Building a Data Management Architecture Using Amazon Web Services

Published 28 November 2022 - ID G00778151 - 76 min read

By Analyst(s): Mayank Talwar, Zain Khan

Initiatives: Data Management Solutions for Technical Professionals; Analytics and Artificial Intelligence for Technical Professionals; Evolve Technology and Process Capabilities to Support D&A

AWS provides comprehensive data management capabilities to help organizations build, scale and manage data and analytics workloads. Data and analytics technical professionals can use this research to select the best solution and product combination to build their data management platform on AWS.

Overview

Key Findings

- AWS provides a mature platform with proven stability, scale and performance of services. It has demonstrated over the years that it can operate its data and analytics tools while adhering to published SLAs.
- AWS has introduced serverless and federation features to ease setup, maintenance and usage in data ingestion, processing and analysis, thereby opening up the services to a wider range of data and analytics professionals.
- AWS has a broad range of services and appeals to customers across many industries. The breadth and number of possible combinations of offerings from AWS can be daunting, so keep up with new announcements and updates to get the most from your AWS investment.

Recommendations

Data and analytics technical professionals responsible for modernizing their data management solutions should evaluate AWS-based solutions to:

- Use AWS Glue, CloudWatch and Lake Formation to cover data discovery, data ingestion, orchestration, transformation, cataloging, monitoring and security.

- Choose available native tools from the ecosystem carefully, especially data integration and data ingestion. Each tool provides a pattern that is best fit for a certain set of use cases, with overlap.
- Supplement native tools and capabilities with marketplace or community products, especially for advanced architectures like data fabric or data mesh.
- Embrace AWS' serverless data offerings for unpredictable workloads. Be familiar with the cost controls and features AWS provides for continuous cost monitoring when using serverless technologies.

Analysis

Amazon Web Services (AWS) continues to offer a wide set of data services to implement a comprehensive, end-to-end data management platform. These services range in maturity from newly released to enterprise-ready, with nearly all of them releasing features and capabilities every quarter. AWS cloud platform offers competitive data ingestion, storage, processing and analytics offerings with which enterprises can innovate for different use cases across a wide spectrum of data workloads.

There are three architectural patterns of cloud data ecosystems: full cloud service provider (CSP)-native, blended and as-is. No matter what position one takes at the start, they are likely to end up in a blended state (a combination of CSP-native and independent software vendor [ISV] services). This note, however, is focused on the "full CSP-native" approach to the cloud data ecosystem on AWS cloud.

It's important to not build in the cloud, but build for the cloud. This is especially useful for those moving from the on-premises solutions. "Forklifting" an existing data and analytics stack "as is" from the data center to the cloud can result in some cost saving and higher business agility, yet it may leave organizations with a stagnant solution hosted in a different place. To truly improve your solutions, costs and results, consider your architecture carefully, use managed services where possible, and iterate on your design to gain efficiency and to lower costs. Assess serverless options for unpredictable workloads but monitor the cost and define proper cost controls.

The cloud is not an escape valve. Even with the cloud, it's still an organization's responsibility to put together all the components and make it work in an automated, predictable and scalable manner.

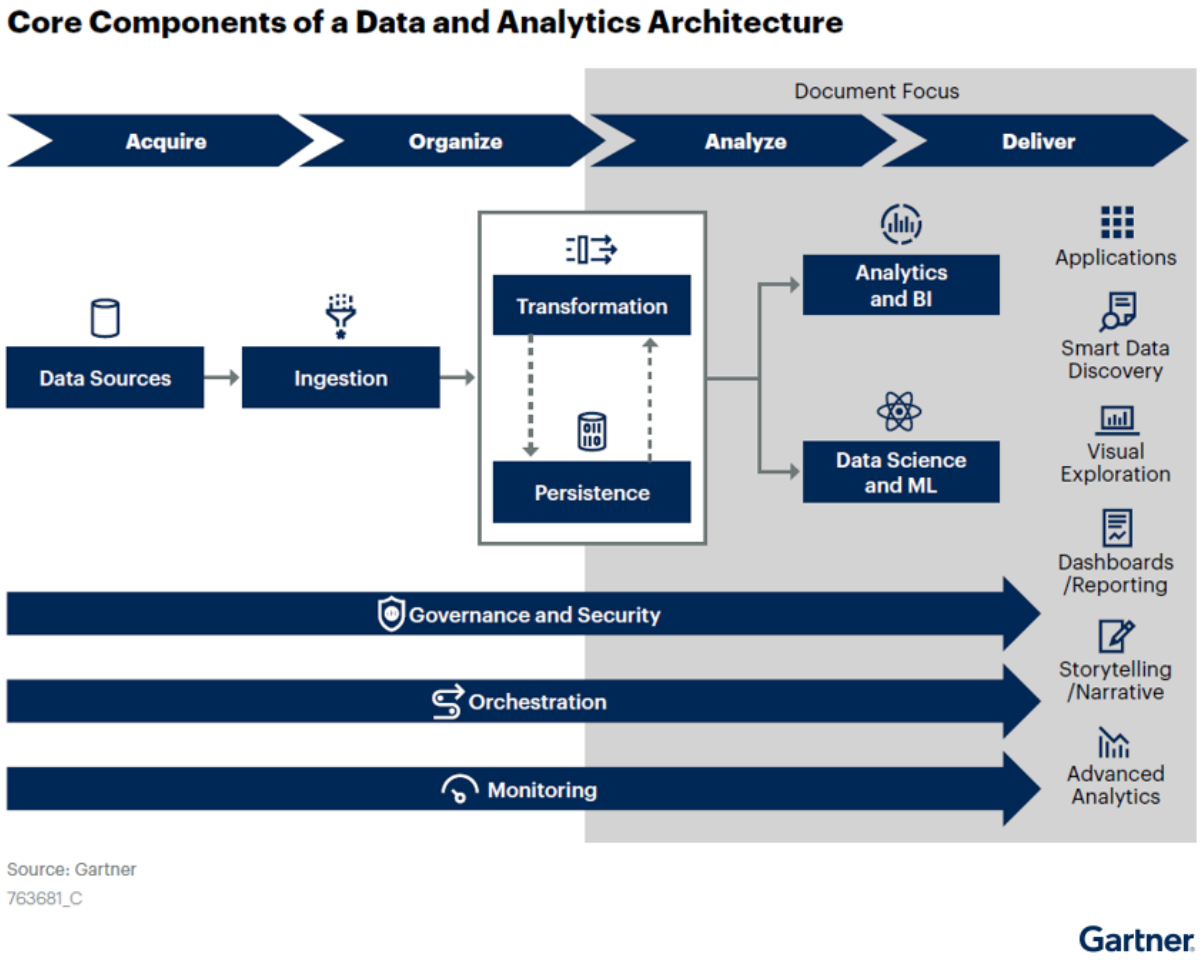
This research explores how well the AWS platform can fulfill the data and analytics requirements of today's enterprises. It highlights various components of AWS and their functionality, as often each category of the pipeline has multiple offerings. In order to do so, Gartner has disaggregated the data and analytics pipeline into four broad sections as shown in Figure 1. These include:

- Acquisition of the data via data integration techniques
- Organization of the data for downstream applications via data transformation and persistence strategies
- Analyzing the data via descriptive, predictive or prescriptive analytics techniques
- Delivery of analytics and exploration-ready curated data via a delivery layer

These **building blocks** are complemented with security, governance, orchestration and monitoring layers. This document explores various architectural patterns and corresponding native product offerings within AWS for the acquisition, organization, persistence and analysis of data for disparate analytics and operational use cases.

Download All Graphics in This Material

Figure 1: Core Components of D&A Architecture



This research provides updates on new and enhanced capabilities within the data management area of the AWS ecosystem as illustrated in Figure 2. Refer to Building an Analytics and AI Architecture Using AWS for the latest offerings from AWS in the analytics and AI space.

Figure 2: Sample Data Management Architecture in AWS

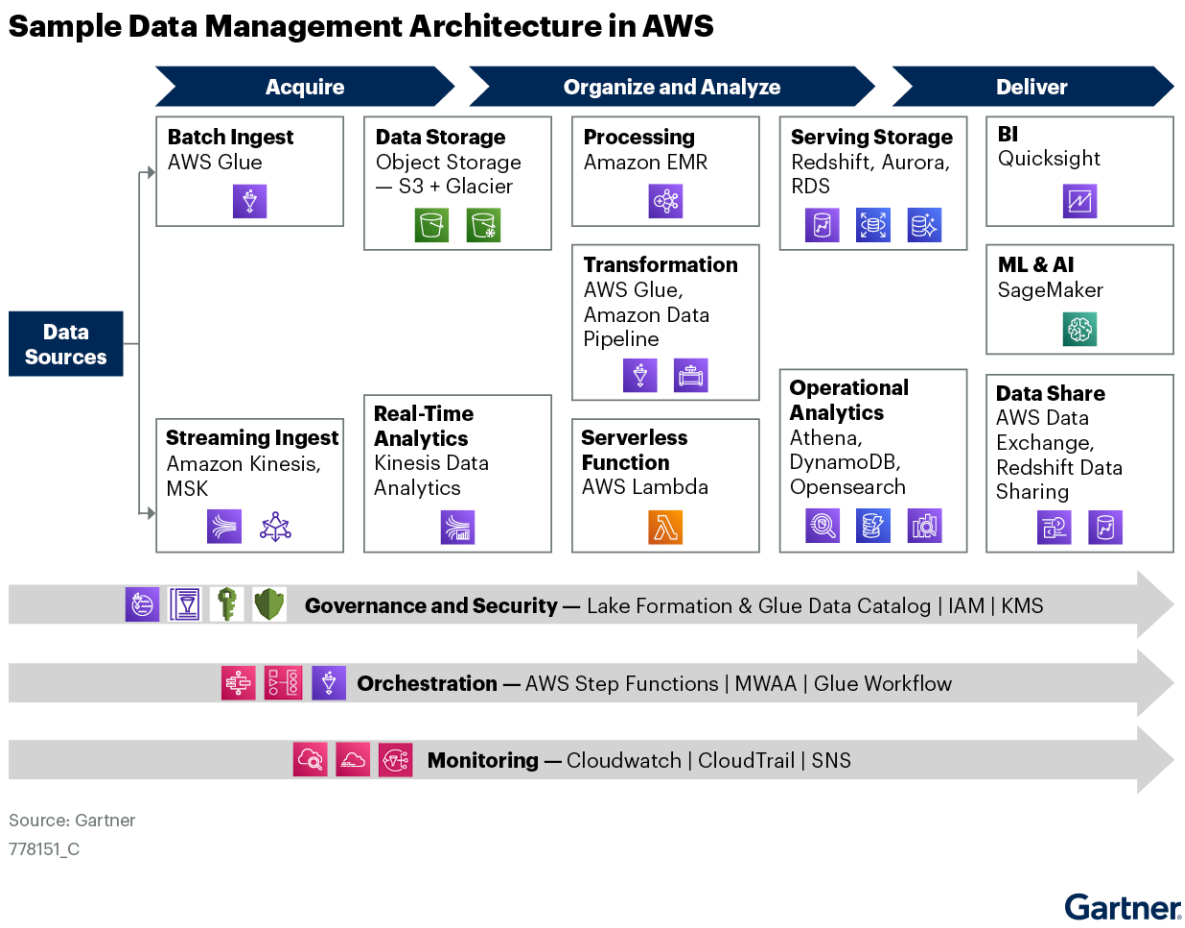


Figure 2 highlights an architecture pattern known as Lambda. The diagram shows sample components of AWS data and analytical offerings that can be chosen to deliver an analytical solution and is not a full representation of AWS' complete breadth of offerings. This research explains the data management capabilities of AWS in terms of the ingestion layer, data processing and transformation, data persistence, data security, governance, orchestration and monitoring. The architecture shows two paths to process data — batch with scheduled data collection and real-time, which send events and messages as soon as they are generated or microbatches of small time duration windows. It is one of many ways to architect your pipeline.

