

Project Statement:

As a client with a social media app, I require a comprehensive solution architecture for cloud-based data management.

The goal is to optimize the app's data storage, processing, and retrieval capabilities, ensuring scalability, reliability, and security.

The solution should leverage cloud technologies and services to enable efficient data handling, analysis, and integration with other systems.

The architecture should address data governance, data privacy, and compliance requirements, while also considering performance optimization and cost-effectiveness.

Ultimately, the aim is to enhance the overall user experience, streamline data workflows, and enable future growth and innovation within the app.

What you need to do:

- i. Select a cloud platform for the project, providing a reason for the specific choice made.
- ii. Using the chosen cloud platform, devise a comprehensive end-to-end solution for the project, including recommendations for storage services, ETL (Extract, Transform, Load) processes, security measures, visualization tools, and more.
- iii. Justify the selection of these services over alternatives, highlighting their superior attributes and benefits.
- iv. Conduct a thorough cost analysis for the entire project, including a breakdown of expenses and the estimated duration of the project.

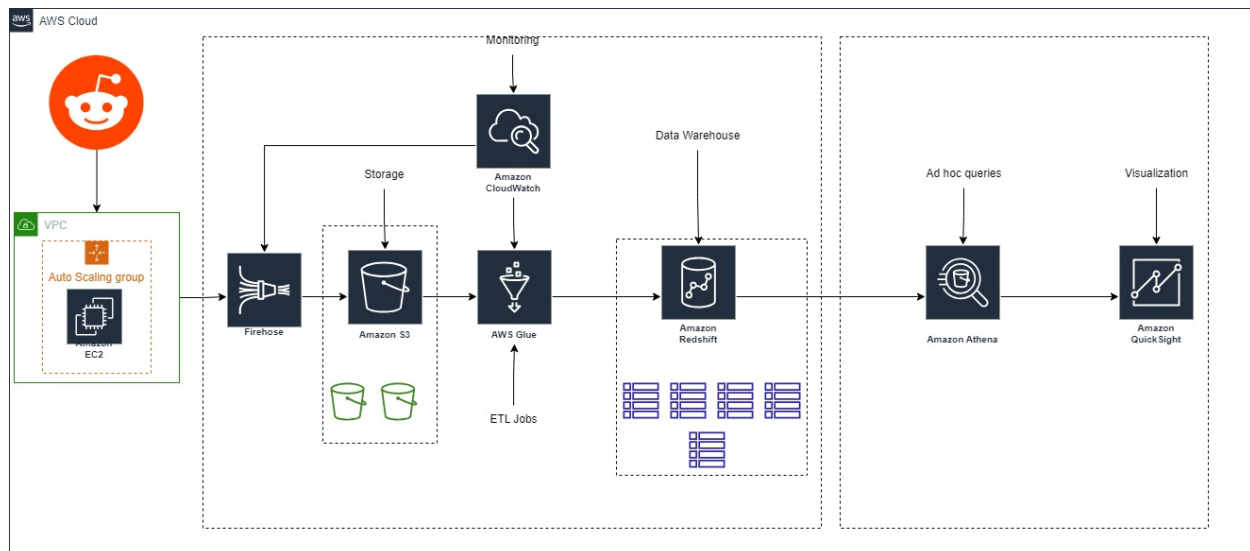
Solution

SELECT A CLOUD PLATFORM FOR THE PROJECT, PROVIDING A REASON FOR THE SPECIFIC CHOICE MADE.

As a Cloud Data Engineer, I recommend AWS for the deployment of social media app because of its feature-rich and simple to use, and the service comes with extensive, informative documentation. Azure & GCP organizes all your account details and stores them in one place. However, its documentation is more challenging to understand and locate. AWS is hosting in multiple locations worldwide. Azure and GCP are also hosting in multiple locations worldwide, but the difference occurs in the number of their respective availability zones. Both AWS, Azure & GCP use subnets to segregate networks. However, AWS offers far more customizations. For instance, AWS VPC provides private and public subnets. This design allows you to run public-facing web applications while keeping the back-end servers private. AWS VPC is also incredibly flexible and allows you to configure the virtual private cloud exactly how you want. In addition, you get access to tools like programmable APIs, CLIs, Cloud Formation Templates, and a management portal to customize your VPC architecture. By default, Azure VNet assigns internet access to all connected resources. Therefore, you don't get the private vs. public segregated networks that AWS provides. However, Azure VNet does provide a management portal, CLI, and PowerShell to help you create and customize your network architectures. But, again, you don't get as many options as AWS VPC provides. In the long run, Azure VNet seems more enterprise-focused, while AWS VPC is more customer-facing.



