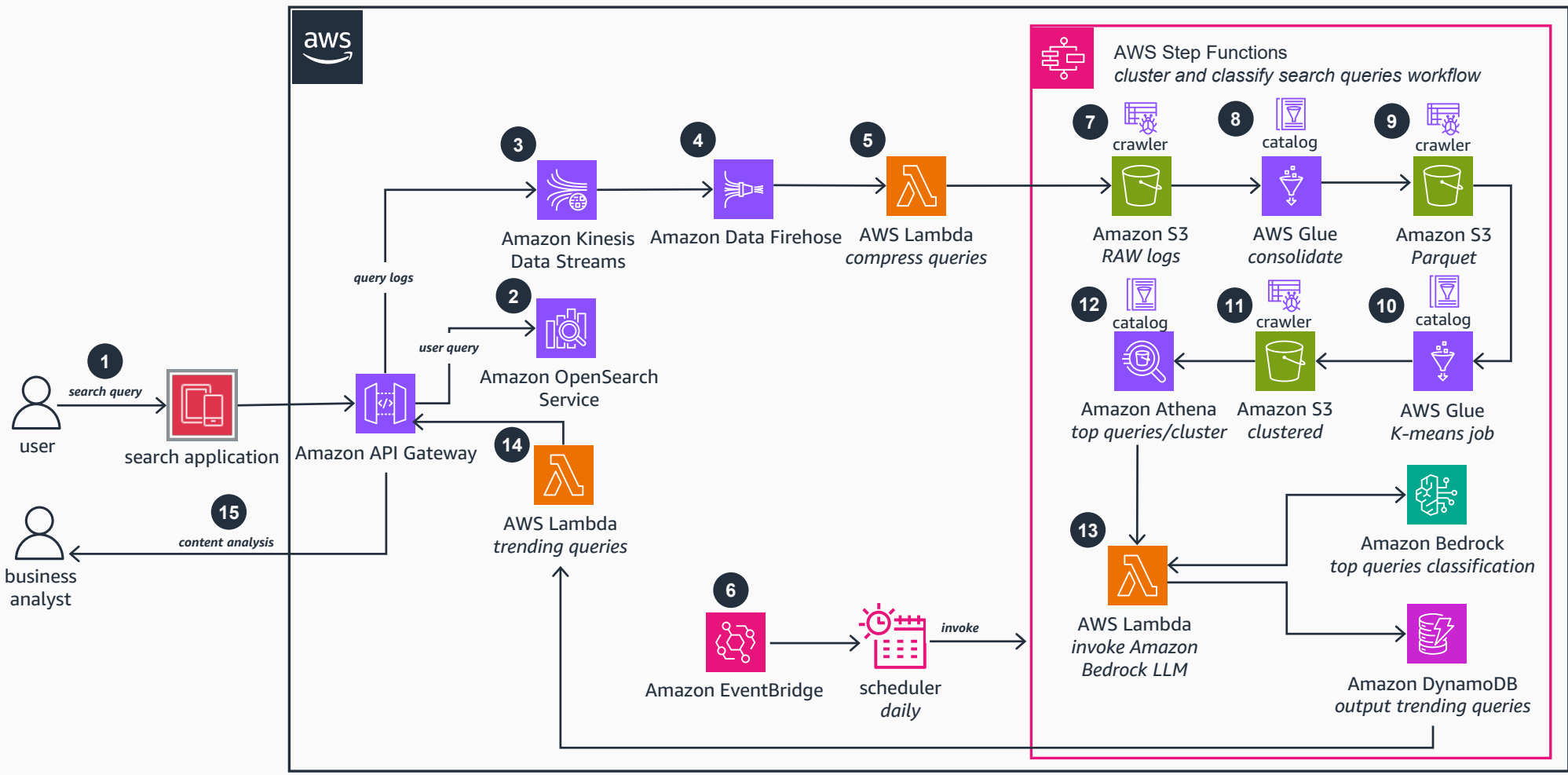


Amazon OpenSearch Trending Queries with AWS Glue and Amazon Bedrock

This reference architecture diagram demonstrates how to leverage AWS services like AWS Glue, Amazon Bedrock, Amazon OpenSearch Service, vector embedding, K-means clustering, and LLMs to identify top trending search queries for optimizing content strategy, improving user experience, and potentially increasing revenue.



- 1 End users search articles on the search page.
- 2 Queries are sent to **Amazon OpenSearch** for results retrieval.
- 3 Search query logs are streamed through **Amazon Kinesis Data Streams** using an **Amazon API Gateway** proxy.
- 4 **Amazon Kinesis Firehose** consolidates search query logs every 15 minutes (the maximum buffer limit).
- 5 An **AWS Lambda** function compresses search query logs for **Amazon Simple Storage Service** (Amazon S3) storage optimization.
- 6 **Amazon EventBridge**'s daily scheduler triggers **AWS Step Functions** for trending query identification.
- 7 An **AWS Glue** crawler creates catalogue tables for search query logs stored in **Amazon S3**.
- 8 An **AWS Glue** job consolidates and transforms query logs to Parquet files to boost query performance.
- 9 An **AWS Glue** crawler creates catalogue table for Parquet files stored in **Amazon S3**.
- 10 An **AWS Glue** job (K-means clustering) processes data, creates search query clusters based and stores it in **Amazon S3**.
- 11 An **AWS Glue** crawler creates catalogue table for search queries clusters stored in **Amazon S3**.
- 12 **Amazon Athena** queries the top *n* queries per cluster and passes a CSV file as input to an **AWS Lambda** function.
- 13 **Lambda** processes the CSV file in a loop, invokes **Amazon Bedrock** to identify the most relevant search query per cluster, and stores it in **Amazon DynamoDB**.
- 14 When the user opens the search page, application logic uses **Amazon API Gateway** to retrieve top trending queries using **AWS Lambda** and the **Amazon DynamoDB** table to display them on the search page.
- 15 Business analysts use a trending query API to analyze trending search queries to define content strategy.

