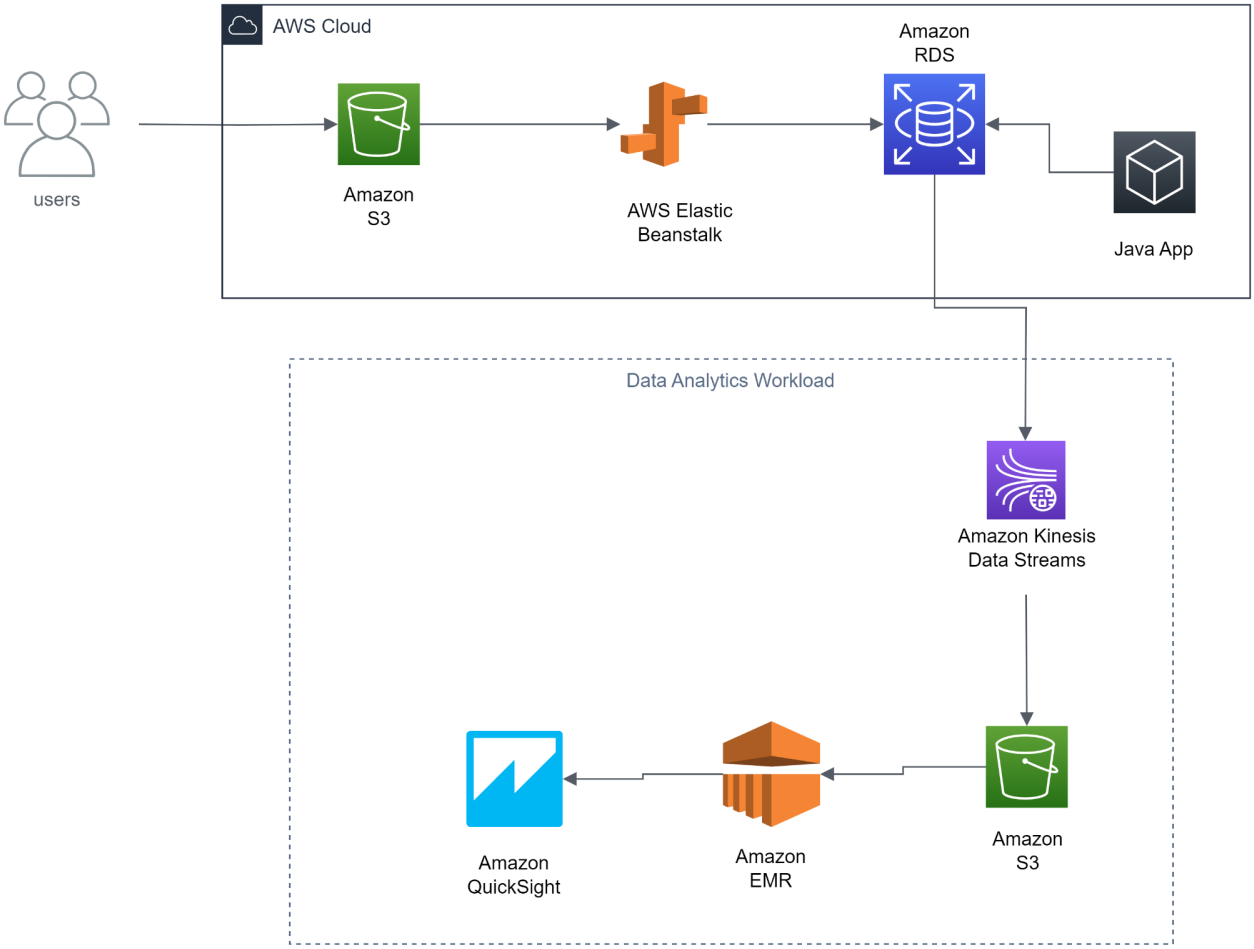
I would start off with decoupling the application layers. To do that I would house the frontend (HTML, CSS, JS) on Amazon S3, a highly scalable and durable object storage service. I would replace the Apache Web Server and Java application with AWS Elastic Beanstalk, a fully managed service for deploying and running applications. Deploying the Java application as an Elastic Beanstalk application will automatically handle the infrastructure provisioning, scaling and load balancing. And for the database, I would migrate the MySQL database to Amazon RDS, a managed database service. RDs will handle database administration tasks such as backups, patching and automatic software updates.

When it comes to the data analytics workload, for ingestion I will use Amazon Kinesis Data Streams to ingest the massive amount of data from on-premises sources. Kinesis Data Streams can handle real-time streaming data. For storage, I will store the data in Amazon S3, a scalable and cost-effective object storage service. You can use S3's durability, availability, and scalability to handle large volumes of data for analytics. For processing, I will spin up an Amazon EMR (Elastic MapReduce) cluster to run Apache Hadoop for data processing and analytics. EMR simplifies the setup, configuration, and management of Hadoop clusters. And for visualization, I will utilize Amazon QuickSight, a fully managed business intelligence service, for data visualization and deriving insights from the analyzed data. QuickSight integrates with various data sources, including Amazon S3, to create interactive dashboards and reports.

The reason I went with AWS Elastic Beanstalk is that it simplifies application deployment and management, allowing for easy scaling and handling of backend infrastructure. Amazon RDS provides managed database services, offloading administrative tasks and ensuring high availability and durability. Amazon S3 offers scalable object storage for hosting the frontend, storing ingested data, and integrating with other services in the architecture. Amazon Kinesis Data Streams enables real-time data ingestion and processing for streaming use cases. Amazon EMR simplifies the deployment and management of Apache Hadoop clusters for data analytics and processing. And I chose Amazon QuickSight because it provides an intuitive and interactive interface for visualizing and analyzing the processed data.


AWS Cloud
users
Amazon
S3
AWS Elastic
Beanstalk
Amazon
RDS
Java App
Data Analytics Workload
Amazon Kinesis
Data Streams
Amazon
QuickSight
Amazon
EMR
Amazon
S3