USING THE CHOSEN CLOUD PLATFORM, DEVISE A COMPREHENSIVE END-TO-END SOLUTION FOR THE PROJECT, INCLUDING RECOMMENDATIONS FOR STORAGE SERVICES, ETL (EXTRACT, TRANSFORM, LOAD) PROCESSES, SECURITY MEASURES, VISUALIZATION TOOLS, AND MORE.



Above is the end-to-end solution architecture diagram for social media app.

With a social media app, we are collecting data in a streaming format for that we need **EC2** machine where we can run Kafka to gather real time data for every record happened also, we need **AWS kinesis data firehose** for performing ETL on streaming data and placed them in **S3** buckets. We set VPCs that come with built-in security features. You can define network access control lists (ACLs) and security groups to control *inbound and outbound traffic to your resources*. This allows you to enforce *fine-grained security policies and restrict access based on protocols, ports, and IP addresses*. VPCs support the scalability and elasticity of your applications. You can easily *scale your VPC* by adding or removing resources as needed. This flexibility allows you to accommodate changes in your application demands without disruption. AWS services, such as EC2 instances, RDS databases, and Lambda functions, can be provisioned within your VPC. This enables you to build *highly available and fault-tolerant applications* that utilize other AWS services while *keeping your network traffic within your VPC*. Then we store all the raw data in the s3 bucket where we have millions of data generated every day. S3 provides fine-grained access control to buckets and objects. Access control can be managed using *AWS Identity and Access Management (IAM) policies, bucket policies, and Access Control Lists (ACLs)*. These mechanisms allow you to control who can perform actions on your S3 resources and what actions they can perform. S3 allows you to define bucket policies to control access at the bucket level. Bucket policies are written in JSON format and can be used to enforce *specific security requirements, such as restricting access to certain IP addresses or allowing access only over SSL*. Then for transformation purpose we used **AWS Glue** for inferring the right schema and changing the data types according to its nature. AWS Glue allows you to create ETL jobs using Apache Spark. These jobs can be used to transform and cleanse the data before loading it into a target destination,

such as a data warehouse or another data lake. Glue provides a visual interface for designing ETL workflows, or you can write custom code in Python or Scala. *AWS Glue integrates with AWS Identity and Access Management (IAM)*, which allows you to manage access to Glue resources. IAM enables you to define *fine-grained access policies and control* who can perform actions on Glue resources, such as running crawlers or executing ETL jobs. AWS Glue has a strong focus on data cataloging, making it easier to discover and manage data assets. ADF and Dataflow also support data cataloging but may require additional setup and integration with other Azure or Google services. Then we put all our structured data into **AWS Redshift** where we have fact and dimension tables. AWS Redshift follows a *columnar storage format*, where data is stored column-wise rather than row-wise. This structure allows for efficient compression, improved query performance, and optimized storage utilization. Redshift organizes data into clusters, which consist of a leader node and one or more computer nodes. The leader node handles query planning and coordination, while the compute nodes perform the actual data processing. Redshift allows you to configure Virtual Private Cloud (VPC) endpoints to access your clusters securely within your private network. *This helps protect against unauthorized access from the internet.* Redshift also supports encryption of data transferred between Redshift and client applications using SSL/TLS protocols. Redshift uses a columnar storage format, while Azure Synapse Analytics and Big Query use a combination of columnar and row-based storage. This difference can impact the performance and storage efficiency of queries. After that we can write ad hoc sql queries for that we can use **AWS Athena**. AWS Athena is based on Presto, an open-source distributed SQL query engine. Athena uses a serverless architecture, which means that there are no infrastructure management tasks involved. When you execute a query in Athena, it dynamically provisions resources to process the query and scales them up or down based on the workload. Athena leverages a metadata catalog, which holds the schema and location information of the data stored in S3. *Athena supports data partitioning, allowing you to organize your data in S3 based on specific columns.* Partitioning improves query performance by limiting the amount of data scanned. You can set up partitioning based on time, location, or any other relevant column in your data. Also, for the visualization we can use aws native cloud service which **AWS QuickSight**. QuickSight offers built-in data preparation capabilities that allow users to cleanse, transform, and enrich their data before visualization. This includes features like data cleansing, filtering, joining, and aggregating data from different sources. QuickSight provides a drag-and-drop interface to create interactive visualizations, charts, graphs, and dashboards. Users can customize the appearance, apply filters, and create drill-down capabilities to explore data in a user-friendly manner. QuickSight allows you to set up *data-level permissions to control access to specific data sources or tables*. This ensures that users can only view and analyze the data they have been granted access to.

Justify the selection of these services over alternatives, highlighting their superior attributes and benefits.

