Analytics

Đà Nẵng, năm COVID thứ 2

Kinesis Stream family:

- Kinesis Data Streams: Build custom applications that analyze data streams using popular stream-processing frameworks.
- Kinesis Data Firehose: Load data streams into data stores.
- Kinesis Data Analytics: Process and analyze streaming data using SQL or Java.
- Kinesis Video Streams: Capture, process, and store video streams for analytics and machine learning.

Kinesis Data Streams:

- A Kinesis data stream is an ORDERED sequence of data records meant to be written to and read from in real time.
- The producers continually push data to Kinesis Data Streams, and the consumers process the data in real time.
- The delay between the time a record is put into the stream and the time it can be retrieved (put-to-get delay) is typically less than 1 second.
- Shards:
  - The data records in a data stream are distributed into shards.
  - Each shard provides a processing capacity.
  - You are charged on a per-shard basis.
  - A single shard can ingest up to 1 MB of data/s or 1,000 records/s for writes.
  - Each shard can support up to 5 read transactions/s, up to a maximum total data read rate of 2 MB per second.
  - No auto scaling of the number of shards.

- Records:
  - Maximum size of data payload (Blob) of a record is 1 MB.
  - Each data record has a sequence number (assigned by the stream).
  - Data records are distributed into shards based on the Partition Key.
  - The Partition Key is specified by the applications putting the data into a stream.
- Records Retention:
  - Retention period: 24 hours by default, up to 365 days.
  - When a shard is removed while having records, it stays in a "closed" state until the retention period ends.
- Producers:
  - Push records into streams using partition keys.

- Consumers:
  - Consumers get records in a Pull model over HTTP using GetRecords.
  - By default, a shard provides 2 MB/sec of read throughput per shard that is shared across all the consumers that are reading from that shard.
  - When you configure Lambda as a Consumer, the Lambda service will use one function concurrency per open shard.
- Enhanced fan-out consumers:
  - With this feature, each consumer gets its own 2 MB/sec allotment of read throughput, without contending for read throughput with other consumers.
  - Kinesis Data Streams can push the records to you over HTTP/2 using SubscribeToShard.
  - There is a data retrieval cost and a consumer-shard hour cost.
- If you need to send stream records directly to services such as S3, Redshift or Splunk, it's better to use Kinesis Data Firehose.
- Supports Server Side Encryption using AWS KMS.

Kinesis Data Firehose:

- A fully managed service for delivering real-time streaming data to delivery streams for archiving and analysis purposes.
- Sources can be:
  - Your application pushing directly data into the delivery stream.
  - AWS services like AWS IoT, CloudWatch Logs and CloudWatch Events.
  - Kinesis Data Stream.
- Destinations include Amazon S3, Redshift, Elasticsearch, Splunk, and any custom HTTP endpoint.
- Data Processing:
  - You can also configure Kinesis Data Firehose to transform your data before delivering it.
  - Transformation is done using Lambda functions.
  - Supports a Lambda invocation time of up to 5 minutes.
  - You can optionally convert data formats.
  - You can optionally store both original data and transformed data to S3.

- Supports records up to 1 MB.
- Stores data for up to 24 hours in case of delivery failure.
- Supports Server Side Encryption using AWS KMS.

Kinesis Data Analytics:

- Service to Process and analyze streaming data using SQL or Apache Flink.
- The service enables you to quickly author and run powerful SQL or Apache Flink code against streaming sources to perform time series analytics, feed real-time dashboards, and create real-time metrics.
- Use cases:
  - Generate time-series analytics – You can calculate metrics over time windows, and then stream values to S3 or Redshift through a Kinesis data delivery stream.
  - Feed real-time dashboards – You can send aggregated and processed streaming data results downstream to feed real-time dashboards.
  - Create real-time metrics – You can create custom metrics and triggers for use in real-time monitoring, notifications, and alarms.
- Both the Streaming Source and the External destination can be either a Kinesis data stream or a Kinesis Data Firehose data delivery stream.
- You can optionally configure a reference data source to enrich your input data stream within the application. This data is loaded in a "reference table" inside the application.

- Application code:
  - A series of SQL statements that process input and produce output.
  - You can write SQL statements against in-application streams and reference tables. You can also write JOIN queries to combine data from both of these sources.
- Data journey: Streaming Sources ==> In-application input streams ==> Application Code ==> In-application output streams ==> External destinations.

Athena:

- an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL.
- Athena helps you analyze unstructured, semi-structured, and structured data stored in Amazon S3. Examples include CSV, JSON, or columnar data formats such as Apache Parquet and Apache ORC.
- Can be used to search logs in S3.
- How it works:
  - You specify the structure of your file and where the fields are. You can use the Data Catalog feature of AWS Glue to extract automatically this structure.
  - You submit a SQL request to Athena.
  - Athena instantiates a swarm of workers that read your files to extract the fields.
  - The SQL request is run against the collected fields.
- Serverless.
- Pay per query.
- Supports JDBC connections.
- Athena integrates with Amazon QuickSight for easy data visualization.

Athena vs Redshift Spectrum:

- The basic functionality offered by Athena and Redshift Spectrum is similar: querying S3 using standard SQL, and storing the results of that query.
- The main differences are:
  - Resource provisioning: While both are serverless, Athena relies on pooled resources provided by AWS, whereas Spectrum resources are allocated according to your Redshift cluster size. You have therefore more control over performance with Redshift Spectrum.
  - Loading data into Redshift: Athena stores query results on S3, and they can be loaded into Redshift from there; whereas Spectrum can be used to join tables stored on Redshift directly.

- A serverless data integration service that makes it easy to discover, prepare, and combine data for analytics, machine learning, and application development.
- Provides both visual and code-based interfaces to make data integration easier.
- ETL (extract, transform, and load) developers can visually create, run, and monitor ETL workflows with a few clicks in AWS Glue Studio.
- AWS Glue generates the code that’s required to transform your data from source to target.
- Data Sources: S3, RDS, DynamoDB, any JDBC DB, MongoDB.
- Data Targets: S3, RDS, any JDBC DB, MongoDB.

Amazon Elasticsearch Service (Amazon ES):

- A managed service to deploy, operate, and scale Elasticsearch clusters in the AWS Cloud.

AWS AppSync:

- Provides a robust, scalable GraphQL interface for application developers to combine data from multiple sources, including Amazon DynamoDB, AWS Lambda, and HTTP REST APIs.
- Integration with Amazon Cognito user pools for fine-grained access control at a per-field level.