This research is focused on native and managed AWS offerings and doesn't necessitate staying with them for building a D&A architecture in AWS.

Some organizations prefer to use third-party products due to reasons like deeper functionality or certain domain-specific features that general-purpose tools may lack. Cloud vendor neutrality for organizations that have footprints on other public clouds or on-premises also leads them to reuse incumbent tools with existing skills, licensing and vendor relations.

Acquire

The ingestion and integration layers are about getting data migrated into the AWS ecosystem. The ingestion layer is responsible for connecting both pull- and push-based data sources and extracting data. With the rapid proliferation of a variety of data sources, organizations want to ingest a wide range of datasets both from internal and external data sources to enrich their data and analytics platform. The ingestion layer thus must be agile, adaptable, flexible and robust to support a wide variety of data sources, data formats and transfer protocols.

As a first step to cloud migration, data ingestion is becoming more and more critical as organizations move to hybrid, multicloud and intercloud-based deployment models, where data generation happens in one place while the data storage, analytics and insights are generated across the WAN. Moreover, as organizations adopt SaaS applications, they are increasingly looking to leverage valuable business data that is locked in these applications by using the analytics and machine learning services offered by AWS to draw unique business insights and improve operational efficiency.

Data Ingestion Options in AWS

AWS provides a wide variety of services to ingest data from across different data sources. Figure 3 shows the different categories of tools in the AWS ecosystem for data ingestion (see Table 1 for AWS' data ingestion capabilities).

Figure 3: AWS Data Ingestion Offerings

These products are intended for different use cases and selecting them according to data volume, velocity, variety and end-user expectations becomes of utmost importance. The most commonly used ones are discussed below.

AWS Glue

- Glue is Amazon's serverless data integration service for discovery, cataloging, preparation and transformation of data. It makes data integration easier by providing both visual and code-based interfaces.

- Glue's capabilities for the extract and load (the E and L) of the ETL functionality are covered in this ingestion section, while transformation (the T of ETL) is covered in the next section.

- AWS Glue has been constantly evolving and is catering to users of different personas:
  - AWS Glue Data Catalog can be used by data stewards to catalog the data and can be used by all other users to quickly find and access the data.
  - AWS Glue Studio can be utilized by data engineers to create, monitor and run ETL workflows. It offers a graphical user interface and generates ETL code in Python or Scala on users' behalf. Users can modify this code or upload their own code if needed.
  - AWS Glue DataBrew can be utilized by data analysts and data scientists to curate data without writing code.
- AWS Glue as an extraction tool provides the ability to automatically discover source schemas using AWS Glue crawlers. Users can also manage and enforce schemas from streaming sources using AWS Glue Schema Registry.
- AWS Glue is completely serverless, which means no infrastructure needs to be provisioned or managed. Along with that, users can auto scale their infrastructure and enhance processing power based on data processing units (DPUs). Enabling job metrics in Glue ETL jobs can help users understand the required max capacity in DPUs.
- AWS Glue supports reading and writing from Amazon Aurora, Amazon RDS for MySQL, Amazon RDS for Oracle, Amazon RDS for PostgreSQL, Amazon RDS for SQL Server, Amazon Redshift, Amazon DynamoDB and Amazon S3, as well as MySQL, Oracle, Microsoft SQL Server, and PostgreSQL databases in your Virtual Private Cloud (Amazon VPC) running on Amazon EC2. AWS Glue also supports data streams from Amazon MSK, Amazon Kinesis Data Streams, and Apache Kafka.
- Glue custom connectors support SaaS data sources like Salesforce, SAP and Snowflake.
- AWS Glue connections are Data Catalog objects that store connection information for a particular data store. Connections store login credentials, uniform resource identifier (URI) strings, virtual private cloud (VPC) information, and more. With AWS Glue Studio, you can also create connections for custom connectors or connectors you purchase from AWS Marketplace.

- Another useful feature is the blueprint capability in Glue through which data engineers can author a blueprint with a job script, configuration file and layout script. This blueprint can be shared with other less-technical users, saving a lot of time and effort, at the same time improving collaboration.
- All notifications are sent to Amazon CloudWatch events by AWS Glue service. To be informed of job failures or completions, you can set up SNS notifications using CloudWatch actions.
- Glue also provides orchestration capabilities through which you could manage dependencies between multiple jobs in data ingestion. These jobs can be triggered on a schedule or a job completion event or on-demand. A thing to note is that when scheduling jobs, the minimum precision for a schedule is 5 minutes.
- Both Glue and Kinesis Data Firehose can be used for streaming ETL. Glue is suitable for complex transformations, and Firehose is suitable for data delivery and preparing it to be processed later by other services. Firehose can be used for certain kinds of transformations as well, like converting the formation of input data from JSON to Apache Parquet or Apache ORC before storing the data in Amazon S3.

Table 1: AWS Data Ingestion Capabilities

(Enlarged table in Appendix)

Capability	Glue Features
Volume	Maximum capacity is expressed in terms of the number of AWS Glue data processing units (DPUs) that can be allocated when a job runs. For more info go to AWS Glue Jobs API.
Source Types	AWS Glue natively supports data stored in Amazon Aurora, Amazon RDS for MySQL, Amazon RDS for Oracle, Amazon RDS for PostgreSQL, Amazon RDS for SQL Server, Amazon Redshift, Amazon DynamoDB and Amazon S3, as well as MySQL, Oracle, Microsoft SQL Server, and PostgreSQL databases in your Virtual Private Cloud (Amazon VPC) running on Amazon EC2. AWS Glue also supports data streams from Amazon Managed Streaming for Apache Kafka, Amazon Kinesis Data Streams, and Apache Kafka. You can use a crawler to populate the AWS Glue Data Catalog with tables. This is the primary method used by most AWS Glue users. A crawler can crawl multiple data stores in a single run. Upon completion, the crawler creates or updates one or more tables in your Data Catalog. Extract, transform, and load (ETL) jobs that you define in AWS Glue use these Data Catalog tables as sources and targets. The ETL job reads from and writes to the data stores that are specified in the source and target Data Catalog tables. For a list of file-based and table-based data stores that crawlers can crawl, go to Which Data Stores Can I Crawl? You can also write custom Scala or Python code and import custom libraries and Jar files into your AWS Glue ETL jobs to access data sources not natively supported by AWS Glue. For more details on importing custom libraries, refer to our AWS Glue Documentation.
Source Formats	AWS Glue supports many file formats, including flat files, TXT, JSON, CSV, Parquet, ORC, Avro, and XML. AWS Glue differentiates its capability with its extensions to Apache Spark called DynamicFrame, which allows developers to handle frequently changing schemas gracefully. AWS Glue also supports modern transactional formats such as Apache Hudi, Apache Iceberg and Delta Lake formats for running transactional workloads in data lakes. Additional file formats can be supported by a number of third-party libraries and offerings from the AWS Marketplace.
Targets	AWS Glue supports the following data targets: Amazon S3, Amazon Relational Database Service (Amazon RDS), third-party JDBC-accessible databases, and MongoDB and Amazon DocumentDB (with MongoDB compatibility).
Extensible Connectors	AWS Glue supports connecting to many relational and nonrelational database systems via Glue's own connectors, connectors offered in the AWS Marketplace, and the ability to use publicly available connection libraries. For example, RDBMS support includes PostgreSQL, MySQL, Oracle, Microsoft SQL Server, Google BigQuery, Vertica, Snowflake, IBM Db2 Database, and more. AWS Glue natively supports reading from and writing to nonrelational databases including Amazon DynamoDB, Amazon DocumentDB, and MongoDB, and with connectors from the AWS Marketplace to Apache Cassandra, Apache HBase, Azure Cosmos DB, and more. Connectors available in the AWS Marketplace also offer connectivity to legacy systems, mainframes, and ERP/CRM systems.
Batch Data Capture	AWS Glue provides a number of connectors, data transformations, and load utilities to process data in batch/microbatch and streaming modes using AWS Glue Studio (a visual IDE for ETL development), AWS Glue Interactive Sessions (for notebook-based interface for data interaction development), and AWS Glue DataBrew (for no-code data wrangling experience). AWS Glue offers a rich set of transformation capabilities to process both structured and semistructured data.
Incremental Data (CDC)	AWS Glue streaming ETL allows customers to process real-time data from Amazon Kinesis Data Streams, Apache Kafka, or Amazon Managed Streaming for Apache Kafka (Amazon MSK) for both real-time and near-real-time streaming ETL. AWS Glue Streaming supports autoscaling to keep costs low and in line with data volume. In addition, AWS Glue integrates with AWS Database Migration Service (AWS DMS) to process and apply change data capture (CDC) records. Customers can use Amazon DMS to set up CDC on sources and use Amazon Simple Storage Service (Amazon S3) to store data or Amazon Kinesis Data Streams to stream the changes. AWS Glue then crawls these data sources and processes the data in real-time or batch mode, depending on customer use case. Users can incorporate the capabilities delivered by the Job Bookmarks feature to track data that has been previously processed.
Velocity of Data	AWS Glue helps customers build extract, transform, and load (ETL) pipelines operating at varying data ingestion frequencies.

Source: Gartner (November 2022)

Amazon Kinesis Data Streams

- Kinesis Data Streams is an AWS-native service for streaming data from a variety of sources ranging from clickstreams, applications logs, or social media (see Table 2 for a comparison of AWS streaming options). Kinesis can capture and analyze terabytes of data per hour to power near-real-time dashboards, generate alerts, implement dynamic pricing and advertising, and more.

- There is less overhead involved with Kinesis as it manages the infrastructure, storage, networking and configuration needed to stream the data. You only need to configure the number of shards needed based on throughput. It also replicates the data across different availability zones, providing high availability and data durability.