The selection of the following AWS services in your social media app architecture offers several superior attributes and benefits, which I will explain below:

EC2 (Elastic Compute Cloud) Instance: EC2 instances are highly scalable virtual servers in the cloud. By using EC2, you can easily provision and manage the compute capacity required for your social media app. EC2 provides flexibility, allowing you to choose the instance types and sizes that best fit your application's needs. It also offers a wide range of operating system options and allows for easy integration with other AWS services.

Alternatives: Other cloud providers also offer compute services, such as Google Cloud Compute Engine and Microsoft Azure Virtual Machines. However, by selecting EC2, you benefit from AWS's vast ecosystem, extensive documentation, and industry-leading experience in cloud computing.

AWS Kinesis Data Firehose:

AWS Kinesis Data Firehose is a fully managed service provided by Amazon Web Services (AWS) that enables you to easily capture, transform, and load streaming data into storage and analytics services. It simplifies the process of ingesting and delivering real-time data streams from various sources, such as application logs, website clickstreams, sensor data, social media feeds, and more, to AWS storage and analytics services.

Alternatives: AWS Kinesis Data Firehose is designed to handle high-volume, real-time streaming data at scale. It automatically scales to accommodate increased data throughput and ensures durability and fault tolerance. It also offers various buffering and compression options to optimize data delivery. While the alternatives in Azure and GCP are also scalable, AWS Kinesis Data Firehose has proven its capabilities in handling large-scale streaming workloads.

S3 (Simple Storage Service): S3 is an object storage service designed for storing and retrieving large amounts of data. It provides high durability, availability, and scalability. S3 is an ideal choice for storing user-generated content, such as images, videos, and other media files in your social media app. Its simple API allows for easy integration with your application, and it supports various storage classes to optimize costs based on data access patterns.

Alternatives: Alternatives to S3 include Google Cloud Storage and Microsoft Azure Blob Storage. However, S3's popularity, durability, and cost-effectiveness make it the preferred choice for many developers and enterprises.

AWS Glue: AWS Glue is a fully managed extract, transform, and load (ETL) service. It allows you to prepare and transform data from various sources for analytics purposes. In your social media

app architecture, AWS Glue can be used to extract and transform data from different data sources, perform data cleansing and enrichment, and load the data into your data warehouse.

Alternatives: Alternative ETL services include Google Cloud Dataflow and Microsoft Azure Data Factory. However, AWS Glue stands out due to its seamless integration with other AWS services, such as S3 and Redshift, and its ability to automate much of the ETL process.

AWS Redshift: AWS Redshift is a fully managed data warehousing service. It is optimized for online analytical processing (OLAP) workloads and can handle large volumes of data with high performance. Redshift is an excellent choice for analyzing and querying the data collected from your social media app. It provides columnar storage, advanced compression techniques, and parallel query execution, enabling fast and efficient data analysis.

Alternatives: Competing data warehousing solutions include Google BigQuery and Microsoft Azure Synapse Analytics. However, Redshift's integration with other AWS services, its cost-effectiveness, and its performance make it a popular choice for data warehousing.

AWS Athena: AWS Athena is an interactive query service that allows you to analyze data stored in S3 using standard SQL queries. It is a serverless service, meaning you don't have to provision or manage any infrastructure. Athena is ideal for ad-hoc queries and exploratory data analysis in your social media app. It enables you to query large amounts of data without the need to set up and manage a traditional database infrastructure.

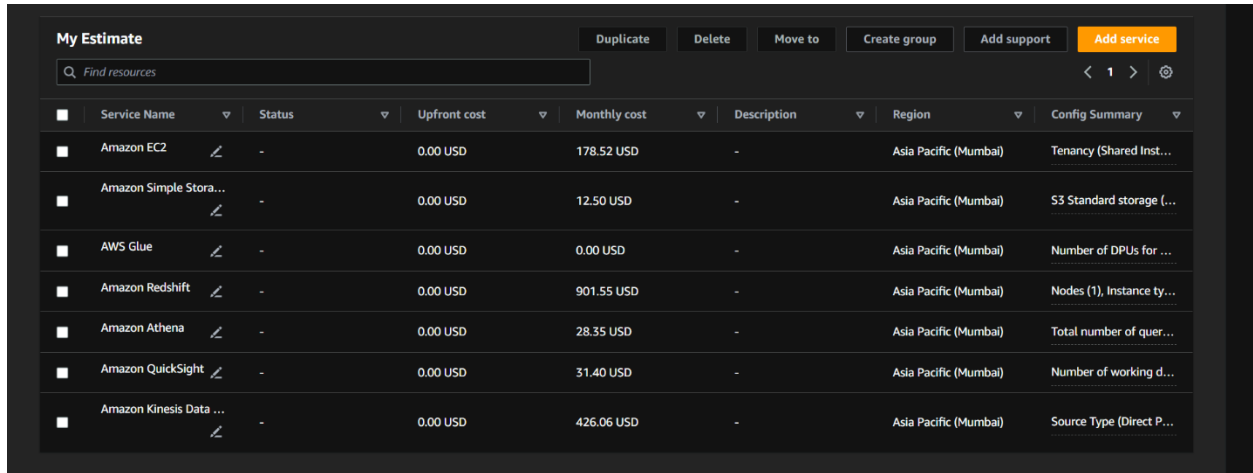
Alternatives: Google BigQuery and Azure Data Lake Analytics are alternative serverless query services. However, Athena's tight integration with other AWS services, its compatibility with SQL, and its pay-per-query pricing model contribute to its appeal.

