.

3.6 Creating an Amazon Kinesis Analytics Application

we are now ready to create the Amazon Kinesis Analytics application to aggregate data from our streaming web log data and store it in our Amazon ES domain. To create the Amazon Kinesis Analytics application configure the Source data for the Amazon Kinesis Analytics application and Connect to a source**.**

Amazon Kinesis Analytics will analyze the source data in our Firehose delivery stream and create a formatted sample of the input data for our review:

host VARCHAR(16)	datetime VARCHAR(32)	request VARCHAR(64)	response SMALLINT	bytes SMALLINT	referer VARCHAR(64)
153.233.179.68	29/Aug/2016:13:30:36 -0700	PUT /posts/posts/explore HTTP/1.0	200	5015	http://marquez.biz
28.65.158.143	29/Aug/2016:13:30:36 -0700	GET /apps/cart.jsp?appID=1303 HTTP/1.0	404	5040	http://vazquez.cor
79.217.236.155	29/Aug/2016:13:30:37 -0700	GET /explore HTTP/1.0	200	5037	http://www.johnso
120.18.76.166	29/Aug/2016:13:30:37 -0700	POST /wp-content HTTP/1.0	200	4945	http://taylor-clayto
98.153.196.4	29/Aug/2016:13:30:37 -0700	GET /posts/posts/explore HTTP/1.0	200	5005	http://www.ward-jc
192.182.180.218	29/Aug/2016:13:30:37 -0700	GET /list HTTP/1.0	200	5057	http://www.bradfo
59.139.143.190	29/Aug/2016:13:30:37 -0700	GET /app/main/posts HTTP/1.0	200	5040	http://www.king-ne
251.75.71.244	29/Aug/2016:13:30:38 -0700	GET /posts/posts/explore HTTP/1.0	200	4914	http://price.com/m
188.143.103.141	29/Aug/2016:13:30:38 -0700	POST /list HTTP/1.0	200	5094	http://gutierrez-an
103.54.46.71	29/Aug/2016:13:30:38 -0700	GET /search/tag/list HTTP/1.0	200	5127	http://www.walters

We get all data and we can also write queries for filtering the useful data. In kinesis firehose we will also have flexibility to write our queries directly in there and SQL code as follows

```
CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" (
    datetime VARCHAR(30),
    status INTEGER,
    statusCount INTEGER);

CREATE OR REPLACE PUMP "STREAM_PUMP" AS
    INSERT INTO "DESTINATION_SQL_STREAM"
        SELECT
            STREAM TIMESTAMP_TO_CHAR('yyyy-MM-dd''T''HH:mm:ss.SSS',
LOCALTIMESTAMP) as datetime,
            "response" as status,
            COUNT(*) AS statusCount
        FROM "SOURCE_SQL_STREAM_001"
        GROUP BY
            "response",
            FLOOR(("SOURCE_SQL_STREAM_001".ROWTIME - TIMESTAMP '1970-01-
01 00:00:00') minute / 1 TO MINUTE);
```

3.7 Viewing the Aggregated Streaming Data

After approximately 5 minutes, the output of the SQL statement in our Amazon Kinesis Analytics application will be written to our Amazon ES domain. Amazon ES has built-in support for Kibana, a tool that allows users to explore and visualize the data stored in an Elasticsearch cluster. To view the output of your Amazon Kinesis Analytics application we are using Kibana. It is an open source data visualization plugin for Elasticsearch. It provides visualization capabilities on top of the content indexed on an Elasticsearch cluster.

Kibana will automatically identify the DATETIME field in your input data, which contains time data
To visualize the data in our Elasticsearch index, you will create and configure a line chart that shows how many of each HTTP response type were included in the source web log data per minute.

4. Future Work & Conclusion

The project was done to quickly analyze the log data and classify it into different file categories for further use and we were successful in doing just that. The goal of the project was accomplished although the penultimate move would be to integrate the system in a real world application.

Log analysis is an integral part of the data management systems and hence we hope to improve our log analysis more and also implement the log analysis to digital marketing and other fields.

We tried it to use this analysis for state of mind analysis. For that we have taken the twitter data and refined the twitter data as it is from rest API we couldn't use that. This project requires streaming API which is generated from the infinite-log-generator

Log analysis in digital marketing helps companies create focus groups for customers and also helps to create customized advertisement's and product suggestions based on the user.

5. Advantages

- It's a fully managed service, so while your data is in the Kinesis stream, you don't have to worry about maintenance, storage, load balancing the streaming data
- You can incorporate this data with other AWS services. For example, you can easily integrate Kinesis with other AWS services like S3, Glacier and Redshift for long-term storage, or with EC2 instances
- You have practically unlimited data storage capabilities by leveraging services like Redshift & DynamoDB, even with hundreds of large staging tables.

References

[1] "Building log analytics," [Online]. Available: https://aws.amazon.com/getting-started/projects/build-log-analytics-solution/.

[2] "ManageEngine - Enterprise IT Management," [Online]. Available: https://www.manageengine.com/products/eventlog/application-log-processing.html.

[3] J. V. Mathew, "Overview of Amazon Web Services," January 2014.

[4] B. J. Jansen, Handbook of Research on Web Log Analysis, new york: Information Science Reference.

[5] ACM, *Communications of the ACM,* pp. 20-28, April 2010 .

[6] Y. Gupta, Kibana Essentials, PACKT Publishing, 2015.