- To feed data into the Kinesis Data Streams, users can use PutRecord and PutRecords operations, Amazon Kinesis Producer Library (KPL), or Amazon Kinesis Agent.
- Amazon Kinesis Producer Library (KPL) is an easy-to-use and highly configurable library that helps you put data into an Amazon Kinesis data stream. KPL presents a simple, asynchronous, and reliable interface that enables you to quickly achieve high producer throughput with minimal client resources.
- KPL can be used for synchronous as well as asynchronous use cases; it is recommended to use the higher performance of the asynchronous interface unless there is a use case to use synchronous behavior.
- You can collect, monitor, and analyze your Kinesis Data Streams producers using Amazon CloudWatch. The KPL emits throughput, error, and other metrics to CloudWatch on your behalf, and is configurable to monitor at the stream, shard, or producer level.
- The KPL can incur an additional processing delay of up to RecordMaxBufferedTime within the library (user-configurable). Larger values of RecordMaxBufferedTime results in higher packing efficiencies and better performance. Applications that cannot tolerate this additional delay may need to use the AWS SDK directly.
- Kinesis Agent is a Java software that allows you to easily gather and transfer data to Kinesis Data Streams. The agent watches a group of files continually and feeds fresh data to your stream. The agent handles file rotation, checkpointing, and retries upon failure. It also generates Amazon CloudWatch metrics to assist you in monitoring and troubleshooting the streaming operation. The agent may be installed on Linux-based server settings such web servers, log servers, and database servers.
- To read and process data from Kinesis streams, you need to create a consumer application. There are varied ways to create consumers for Kinesis Data Streams. Some of these approaches include using Amazon Kinesis Data Analytics to analyze streaming data using KCL (Kinesis Client Library), AWS Lambda, Kinesis Data Firehose, AWS Glue streaming ETL jobs, or using the Kinesis Data Streams SDK directly. Amazon Kinesis Client Library (KCL) for Java, Python, Ruby, Node.js, and .NET is a prebuilt library that helps you easily build Amazon Kinesis applications for reading and processing data from an Amazon Kinesis data stream. KCL handles complex issues such as adapting to changes in data stream volume, load-balancing streaming data, coordinating distributed services, and processing data with fault tolerance. KCL enables you to focus on business logic while building applications.

- Users can subscribe to Lambda functions to automatically read batches of records off your Kinesis stream and process them if records are detected on the stream. AWS Lambda periodically polls the stream (once per second) for new records, and when it detects new records, it invokes the Lambda function passing the new records as parameters. The Lambda function is only run when new records are detected. You can map a Lambda function to a shared-throughput consumer as well as to an enhanced fan-out consumer for dedicated throughput.
- You can build a consumer that uses a feature called enhanced fan-out when you require dedicated throughput that you do not want to contend with other consumers that are receiving data from the stream. This feature enables consumers to receive records from a stream with throughput of up to 2MB of data per second per shard. Also, it improves the data delivery speed between producers and multiple consumers by more than 65%.
- Kinesis Data Streams is designed and optimized for large data throughputs and for data retention up to 365 days. By default, data is retained for 24 hours.

Amazon Managed Streaming for Apache Kafka

- Amazon Managed Streaming for Apache Kafka (Amazon MSK) is an AWS streaming data service that can be used to manage Apache Kafka infrastructure and operations. Amazon MSK makes operating, maintaining and scaling Apache Kafka clusters much easier with enterprise-grade security, and built-in integration with AWS service.
- You can migrate existing Apache Kafka workloads with Kafka Connect connectors into Amazon MSK using MirrorMaker.
- Amazon MSK replaces unhealthy brokers, automatically replicates data for high availability, manages Apache ZooKeeper nodes, automatically deploys hardware patches as needed, manages the integrations with AWS services, makes important metrics visible through the console, and supports Apache Kafka version upgrades so you can take advantage of improvements to the open-source version of Apache Kafka.
- Amazon MSK serverless makes it easy for users to run Apache Kafka clusters without having to manage compute and storage capacity. Users only pay for the data volume they stream and retain. Each cluster supports 200 MBps of write capacity and 400 MBps of read capacity per cluster.

- Amazon MSK Serverless allocates up to 5 MBps of instant write capacity and 10 MBps of instant read capacity per partition.
- Amazon MSK Serverless encrypts data through AWS Key Management Service. It also supports AWS PrivateLink and Identity and Access Management (IAM) access controls.
- To replicate data with an existing Apache Kafka cluster to Amazon MSK, users can use third-party tools or open-source tools like MirrorMaker.
- You can monitor the performance of your clusters using the Amazon MSK console, Amazon CloudWatch console, or via JMX and host metrics using Open Monitoring with Prometheus, an open-source monitoring solution.
- Currently, Amazon MSK doesn't offer Reserved Instance pricing.
- Amazon MSK is not suited for ad hoc queries and long-term data storage.
- Refer to Amazon MSK Quota for MSK quota.

Amazon Simple Queue Service

- AWS introduced Simple Queue Service (SQS), its first infrastructure service for public usage, in November 2004. It is a fully managed message queue and topic service and can scan billions of messages per day.
- As opposed to standard queues' feature of SQS, which provides at-least-once delivery, FIFO queue of Amazon SQS provides exactly-once processing, preventing duplicates in the queue. However, in certain cases, there is a slight chance of duplicates — if the producer sends a message, does not receive a response, and then resends the same message. Amazon SQS APIs provide deduplication functionality that prevents your message producer from sending duplicates. Any duplicates introduced by the message producer are removed within a 5-minute deduplication interval.
- Standard queues support a nearly unlimited number of transactions per second. FIFO can support up to 30k messages per second.
- It can be made more flexible and scalable by integrating with compute services such as Amazon EC2, Amazon Elastic Container Service (ECS), and AWS Lambda, as well as with storage and database services such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB.

- To access the service, customers can use AWS Management Console, a web service API or AWS SDKs.
- To handle messages that can't be processed, dead-letter queues can be configured. When configuring a dead-letter queue, you are required to set appropriate permissions for the dead-letter queue redrive using RedriveAllowPolicy. These dead-letter queues receive messages after a maximum number of processing events cannot be completed.
- An SQS message can contain up to 10 metadata attributes.
- Users should opt for SQS long polling to save cost. While the regular short polling returns immediately, even if the message queue being polled is empty, long polling doesn't return a response until a message arrives in the message queue, or the long poll times out. In general, you should use a maximum of 20 seconds for a long-poll timeout. Since higher long-poll timeout values reduce the number of empty ReceiveMessageResponse instances returned, try to set your long-poll timeout as high as possible.
- The following features of AWS services aren't currently compatible with FIFO queues:
    - Amazon EC2 Auto Scaling lifecycle hooks
    - AWS IoT rule actions
    - AWS Lambda dead-letter queues
- SQS supports multiple producers, but if multiple producers send messages in parallel but without waiting for the success response, the order between producers might not be preserved.
- SQS messages retention period can be configured between 1 minute and 14 days, 4 days being the default.
- There is a quota of 120,000 for the number of in flight messages (in flight messages are messages received by customers and not deleted yet) for a standard queue and 20,000 for a FIFO queue.
- The permissions API provides an interface for sharing access to a message queue to developers. However, this API cannot allow conditional access or more advanced use cases.

- Amazon SQS messages can contain up to 256 KB of text data, including XML, JSON and unformatted text.

Table 2: Comparing AWS Streaming Options

(Enlarged table in Appendix)

Feature	Kinesis	MSK	SQS
Message Size Limit	1 MB	Dependent on instance type, replication factor, throughput, etc. 8 MB (for serverless)	256 KB
Retention Period	7 days (365 with long-term retention, optional cost)	Unlimited	14 days
Decision Complexity	Easy (with On-Demand mode, capacity scales automatically)	Medium (need to choose between serverless that requires minimal setup and provisioned that requires deciding between broker types and numbers)	Easy (need to create queues, no decision to make)
Multiple Consumer Support	Yes	Yes	No
Replay Support	Yes	Yes	No
Analytics Capability	Yes	Yes	No
Message Ordering	Yes	Yes	Only for FIFO
Cloud Agnostic	No	Yes	No
Granularity Level	Shards	Kakfa partition	Messages
Checkpointing	With Amazon DynamoDB	Yes	Yes
Managed/Serverless	Serverless	Both	Managed
Customization	Minimum	Maximum	Not required

Source: Gartner (November 2022)

AWS Database Migration Service

- AWS Database Migration Service (DMS) helps customers migrate their existing data into AWS quickly and securely.
- DMS can be used to stream data (via CDC) to Kinesis and S3 data lakes.
- It supports homogeneous migrations such as Oracle to Oracle, as well as heterogeneous migrations between different database platforms, such as Oracle or Microsoft SQL Server to Amazon Aurora.

- Heterogeneous migration can be achieved by first using AWS Schema Conversion Tool to convert the source schema and code to match that of the target database. After schema conversion, DMS is used to migrate data from source to target database. Schema Conversion Tool (SCT) is not a mandatory tool for heterogeneous migration, it is optional for cases where you need to migrate stored procedures or want to do a schema mapping assessment. For example, migrating to Neptune Graph DB has a different approach and does not use SCT.
- AWS SCT automates the conversion of Oracle PL/SQL and SQL Server T-SQL code to equivalent code in the Amazon Aurora/MySQL dialect of SQL or the equivalent PL/pgSQL code in PostgreSQL. Code fragments that cannot be automatically converted to the target language will be clearly documented by SCT.
- SCT is primarily used to migrate large data warehouse workloads, while DMS typically moves smaller relational workloads (up to 10 TB) and MongoDB. Continuous replication is supported by DMS, but not by SCT.
- DMS supports encrypted connections, so data doesn't have to travel in decrypted format over the wire. But DMS has to have access to the decrypted row data to perform the conversions. You can use AWS Database Migration Service's Basic Schema Copy feature to quickly migrate a database schema to your target instance.
- When a Basic Schema Copy is performed, new tables and primary keys are automatically created in the target instance if no similar tables already exist in the target instance. Users can also configure to drop the existing tables if they wish. Secondary indexes, foreign keys and stored procedures will not be migrated by Basic Schema Copy. For more customization, Schema Conversion Tool can be used.
- When migrating from relational databases, use limited large binary objects (LOB) mode. If tables have few large blobs and many smaller LOBs, migrate using two separate tasks in full LOB and limited LOB mode.
- With DMS Studio, you can discover databases in your local environment and get recommendations for migrating your data to AWS.
- AWS DMS Fleet Advisor is a free, fully managed capability of AWS Database Migration Service that lets you migrate your database and analytics fleet to the cloud at scale with minimal effort. It automatically inventories and assesses your on-premises database and analytics server fleet and identifies potential migration paths.

