|  |
| --- |
| Real-time analysis of user clickstream data from client’s website |

**Steps**

In this step it shows how the business problem is solved with utmost efficiency by using various techniques to reach the goal.

1. User clickstream logs on CollegeDunia’s website are captured.
2. These captured logs are restructured to an optimised format by using AWS lambda.
3. These restructured logs are stored in S3.
4. Amazon kinesis firehose service then streams these logs from S3 to amazon redshift.
5. Simultaneously, logging for these services is done in AWS cloudwatch.
6. Redshift is a data warehousing service that can hold large amount of data. This data can be analysed using high performance queries.

**Challenges Faced and resolved**

There were following issues with the architecture :

1. Setting up pipeline for clickstream log analysis was done using multiple AWS services

2. Storage of real-time logs by using s3

3. Optimized storage by using Redshift

4. Aggregation of non-uniform logs by using Kinesis Firehose

The issues were solved by restructuring the data through lambda and using redshift for data warehousing.

**Test Cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status (pass/fail)** |
| #1 | Log integrity check | Application logs | Identical | Identical | Pass |
| #2 | Connectivity check | Application logs | Connection successful | Connection successful | Pass |

Solution Summary - Training & Handover Notes

**About Challenge(s)**

Clickstream events are small pieces of data that are generated continuously with high speed and volume. In this business problem, they are generated by user actions and it is very useful to analyze them for data-driven decision making, influenced by the user behavior.

But capturing and processing data clickstream events in real time is difficult due to its large volume. The customer wanted to analyze the effectiveness of its new application features, website layout and marketing campaigns, in real time so that he can take actions faster.

For doing this the biggest challenge is to store metadata coming from collegedunia’s website, mainly the metadata about which links are clicked most, which link is seen for most of the time duration etc.

To cater to this need, the metadata is extracted from their website and is restructured to a specific format which ensures seamless data flow in the pipeline. Troubleshooting and root-cause analysis further identifies anomalies.

**Proposed Solution**

MIND discussed the problem with the customer and after analyzing the business problem, following steps were taken to deliver an efficient solution-

• This solution detects the user behavior on collegedunia website by analyzing the sequence of clicks that the user is making, the duration of time the user spends, where it usually begins to navigate, and how the user ends the session.

• By tracking this user behavior in real time, this solution helps to update recommendations, perform advanced A/B testing, push relevant notifications based on duration of session, and a lot more.

• As the number of users and web and mobile assets you have increases, so does the volume of data which is why this solution uses Amazon Kinesis which provides you with the capabilities necessary to ingest this data in real time and generate useful statistics immediately so to take necessary action.

• AWS Lambda service is used to fetch metadata from college dunia website and then process this data so that the output is restructured data, AWS S3 is used to store this restructured data.

• Lambdas are no doubt a powerful service but using them to directly ingest data is not the best approach so we have paired it with some other AWS services which also lowers down the cost of the pipeline.

• Firehose’s job is simply receiving data and pouring it at a target destination. There are options to choose between S3, RedShift, Lambdas, and Elasticsearch. Firehose, however, is an excellent fit for this scenario as its job is simply to receive the data and pour it at the destination location.

• Amazon Firehose streams the restructured data coming from S3 to Amazon Redshift.

• Then Amazon Redshift is used to run high performance queries on this streamed data with the customer then creates powerful reports and dashboards for quick insights.

**AWS Services used**

* Amazon Kinesis Data Firehose
* Amazon Redshift
* AWS Lambda
* Amazon S3

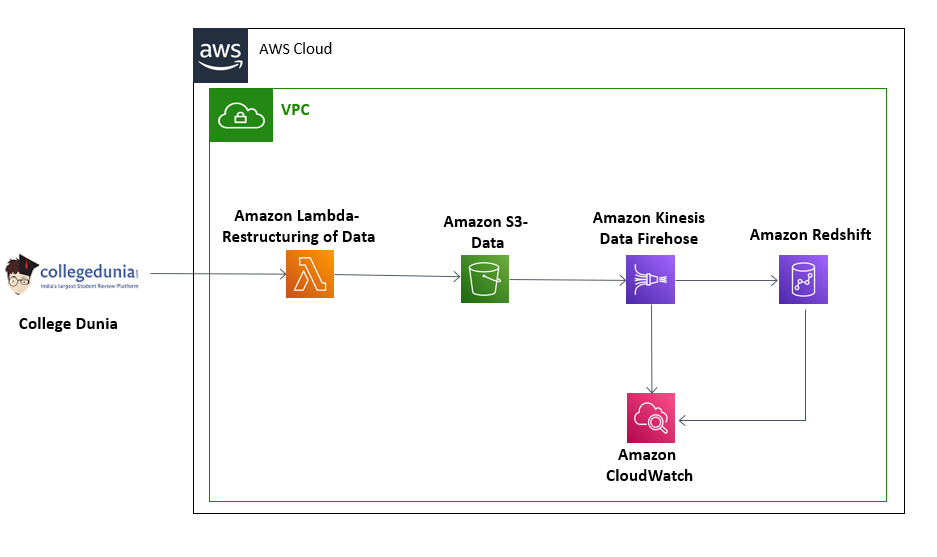
**Solution Outcome**

All in all, we successfully created a solution with the following outcomes.

* Full access to user clickstream data in 1 unified location.
* More precise business decisions.
* Increased reliability of infrastructure at lower cost & significant improvement in performance to cost ratio

There is increased customer satisfaction which possibly will be translated to more orders from same customer.

**Architecture Diagram**



**How AWS services helped in building the model for sales Forecasting**

**Amazon Kinesis Data Firehose**

Amazon Kinesis Data Firehose is the easiest way to load streaming data into data stores and analytics tools. It can capture, transform, and load streaming data into Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk, enabling near real-time analytics with existing business intelligence tools and dashboards that are already in use.

**Amazon RedShift**

Redshift makes it simple and cost effective to run high performance queries on petabytes of semi-structured and structured data so that you can build powerful reports and dashboards using QuickSight or other business intelligence tools.

**AWS Lambda**

It helped to initialize and validate the input and acted as the backend of the whole task. AWS Lambda lets us run code without provisioning or managing servers. Also, it helped to connect with various AWS API’s to acquire various insights from the inputs.

**Amazon S3 to store meta-data**

It is an object storage service that offers industry-leading scalability, data availability, security, and performance. In this solution it.