AWS QuickSight: AWS QuickSight is a business intelligence (BI) service that allows you to visualize and gain insights from your data. With QuickSight, you can create interactive dashboards, perform ad-hoc analysis, and share visualizations with others. In your social media app architecture, QuickSight can help you monitor key metrics, track user engagement, and make data-driven decisions.

Alternatives: Alternative BI services include Google Data Studio and Microsoft Power BI. However, QuickSight's integration with other AWS services, its ease of use, and its cost-effectiveness make it an attractive choice.

Conduct a thorough cost analysis for the entire project, including a breakdown of expenses and the estimated duration of the project.

Infrastructure Costs:



The screenshot shows the 'My Estimate' interface in AWS Cost Explorer. It features a search bar, action buttons (Duplicate, Delete, Move to, Create group, Add support, Add service), and a table of services. The table columns are: Service Name, Status, Upfront cost, Monthly cost, Description, Region, and Config Summary. The services listed are Amazon EC2, Amazon Simple Storage Service, AWS Glue, Amazon Redshift, Amazon Athena, Amazon QuickSight, and Amazon Kinesis Data Firehose.

Service Name	Status	Upfront cost	Monthly cost	Description	Region	Config Summary
Amazon EC2	-	0.00 USD	178.52 USD	-	Asia Pacific (Mumbai)	Tenancy (Shared Inst...
Amazon Simple Stora...	-	0.00 USD	12.50 USD	-	Asia Pacific (Mumbai)	S3 Standard storage (...)
AWS Glue	-	0.00 USD	0.00 USD	-	Asia Pacific (Mumbai)	Number of DPUs for ...
Amazon Redshift	-	0.00 USD	901.55 USD	-	Asia Pacific (Mumbai)	Nodes (1), Instance ty...
Amazon Athena	-	0.00 USD	28.35 USD	-	Asia Pacific (Mumbai)	Total number of quer...
Amazon QuickSight	-	0.00 USD	31.40 USD	-	Asia Pacific (Mumbai)	Number of working d...
Amazon Kinesis Data ...	-	0.00 USD	426.06 USD	-	Asia Pacific (Mumbai)	Source Type (Direct P...

EC2 Instance: The cost of EC2 instances depends on factors such as instance type, size, and duration of usage.

AWS Kinesis Data Firehose: AWS Kinesis Data Firehose cost depends on many factors but if we run on its minimum configuration which is enough for initialing launching the app on the cloud for achieving basic functionality.

S3 Storage: S3 pricing is based on data storage, data transfer, and API requests. Consider the amount of data you expect to store and transfer in S3, as well as the expected number of API requests.

AWS Glue: AWS Glue pricing is based on the number of data processing units (DPUs) used and the duration of job runs. Consider the complexity and scale of your data transformations and the frequency of job runs.

Redshift: Redshift pricing is based on factors such as cluster type, node size, and usage duration. Determine the appropriate cluster configuration based on your data volume and query performance requirements.

Athena: Athena pricing is based on the amount of data scanned during queries. Consider the expected data volume and query patterns to estimate the costs.

QuickSight: QuickSight pricing is based on factors such as user types, data refreshes, and SPICE (Super-fast, Parallel, In-memory Calculation Engine) capacity. Estimate the number of users and the frequency of data refreshes for cost calculation.

Data Transfer Costs:

Consider the volume of data transferred between services within your architecture, such as data transfers between EC2 and S3, Glue and Redshift, and S3 and Athena. AWS provides free data transfer allowances within the same AWS Region, but data transfer costs may apply for cross-region or external data transfers.

Data Processing Costs:

Consider the complexity and frequency of data processing tasks performed by Glue, such as data extraction, transformation, and loading. Consider the number of DPUs required and the duration of job runs.

Project Duration:

The estimated duration of the project will impact the overall cost. Longer project durations will incur additional costs for the ongoing usage of AWS services. Below image will tell you the rough estimation cost of the project.