AWS Kinesis Data Firehose

- Kinesis Data Firehose is a near-real-time service to load data that is aggregated from multiple sources into AWS services and other destinations like Amazon S3, Amazon Redshift, Amazon OpenSearch Service, Splunk, Datadog, New Relic, Dynatrace, Sumo Logic, LogicMonitor, MongoDB Cloud, Honeycomb, Coralogix, and HTTP endpoint as destinations. It can automatically scale to manage the workload throughputs and does not require any ongoing administration.
- Provides prebuilt Lambda blueprints — can be used without any change or customization.
- It also adds some functionalities like batching, compressing and encrypting data before loading.
- Snappy, zip, and gzip compression formats are currently supported by the service. When further loading data to Amazon Redshift, only gzip is supported.
- It supports more than 30 sources, including:
    - Amazon Kinesis Data Firehose API, which uses the AWS SDK for Java, .NET, Node.js, Python, or Ruby
    - Kinesis Data Streams, where Kinesis Data Firehose reads data easily from an existing Kinesis data stream and load it into Kinesis Data Firehose destinations
    - AWS natively supported services like AWS CloudWatch, AWS EventBridge, AWS IoT, or AWS Pinpoint
    - Kinesis Agents, a stand-alone Java software application that continuously monitors a set of files and sends new data to your stream
    - Fluent Bit, an open-source Log Processor and Forwarder
    - AWS Lambda, a serverless compute service that lets you run code without provisioning or managing servers
- You can write your Lambda function to send traffic from S3 or DynamoDB to Kinesis Data Firehose based on a triggered event.
- It has built-in format conversion like JSON formats to Apache Parquet and Apache ORC. It also supports static or dynamic partitions to achieve better performance of data loaded into S3.

- With CloudWatch Logs subscriptions, you can stream data from CloudWatch Logs to Kinesis Data Firehose. You should keep Firehose's compression configuration as uncompressed since CloudWatch Logs are already compressed in gzip format.
- KDF buffers data before delivering it to S3, the buffer size can be configured. The frequency interval for data delivery depends on the buffer size or the buffer interval configured.
- Refer to the following page for limits on KDF — Amazon Kinesis Data Firehose Quota.
- Firehose strives to deliver effectively once delivered. However, in rare circumstances, users may also find the same record(s) delivered to different S3 objects. For other destinations such as HTTP/EP and/or Redshift, Firehose might deliver duplicates due to retries after failure to deliver record batches. If data delivery fails, then Amazon Kinesis Data Firehose will retry to deliver data every 5 seconds for up to a maximum period of 24 hours. If the issue continues beyond the 24-hour maximum retention period, then Amazon Kinesis Data Firehose discards the data. Kinesis Data Streams data source will retry to deliver data to your Amazon S3 bucket every 5 seconds for a maximum period of time configured on Kinesis Data Streams.

AWS DataSync

- The Amazon Web Services DataSync service streamlines, automates, and accelerates the copying of large amounts of data between on-premises, edge, or other cloud storage and AWS Storage services, as well as between AWS Storage services.
- Using DataSync, you can copy data between Network File System (NFS) shares, Server Message Block (SMB) shares, Hadoop Distributed File Systems (HDFS), self-managed object storage, Google Cloud Storage, AWS Snowcone, Amazon Simple Storage Service (Amazon S3), Amazon Elastic File Systems (Amazon EFS), Amazon FSx for Windows File Server file systems, Amazon FSx for Lustre file systems and Amazon FSx for OpenZFS file systems.
- Data transfer between storage systems and AWS services is accelerated by a purpose-built network protocol and scale-out architecture. Files and objects are automatically moved, scheduled, monitored, encrypted and verified, and customers are notified of any issues through DataSync. Pay only for data copied, no minimum commitments required.

- DataSync supports both standard storage protocols (NFS, SMB), HDFS, and the Amazon S3 API. After transferring your initial dataset, you can schedule subsequent transfers of new data to AWS.
- To ensure your data is secure, intact, and ready to use, DataSync uses encryption and integrity validation. With the built-in bandwidth throttle, you can limit the amount of bandwidth that DataSync consumes to minimize impact on workloads that rely on your network connection.
- With DataSync's filtering functionality, you can exclude temporary files and folders and copy only a subset of files from your source location.
- Using the built-in task scheduling functionality, you can regularly archive data that needs to be retained for compliance or auditing purposes, such as logs, raw footage, or electronic medical records.
- By using AWS DataSync, you can copy data from Google Cloud Storage using the S3 API, or from Azure Files using SMB. A DataSync agent needs to be deployed in your cloud environment or on Amazon EC2.
- Deploying an agent is not required to transfer data between AWS storage services within the same AWS account.
- DataSync performs integrity checks to ensure that data in both source and destination is consistent; additionally, it can use optional full-file checksums. If for any reason the task is interrupted, the next run will transfer the missing files to maintain consistency.
- AWS management console or command line interface (CLI) can be used to check the status of transfers. To check the number of files and amount of data transferred, Amazon CloudWatch metrics can be used.
- DataSync can be used with AWS Direct Connect link to access public service endpoints, or with VPC endpoints. The advantage with VPC endpoints is that data is not transferred over public internet, increasing the security of data.
- When DataSync is used to transfer data between S3 objects, it can also transfer object metadata and tags, but it cannot copy other object information like object ACL or prior object versions.
- AWS DataSync may affect the performance of source data sources depending upon quantity and size of data transferred. Configuring a bandwidth limit can reduce the load in cases where the source data source is too busy with I/O operations.

- Data transferred between source and destination is encrypted via Transport Layer Security (TLS), which replaced Secure Sockets Layer (SSL). Data is never persisted in AWS DataSync itself.
- "If you currently use SFTP to exchange data with third parties, AWS Transfer Family provides a fully managed SFTP, FTPS, and FTP transfer directly into and out of Amazon S3, while reducing your operational burden." (From the AWS DataSync FAQs)
- Use AWS DataSync for secure, managed, online transfers of TB to PB of active or passive data, if your available network resources can support the volume of data you need to move.

AWS Snow Family

- Snow Family is a collection of highly secure and portable devices that can be used to migrate large amounts of data onto the cloud without depending on the network. The members of this family are:
  - AWS Snowcone — Smallest member of the family; data can be moved to AWS servers by offline shipping the device or online with AWS DataSync.
  - AWS Snowball — Suitcase-sized data migration, comes with two device options: compute optimized or storage optimized.
  - AWS Snowmobile — It is a shipping container moved with a tractor-trailer.
- These services can assist with data migration, disaster recovery, data center shutdown and remote data collection projects.
- These devices are used mainly because movement of data could take a lot of time over the network due to certain factors like data volume, connectivity, bandwidth or stability.
- As a rule of thumb, if it takes more than a week to transfer data over the network then organizations should use snowball devices.

Organize and Analyze

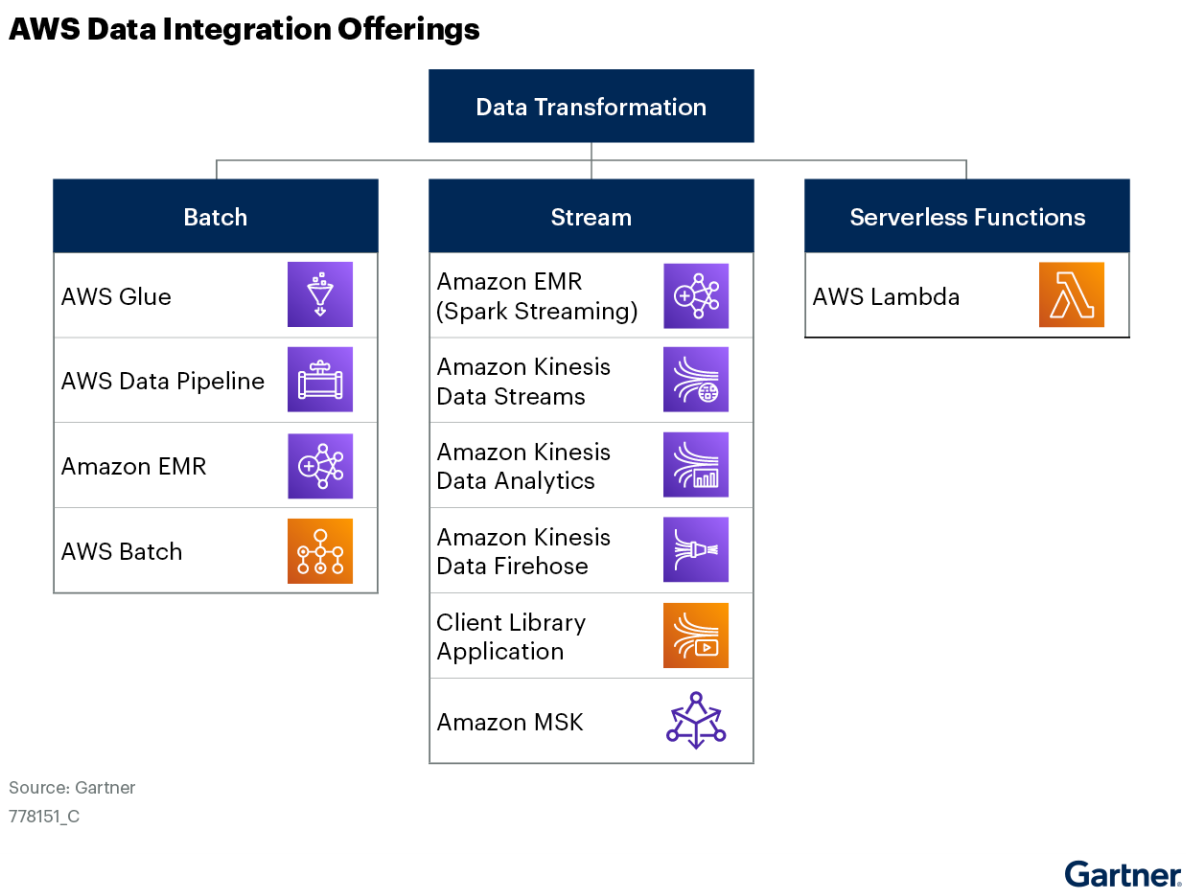
Data engineers are tasked with preparing source data for consumption by analysts, data scientists and downstream systems. This requires data collected from multiple sources to be integrated and transformed appropriately.

Data Integration and Transformation (Batch and Real Time)

Data engineers are tasked with preparing source data for consumption by analysts, data scientists and downstream systems. This requires data collected from multiple sources to be integrated and transformed appropriately.

Figure 4 shows AWS' native data transformation options.

Figure 4: AWS Data Integration Offerings



This section discusses the some of AWS' native data transformation options (see Table 4):

AWS Glue

- As mentioned earlier, AWS Glue is a serverless data integration service for discovery, cataloging, preparation and transformation of data. It has the following modules:
  - Data discovery — Automatic schema registration using AWS Glue crawlers, also manages schemas for data streams with AWS Glue Schema Registry
  - Data catalog — Central metadata repository supporting various AWS services; Hive metastore compatible
  - ETL engine — For ingesting and transforming data, orchestration job schedules and dependencies
  - GUI-based data preparation and transformation — Drag and drop interface that can automatically generate code in Python and Apache Spark
- Glue offers a GUI-based interface referred to as Glue Studio to author data processing jobs, generating spark code on users' behalf. Users can also write their own code or modify the ETL code generated by AWS Glue using Scala or Python. Users can also create and connect to development endpoints that offer ways to connect to notebooks and integrated development environments (IDEs).
- Along with data transformation, Glue offers features for data orchestration to manage job schedules and interdependencies. In case of any errors, AWS Glue can push notifications to Amazon CloudWatch.
- Glue can also support streaming workloads like Amazon Kinesis Data Streams, Apache Kafka, and Amazon MSK. Add the stream to the Glue Data Catalog and then choose it as the data source when setting up your AWS Glue job.
- Glue DataBrew is a visual data preparation tool for data scientists and data analysts. It can be used to visualize, clean, and normalize terabytes and even petabytes of data directly from your data lake, data warehouses, and databases, including Amazon S3, Amazon Redshift, Amazon Aurora, and Amazon RDS.
- DataBrew supports 250 built-in transformations to combine, pivot and transpose the data without writing code.
- As part of the AWS Glue DataBrew, data is automatically filtered for anomalies, corrected for invalid, incorrectly classified or duplicate data, normalized for standard date and time values, or aggregated for analysis.

- Natural language processing (NLP) can be applied to complex transformations, such as converting words into a common base or root word. Multiple transformations can be grouped together, saved as recipes and applied directly to new incoming data.
- AWS recently launched Glue Interactive Sessions. With this new release, users can build, test and run data preparation or analytics applications from the environment of choice. Users can use Jupyter-compatible notebooks to author and test their scripts. It provides an open-source Jupyter kernel that integrates almost anywhere that Jupyter does; for instance, integrating with IDEs such as PyCharm, IntelliJ, and Visual Studio Code.
- AWS Glue Studio also allows users to interactively author jobs in a notebook interface based on Jupyter notebook.
- Glue Auto Scaling (available with Glue 3.0 or later) is able to automatically add and remove workers from the cluster depending on the parallelism at each stage or microbatch of the job. If you choose the maximum number of workers, AWS Glue will choose the right size resources for the workload.
- Since AWS Glue ETL jobs are Spark-based, it might not be recommended if the use cases require an engine other than Apache Spark or if a variety of heterogenous jobs on a variety of execution engines like Apache Hive or Apache Pig is needed. Amazon Elastic MapReduce (EMR) might be a better option in such cases.
- AWS Glue doesn't offer extensive changes to configuration parameters since it's a fully managed service. If that is needed, then please consider Amazon EMR or Amazon on EKS.

Amazon EMR

- Amazon EMR is a cloud big data platform for processing vast amounts of data using open-source tools such as Apache Spark, Apache Hive, Apache HBase, Apache Flink, Apache Hudi, and Presto.
- With Amazon EMR, AWS managed to integrate a hadoop framework with EC2, which eases users' concern for compute capacity management. Other services like AWS CloudWatch can provide alerts and recommendations. With Kubernetes, users can use EMR to submit workloads to Amazon EKS clusters.
- It has 4 deployment modes — EMR on EC2, EMR on EKS, EMR Serverless and EMR on AWS Outposts.

- EMR provides access to underlying infrastructure to users and allows customization; this is in contrast to AWS Glue, where users do not have access to the underlying infrastructure and therefore are provided with very limited customization options. EMR workloads can be triggered by an EMR console, API, SDK or CLI. To have an interactive session, EMR Studio can also be used.
- EMR can also be triggered by orchestration tools such as Apache Airflow, Amazon Managed Workflows for Apache Airflow (MWAA) and AWS Step Functions.
- AWS EMR allows resize of a running cluster. At any time, you can add core nodes that hold HDFS to increase your processing power and HDFS storage capacity. You can also use Amazon S3 natively, or EMR File System (EMRFS) instead of local HDFS, which provides flexibility and cost-efficiency by decoupling memory and compute from storage.
- At any time, users can also add and remove task nodes that can process Apache Hadoop jobs but do not maintain HDFS. Some customers add hundreds of instances to their clusters when their batch processing occurs, and remove the extra instances when processing completes. For example, you may not know how much data your clusters will be handling in six months, or you may have spiky processing needs.
- With Amazon EMR, you can launch a persistent cluster that runs indefinitely or a temporary cluster that ends when the steps are completed. In both cases, users will only be charged for the cluster active time. The performance of EMR depends on the type and number of Elastic Compute Cloud (EC2) instances chosen, along with other factors. Users should choose an instance type suitable for their processing requirements, with sufficient memory, storage and processing power.
- Users can also configure instance fleets for a cluster to choose from a variety of EC2 instance provisioning options. Within instance fleets, users can specify target capacities for On-Demand Instances and Spot Instances. Based on the availability zones you select, Amazon EMR tries to provide you with the optimal mix of capacity and price.

- An EMR cluster is a collection of EC2 instances referred to as nodes. The nodes are classified into the following three categories:
  - Master node
    - Runs master components of distributed applications.
    - Runs HDFS NameNode service, tracks the status of submitted jobs and monitors the health of the instance groups.
    - Each cluster must have a master node, and so it is also possible to have a cluster with just a single node that is also the master node.
  - Core node
    - Runs tasks and stores data.
    - Multinode clusters have at least one core node.
    - Once provisioned, they should not be removed, as it may lead to loss of data.
  - Task node
    - Runs tasks but does not store data.
    - These are optional nodes to increase processing capacity.
    - Cost-effective when used with Spot instances.
- EMR provides flexibility to enable/install several Hadoop applications to read from or write to other Amazon services. For example, streaming consumer applications deployed in EMR can query data from Kinesis Data Streams. This can help perform real-time streaming or microbatch processing of Kinesis data streams using existing Hadoop applications, such as Flink or Spark streaming.

- EMR Studio is an Integrated Development Environment (IDE) that provides a number of features, like:
  - Fully managed Jupyter notebooks that can be accessed outside AWS Console; this is specially useful if data scientists or data analysts do not have access to the AWS console
  - Integration with Single Sign-on (SSO)
  - Load custom kernels and Python libraries in notebooks
  - GitHub integration
  - Passing parameters to notebooks
- Amazon EMR also supports a variety of other popular applications and tools in the Hadoop ecosystem, such as R (statistics), Apache Mahout (machine learning), Ganglia (monitoring), Apache Accumulo (secure NoSQL database), Hue (user interface to analyze Hadoop data) and HCatalog (table and storage management).
- Amazon EMR's S3DistCp is an extension of the open-source tool DistCp that uses MapReduce to efficiently move large amounts of data from S3 to HDFS, from HDFS to S3, and between S3 buckets.
- EMR Serverless automatically estimates the processing required for a particular job and assigns workers that it can automatically scale up or down based on the workload and parallelism required. Users can specify the minimum and maximum number of concurrent workers and the vCPU and memory configuration for workers. The ability to scale up on workload requirements prevents jobs from failing, which is a big plus.
- Although the service is serverless, you still need to initialize the count and configuration of worker nodes. An EMR Serverless application without predefined workers can take up to 120 seconds to determine the required resources and provision them.

- With EMR Serverless, workers can be initialized and ready to respond within seconds, effectively creating an on-call pool of workers. This feature is called pre-initialized capacity, and it can be configured by setting the initial-capacity parameter of each application. The EMR Serverless automatically adds more workers (up to the maximum concurrent limit that you specify) if the job requires more workers than you have pre-initialized. As soon as the job finishes, EMR Serverless automatically returns to maintaining the pre-initialized workers. After 15 minutes of inactivity, workers are automatically shut down. Using the UpdateApplication API or EMR Studio, you can change the default idle timeout for your application.

Table 3 highlights the differences between EMR Serverless and EMR on EC2.

Table 3: Differences Between EMR Serverless and Amazon EMR on EC2

(Enlarged table in Appendix)

Feature	EMR Serverless	Amazon EMR on EC2
Resilience to Availability Zone failures	Y	N
Open-source frameworks supported	Apache Spark and Apache Hive	Apache Hive, Apache Pig, Apache Sqoop, Apache Spark, Apache Tez, Apache HBase, Apache Oozie, Presto, Trino, etc.
Support for fine-grained authorization using AWS Lake Formation	N	Y
Integration with Apache Ranger for table- and column-level permission control	N	Y
Customize operating system images	N	Y
Customize and load additional libraries and dependencies	Y	Y
Run workloads from SageMaker Studio as part of machine learning (ML) workflow	N	Y
Connect to self-hosted Jupyter Notebooks	N	Y
Build pipelines using Apache Airflow and Amazon Managed Workflows for Apache Airflow (MWAA)	Y	Y
Build and orchestrate pipelines using AWS Step Functions	N	Y

Source: Gartner (November 2022)

AWS Data Pipeline

- AWS Data Pipeline is a web service by Amazon that supports easy scheduling of data movement and data processing capability in the cloud. It integrates with both on-premises and cloud data sources. AWS Data Pipeline allows you to quickly define a dependent chain of data sources, destinations, and predefined or custom data processing activities called a pipeline.
- Data Pipeline submits MapReduce jobs to EMR using EmrActivity or HiveActivity. Pipeline itself does not have capability to run, rather it submits a job as a remote application or orchestrator.

- Data Pipeline defines task dependencies and can do retries on failures or provide necessary alerts. It also supports pipelines that run across regions and provides precondition checks.
- Pipelines can be created through a drag-and-drop console. Common preconditions are built into the service, like checking the existence of a file in S3 by simply providing the name of the Amazon S3 bucket and the path of the file that you want to check for.
- Data Pipeline provides built-in support for the following activities:
  - CopyActivity: This can copy data between Amazon S3 and JDBC data sources, or run an SQL query and copy its output into Amazon S3.
  - HiveActivity: This allows you to execute Hive queries easily.
  - EmrActivity: This allows you to run arbitrary Amazon EMR jobs.
  - ShellCommandActivity: This allows you to run arbitrary Linux shell commands or programs.

AWS Lambda

- AWS Lambda provides the ability to run snippets of codes in the cloud without provisioning or managing any servers. AWS Lambda is completely serverless, and customers don't have to provision any EC2 servers to run their jobs; it does it all for you and even scales to the rate of incoming events.
- With AWS Lambda you only pay for the compute time when your code is running. This is especially useful for integration methods like batch where you might not want to pay for the servers at times when no job is running. Also for streaming workloads, Lambda can scale up when the request throughput is high and scale down later so you don't pay for peak capacity at all times.
- With AWS Lambda, customers do not have the option to access the infrastructure even if they wanted to do that, so tasks like performing health checks, monitoring and applying security patches are not applicable.
- AWS Lambda supports code written in Node.js (JavaScript), Python, Java (Java 8 compatible), C# (.NET Core), Go, PowerShell and Ruby. Your code can include existing libraries, even native ones.

- An event from services like S3, Kinesis Data Firehose, Kinesis Data Streams, DynamoDB, SNS, SQS, IoT and CloudWatch can trigger an action for Lambda. Some of these services publish the events to Lambda by invoking the cloud function directly (for example, Amazon S3). Lambda can also poll resources in other services that do not publish events to Lambda. For example, Lambda can pull records from Amazon Kinesis Streams or an Amazon SQS queue and execute a Lambda function for each fetched message.
- Lambda functions can be accessed using the dashboard in Lambda console or through AWS CLI and AWS SDK.
- AWS Lambda uses replication and redundancy to provide high availability and does not have any downtimes or maintenance windows.
- Events in AWS Lambda can be processed in milliseconds as Lambda automatically caches the code for the users.
- AWS Lambda can run execution up to a maximum of 15 minutes for each function, so be careful to write jobs that are time-consuming. For heavier processing scripts running in EC2, EKS, or ECS, AWS Batch or AWS Glue Python shell are better options.
- Lambda is stateless and can not be used to keep track of the last execution when running multiple jobs one after the other. DynamoDB or S3 can be used to store state information.
- Lambda is preferred or positioned as the serverless compute for short-running jobs.

Table 4: A Comparison of Data Transformation and Processing Services on AWS

(Enlarged table in Appendix)

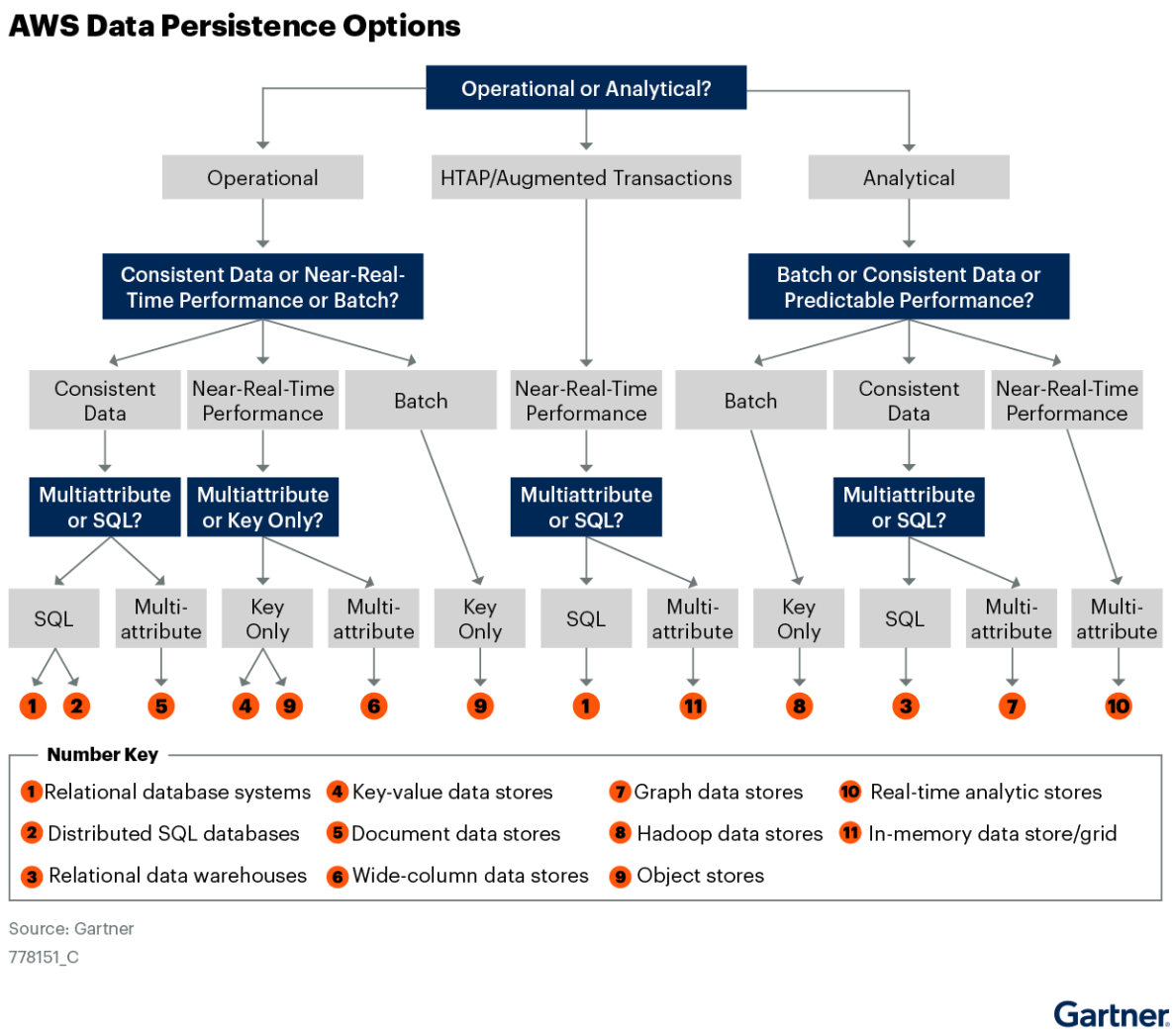
Feature	*Lambda*	*Glue*	*EMR*	*Amazon Data Pipeline*
User Interface	AWS Lambda console (Code interface), AWS CLI, AWS SDK	Glue console (Visual- and Code-based)	EMR console (Code interface), API, SDK or CLI	Console (Visual), API and CLI
Managed/Serverless	Serverless	Serverless	Managed and serverless	Managed
Ease of Use	Code extensive	Easy to use	Code extensive	Easy to use
Batch/Streaming	Both	Both	Both	Batch
Orchestration	Built-in, Also can be invoked from AWS Step Functions and Apache Airflow	Built-in	Can be achieved with another service like AWS Step Functions or Airflow (but may not work for EMR Serverless)	Built-in
Built-In Transformations	No	Yes	No	Limited
Compute Engine	Any	Spark	Apache Spark, Apache Hive and Presto	Spark, Hive, Pig, etc.
Specialization	Run any type of code without the need of provisioning any server	Easy-to-use ETL and cataloging service in AWS with loads of features and integration with other services	Interactive analysis and machine learning using Apache Spark, Hive and Presto. Provides more control over configurations and is ideal for cloud migration of on-premises Hadoop environments	Data movement and processing between AWS services
Anti-Patterns	Has a 15-minute time limit on functions	Doesn't offer extensive changes to configuration parameters since it's a fully managed service	Not as easy to use as Glue	Suitable for batch workloads only

Source: Gartner (November 2022)

Data Persistence and Storage

Some years ago, polyglot persistence became a popular option to deploy multiple data storage technologies to meet the needs of varying use cases. Figure 5 shows a decision point for selecting the right DBMS (see Decision Point for Selecting the Right DBMS for more information).

Figure 5: AWS Data Persistence Options



AWS, too, has many different types of storage options. Table 5 provides a list of AWS-native options for the above decision tree.

Table 5: AWS Options for Decision Tree

(Enlarged table in Appendix)

Number Key	AWS -Native
1	Amazon RDS
2	Amazon Aurora
3	Amazon Redshift
4	Amazon DynamoDB
5	Amazon DocumentDB
6	Amazon Keyspaces
7	Amazon Neptune
8	Amazon EMR
9	Amazon S3
10	Kinesis
11	Amazon ElastiCache, Amazon MemoryDB for Redis

Source: Gartner (November 2022)

Some of the AWS persistence and storage options are discussed below.

Amazon S3

- Data lakes are being adopted exponentially by organizations due to reasons like onboarding a large volume of new and diverse data sources, processing and analyzing real-time streaming data, catering to multiple personas, and supporting different applications and requests.
- Data lakes are complex, and building them successfully involves integrating different technologies because there is no one end-to-end product on the market to implement enterprisewide data lakes. AWS offers a variety of solutions that cater to its data lake in S3. It is also tightly integrated with the data warehouse (Amazon Redshift) and is increasingly getting seamless integration with other services through Glue and Lake Formation.

- Contrary to the current hype of decentralized architectures where a data lake or a warehouse could be considered a monolith, it is not the case. In most cases, data lakes seem to be the heart of modern data architecture, and in some cases, customers build multiple data lakes to cater to the different LOB. AWS supports architectures that enable governed sharing of data products across one or more data lakes.
- Amazon S3 is an object storage capable of storing and retrieving any amount of data from anywhere at low cost. It is highly scalable and you only pay for what you use. Although the total volume is unlimited, individual S3 objects cannot be greater than 5TB, and the largest object that can be uploaded in a single PUT operation is 5GB. These S3 objects are stored in buckets. It seems really similar to files being stored in directories, but there's no concept of directories or hierarchies in S3. These objects just have different keys with very long names that contain slashes ("/"). The UI tricks us to think otherwise, but objects are at the same level.
- Since Dec 2020, all operations in Amazon S3 are now strongly consistent, which means after a successful write/overwrite operation, any subsequent read operation retrieves the latest version of the object in S3.
- S3 offers a range of storage classes to choose from based on data access, resiliency and cost requirements of your workloads. To decide the one that's best for you, consider the access patterns and retention time of your data to optimize the costs. In most cases, the workloads are constantly changing and are unpredictable. In such cases, S3 Intelligent-Tiering should be the default choice.
- Refer to the link to check out a detailed description of different Amazon S3 Storage Classes.
- S3 standard storage class is ideal for frequent data access (more than a month), S3 standard-infrequent access works best if data is retained for at least a month and accessed once every month or two. Amazon S3 glaciers are ideal for data archiving with retrieval flexibility and the lowest cost among the options. Users can also use S3 life cycle rules to transition from one storage class to another over time.
- Versioning in S3 allows users to preserve data against unintended deletes and provides easy rollback to a previous version. It is enabled at the bucket level and is a very good practice to follow. A thing to note is that suspending versioning would not delete the previous versions.

- Users can also opt for S3 Replication to copy data asynchronously between S3 buckets between either the same region (S3 Same-Region Replication) or across regions (S3 Cross-Region Replication). Versioning must be enabled for both the source and destination in order to carry out replication. A thing to note is that after activating versioning, only the new objects are replicated; to replicate existing objects, use S3 batch replication. S3 also allows replicating delete markers (soft-deletes) from source to destination, although it doesn't support deletion with version ID to avoid malicious deletes.
- To query S3 data in place without having to move data into a separate analytics platform, customers can use either S3 select, Amazon Athena or Redshift Spectrum.
- Users can enable S3 event notifications to receive them in response to certain events in S3 bucket, such as PUT, POST, COPY, and DELETE events. These push notifications can be pushed to Amazon EventBridge, Amazon SNS, Amazon SQS, or directly to AWS Lambda.
- Amazon S3 has access control mechanisms like IAM policies, bucket policies, access point policies, Amazon VPC, Access Control Lists, Query String Authentication, service control policies (SCPs) in AWS Organizations, and Amazon S3 Block Public Access to control access. Users can also use Server-Side Encryption (SSE) option to encrypt data stored at rest.
- To get analytics and insights on S3 storage, users can try a feature called Amazon S3 storage lens, which provides wide visibility into object storage usage and activity trends, as well as actionable recommendations to improve cost-efficiency and apply data protection best practices.
- S3 can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per partitioned prefix.
- For high-volume operations, consider naming schemes with more variability at the beginning of the key names for maximum throughput. Latency on S3 operations depends on key names, since prefix similarities become a bottleneck at more than 5,500 read requests per second.
- Use Amazon S3 for write once, read many times-type workloads.
- Do not use S3 to host OS or databases.
- Use AWS SDK and AWS S3 API mechanisms to optimize S3 downloads and uploads via increased parallelism and concurrency when dealing with large transfers.

- To delete or archive based on object tags, tag objects so that it is easier to apply life cycle policies as well as to search.
- Consider compression schemes for large data that isn't already compressed to mitigate S3 bandwidth and cost constraints.

Amazon Athena — A Query Engine for S3 and More

- Athena is an interactive query service that lets you analyze data in S3 and other locations using Presto. It is a serverless engine so there is no overhead to manage infrastructure. Also, unlike a data warehouse, Athena being a query engine doesn't require loading data into it — it works directly with data stored in S3.
- Athena is built on top of presto, a distributed query engine for big data using the SQL query language. Amazon Athena uses Hive for DDL (Data Definition Language) and for creation/modification and deletion of tables and/or partitions.
- Athena supports a variety of file formats like CSV, JSON, ORC, Apache Parquet and Avro. It also supports table formats such as Hudi, Iceberg, Delta Lake and Lake Formation Governed Table.
- Athena uses a managed data catalog to store schema information of data stored in Amazon S3 when it is queried. The preferred choice for this data catalog is AWS Glue; otherwise, Athena can also store it inside its own internal catalog. The catalog can be modified using DDL statements or via the AWS management console. Athena uses schema-on-read technology, which means table definitions are applied to data in S3 when it is queried. Any changes or deletion to table definition or schema doesn't impact the data stored in S3.
- A thing to note is that Athena can work along with Redshift. If data is stored in S3 before loading to Redshift then Athena can be used to query that data. This is especially useful for any sort of data validation before it lands into Redshift. Athena can also be used to provide SQL-based data transformation in S3 before landing the data in Redshift.
- Partitioned or compressed data limits the amount of data scanned, which helps improve performance.
- Federated query in Athena lets users query data outside Amazon S3 as well. With Athena Federated Query, you can run SQL queries across data stored in relational, nonrelational, object and custom data sources.

- Since Athena is serverless, it does require proper monitoring to keep costs under control. A useful feature to manage costs is Athena workgroups, which can be used to separate users, teams, applications or workloads into different groups. By doing so, it's easier to restrict access or limit the queries that a specific group can run.
- These workgroups can also be integrated with IAM, CloudWatch and SNS. Each workgroup can have their own query history so they don't have access to query logs outside their group. CloudWatch and SNS can help track and alert in case any limitation is hit.
- Athena has a pay-as-you-go model and charges per TB scanned. Successful or canceled queries count; failed queries do not. Users can save a lot of money by using columnar formats with partitions (restricting data scanned per query). If users want to add partitions after the table is built, then they can use the MSCK REPAIR TABLE command.
- Small-file compaction helps improve performance in Athena — try to keep a small number of large files rather than a large number of small files.
- Athena also now supports atomicity, consistency, isolation and durability (ACID) transactions to enable multiple concurrent users to concurrently add or delete Amazon S3 objects in an atomic manner. It does so by using Apache Iceberg. This removes the need for custom record locking. Additionally, users now have the option to use time travel capability to jump back to a previous state of data.
- Athena now supports UDFs.
- Athena may not be the ideal choice for some scenarios, and alternative AWS services should be used as listed below:
    - Reporting and BI workloads — Amazon Redshift
    - Complex ETL Workloads — Amazon EMR/AWS Glue
    - RDBMS — MySQL

Amazon Redshift

- Redshift is a fully managed, petabyte-scale data warehouse service that is optimized for datasets ranging from a few hundred gigabytes to petabytes or more. Being a data warehouse, Redshift is designed for OLAP and not OLTP.

- Amazon recently launched Redshift Serverless, using which you won't have to manage or configure your clusters. Redshift Serverless is able to automatically scale based on workloads, and users only pay for what they use.
- Redshift provides fast query and I/O performance by using columnar store technology while parallelizing and distributing data across multiple nodes.
- Maintaining a Redshift data warehouse is easy. As it is fully managed, users don't have to worry about management tasks like hardware provisioning, software patching, setup, configuration, monitoring nodes and drives to recover from failures, or backups.
- Amazon Redshift has automatic tuning capabilities, and surfaces recommendations for managing the data warehouse in Amazon Redshift Advisor, which also provides the capability to query the data inside the data lake using Amazon Redshift Spectrum. The querying executed through the Redshift Spectrum goes through an SQL endpoint in Redshift, which generates a query plan. Spectrum can query exabytes of unstructured data in S3 without loading it inside a data warehouse.
- Like Athena, Redshift Spectrum can access the metadata information from Glue catalog, and then presents the cataloged tables inside the data warehouse and provides the ability to combine data from both data lake and data warehouse. For end users, it's all under one single glass pane and they don't have to worry about what is being accessed from the data lake and what is present in the data warehouse. Spectrum truly provides the ability to separate storage and compute, allowing both to independently.
- Columnar data formats such as Apache parquet are recommended to use with Amazon Redshift Spectrum.
- Setting up Amazon Redshift Spectrum requires creating an external schema and tables. Amazon Athena Data Catalog, AWS Glue Data Catalog or Apache Hive metastore (such as that used by Amazon EMR) can be used to create an external schema. External tables are read-only and won't allow you to perform insert, update or delete operations.
- Make sure that the data files in Amazon S3 and the Amazon Redshift cluster are in the same AWS region to use spectrum.

- Redshift can create materialized views that reference external data sources such as Amazon S3 via Spectrum, or data in Aurora or RDS for PostgreSQL via federated queries. Materialized views can significantly boost query performance for repeated and predictable analytical workloads such as dashboarding, queries from business intelligence (BI) tools, and extract, load, transform (ELT) data processing.
- RA3 nodes configuration lets you scale compute and storage independently for fast query performance and lower costs. Redshift clusters working with RA3 nodes also have Advanced Query Accelerator (AQUA) capability, which is a distributed and hardware accelerated cache that accelerates query performance for repetitive queries and queries with a common pattern.
- Users should choose the traditional provisioned option for predictable workloads and the serverless option for demanding and unpredictable workloads. Redshift Serverless is also a good option for those who do not have experience in data warehouse management. With the serverless options, users don't have to worry about setting up, configuring, managing clusters or tuning the warehouse. You only pay for what you use.
- Redshift also provides a capability called data sharing that lets you share data across Redshift clusters without copying data around. Data sharing builds on Amazon Redshift RA3 managed storage, which decouples storage and compute, allowing either of them to scale independently. Using data sharing, customers can share data between clusters within an AWS account, across AWS accounts and across regions.
- Redshift integrates with CloudWatch to provide monitoring capabilities for compute utilization, storage utilization and read/write traffic to the cluster. Users can also refer to the AWS management console to monitor queries, cluster performance, maintenance status, live and historical user query log, and other useful metrics.
- AWS Data Exchange for Amazon Redshift is a feature that can be used to find, subscribe and access third-party data.

- Amazon Redshift is ideal for online analytical processing (OLAP) using your existing business intelligence tools. Organizations are using Amazon Redshift to:
  - Analyze global sales data for multiple products
  - Store historical stock trade data
  - Analyze ad impressions and clicks
  - Aggregate gaming data
  - Analyze social trends
  - Measure clinical quality, operation efficiency, and financial performance in healthcare
  - Analyze data across the data lake (S3) and Amazon Redshift
- Amazon Redshift uses a variety of innovations to obtain very high performance on datasets ranging in size from hundreds of gigabytes to a petabyte or more. It uses columnar storage, data compression and zone maps to reduce the amount of I/O needed to perform queries. Amazon Redshift has a massively parallel processing (MPP) architecture, parallelizing and distributing SQL operations to take advantage of all available resources. The underlying hardware is designed for high performance data processing, using local attached storage to maximize throughput between the CPUs and drives, and a 10 GigE mesh network to maximize throughput between nodes. Performance can be tuned based on your data warehousing needs: AWS offers Dense Compute (DC2) with SSD drives as well as dense storage (DS) options.
- Automatic Table Optimization (ATO) is a self-tuning capability that helps you achieve the performance benefits of creating optimal sort and distribution keys without manual effort. ATO observes how queries interact with tables and uses machine learning (ML) to select the best sort and distribution keys to optimize performance for the cluster's workload.
- Additional features such as Automatic VACUUM DELETE, Automatic Table Sort, and Automatic Analyze eliminate the need for manual maintenance and tuning of Redshift clusters to get the best performance for new clusters and production workloads.

- Workload management allows you to route queries to a set of defined queues to manage the concurrency and resource utilization of the cluster. Today, Amazon Redshift has both automatic and manual configuration types. With manual workload management (WLM) configurations, you're responsible for defining the amount of memory allocated to each queue and the maximum number of queries, each of which gets a fraction of that memory, which can run in each of their queues. Manual WLM configurations don't adapt to changes in your workload and require an intimate knowledge of your queries' resource utilization to get right. Amazon Redshift automatic WLM doesn't require you to define the memory utilization or concurrency for queues. Instead, it adjusts the concurrency dynamically to optimize for throughput.
- The number and type of nodes can be changed from the console or an API call. While resizing, Amazon Redshift provisions your existing cluster into read-only mode, provisions a new cluster and then copies the data from the old cluster to the new cluster. Once the copy is complete, Redshift automatically points to the new cluster and removes the old one.
- Redshift can also automatically detect and replace a failed node in the data warehouse.

Clearing Confusion Between Amazon Redshift and Amazon Athena

AWS users often compare Athena with Redshift, but they each have different use cases and are by foundation very different. Table 6 lists the differences between the two:

Table 6: Redshift Versus Athena

(Enlarged table in Appendix)

Differentiating Factor	*Redshift*	*Athena*
Use Case	Data warehousing — ideal for collecting data from multiple repositories into a single place for long periods of time, to build sophisticated business reports from historical data	Data lake engine — easy to run interactive queries against data directly in Amazon S3
Schema Declaration	Schema-on-write	Schema-on-read
Management	Managed service (serverless for Redshift Serverless)	Serverless
Data Storage	Data stored primarily in Redshift nodes (for better performance) or S3	Data stored primarily in S3 (federated query for more sources)
Data Structure	Structured Data* (can support semistructured using SUPER datatype or via Redshift Spectrum)	Both structured and unstructured
Foundation	PostgreSQL	Presto
Maintenance Downtime	Yes (no for Redshift Serverless)	No
Stored Procedure	Supported	Not Supported

Source: Gartner (November 2022)

Amazon Relational Database Service

- Amazon Relational Database Service (RDS) is a managed database service for operating relational databases in the cloud.
- It supports:
  - Commercial databases — Oracle And SQL Server
  - Open-source databases — MySQL, MariaDB Server and PostgreSQL
  - Amazon's own cloud-native engine — Amazon Aurora
- Tasks managed by RDS:
  - Setting up relational databases
  - Provisioning the infrastructure capacity for required configuration
  - Performs automated backups
  - Software patching

- Supports replication to enhance availability and improve durability.
- No upfront costs; pay-as-you-go model.

Amazon Aurora

- Amazon Aurora is a fully managed MySQL and PostgreSQL compatible relational database offering high performance at auto-scale (up to 128 TB per database instance) but with the ease of use and cost-effectiveness of open-source databases.
- Aurora is part of the managed database service Amazon Relational Database Service (Amazon RDS).
- Storage and compute are decoupled.
- Users can try Aurora Serverless, which provides autoscaling of compute based on demand.
- It provides a high degree of data durability by storing six copies of data across three Availability Zones (AZs).
- Like RDS, aurora also automates hardware provisioning, database setup, patching and backups.
- The minimum storage allowed for Aurora databases is 10GB and can scale up to 128TB in 10-GB increments. Users don’t have to provision storage in advance.
- Customers can only use the InnoDB storage engine for persistent data. For analytical workloads, Aurora MySQL supports a parallel query feature. This provides the ability to pushdown and distribute the computational load of a single query across thousands of CPUs in Aurora’s storage layer.
- Provides additional features, such as fast database cloning, serverless and backtrack, that are not available on RDS engines.
- Amazon Aurora offers read replicas with lower replication lag on the order of tens of milliseconds and can be leveraged to scale reads.
- Use Amazon Aurora Global Database to replicate the data to other regions for disaster recovery (DR) and low-latency global reads.
- Use Amazon Aurora Serverless if your workload includes infrequent or unpredictable traffic patterns.

- Use Amazon Aurora Auto Scaling for horizontal scaling, which would have to be done manually with Amazon RDS.
- For read-intensive workloads, Amazon Aurora is a good fit.
- If your workload requires high concurrence, high durability and on-demand scaling with enterprise features, leverage Aurora.
- Use Aurora if you're looking to adopt AWS-developed services such as Backtrack, Parallel Query or Aurora Machine Learning.

Amazon DynamoDB

- DynamoDB is a fast, flexible NoSQL database service for single-digit millisecond performance at any scale.
- Like with other AWS services, it is fully managed; users don't have to manage the database software and hardware provisioning themselves.
- Users can choose from three consistency models:
  - Eventually consistent reads (default) — Maximum read throughput; might not reflect results of recently completed writes
  - Strong consistent reads — Reflects all writes that received a successful response before the read
  - ACID transactions — Can be used for use cases that require coordinated inserts, deletes or updates to multiple items as part of a single logical business operation

Governance and Security

Lake Formation

- Lake Formation is a fully managed data lake service that makes it easy for customers to ingest, clean, catalog, transform, and secure their data and make it available for analysis and ML (see Table 7). There is no additional charge for using Lake Formation; it builds capabilities in AWS Glue and uses the Glue data catalog, crawlers and jobs.
- It integrates with multiple other AWS services like AWS CloudTrail, AWS IAM, Amazon CloudWatch, Amazon Athena, Amazon EMR and Amazon Redshift. Lake Formation uses AWS Glue as the transformation engine.

- It provides a centralized approach for data engineers, data analysts and data scientists to utilize the data lake in a secure and governed manner. While AWS Identity and Access Management (IAM) permissions provide coarse-grained security for Glue metadata at the table level, and S3 at the buckets and object level, Lake Formation takes it a level deeper. It provides fine-grained access control on this data, including column level and row level. This allows different types of users to access the same underlying data without duplicating the data with restricted columns removed. Instead, users can now dynamically apply access control policies that enforce their governance requirements.
- Analytics services like AWS Glue, Amazon EMR, Amazon Athena, Redshift Spectrum and Amazon QuickSight follow the Lake Formation permission model.

Table 7: Lake Formation Use Cases

(Enlarged table in Appendix)

Use Case	Description
Data Discovery	▪ Crawls (using Glue Crawlers) and reads your data sources to extract technical metadata (such as schema definitions) and creates an accessible searchable catalog ▪ Discovers data sources provided access by your IAM policies ▪ Can also access on-premises data sources using JDBC connections for Oracle, MySQL, PostgreSQL, SQL Server and MariaDB
Data Ingestion	▪ Blueprints can be used to ingest data into data lake ▪ Glue workflows are created by Lake Formation to crawl source tables and migrate the data to S3
Cataloging	▪ Central catalog for all your data ▪ Single unified experience for accessing data ▪ Provides the ability to manually add labels at table or column level apart from the automated ones created by the crawlers ▪ Searches data by name, contents, sensitive or other custom label
Data Transformation:	▪ Allows setup of transformation jobs ▪ Allows removal of duplicates and linking of matching records through machine learning (ML) algorithms ▪ Job orchestration among AWS services and monitoring capabilities
Data Governance and Security	▪ Catalog data for access by analytics tools ▪ Allows configuration of data access and security policies along with integration with IAM ▪ Audit and control access from AWS analytic and ML services ▪ Allows enforcement of data encryption leveraging the S3 capabilities ▪ Automatic server-side encryption with keys managed by the AWS Key Management Service (KMS) ▪ Comprehensive audit logs with CloudTrail ▪ Central catalog that can specify, grant and revoke permissions on tables; the same information is available to multiple accounts, groups and services ▪ Column-level, row-level and cell-level security ▪ To protect against malicious insider deletions, S3 encrypts data in transit when replicating across regions
Self-Service	▪ Allows labeling of data with business metadata ▪ Allows designation of data owners like data stewards and business units by adding a custom field

Source: Gartner (November 2022)

There are two main types of permissions in AWS Lake Formation:

- Metadata access — Lake Formation has access to data catalog resources.
- Underlying data access — Lake Formation has permissions on locations in Amazon S3.

For both, Lake Formation uses a combination of its own permissions and IAM permissions. The Lake Formation permissions model is implemented as DBMS-style GRANT/REVOKE commands, such as:

Grant SELECT on tableName to username

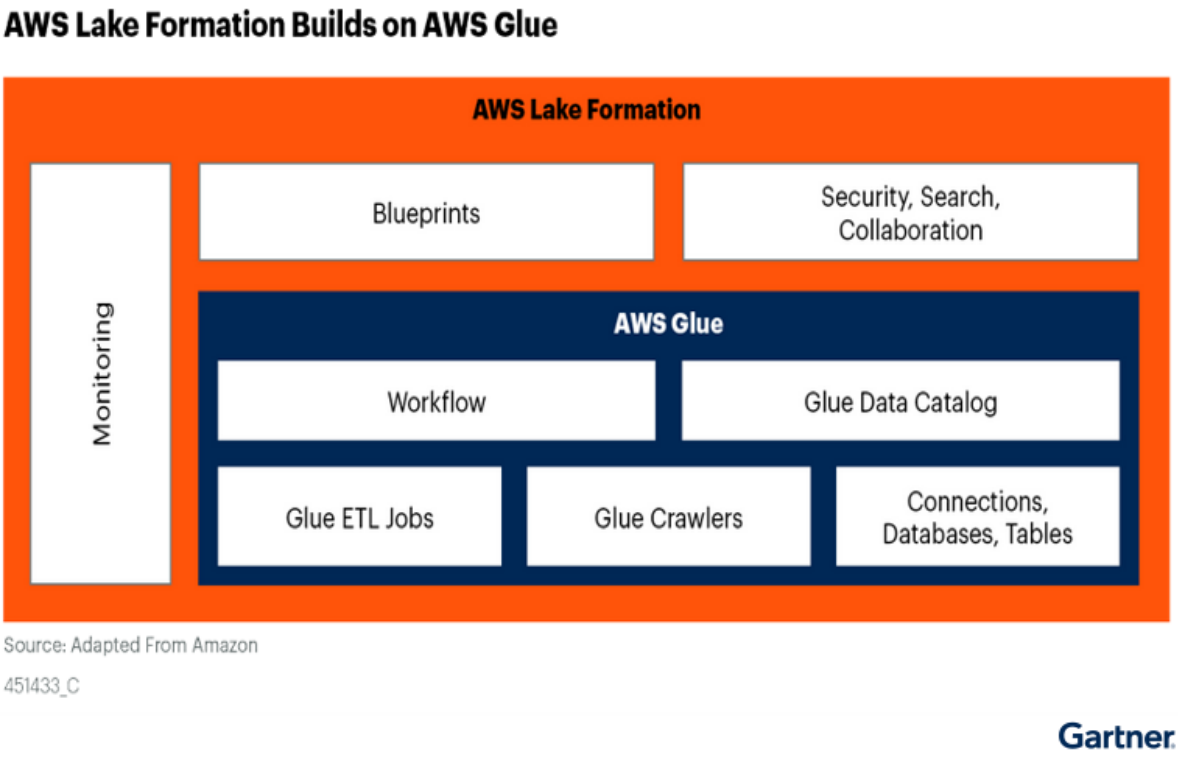
To access Data Catalog resources or underlying data, both Lake Formation and IAM permission checks must be passed.

Lake Formation Relationship With Glue

AWS Lake Formation is a service built on top of Glue to build and secure data lakes. It offers the following features:

- Blueprints: These are built-in templates that ease the process of ingesting data from common data sources. Blueprints create a workflow that is run as a Glue task. Blueprints exist for well-known RDBMSs such as Oracle and MySQL, and for common log formats like AWS CloudTrail, ALB and Amazon Elastic Load Balancing.
- Simplified permissions management: Glue and Lake Formation use the same data catalog; they are the same. Lake Formation provides an additional authorization layer to catalog resources (e.g., databases, tables and columns). So instead of just IAM policies governing access to catalog resources, both IAM and Lake Formation policies now govern access. The expected usage is coarse-grained IAM policies and fine-grained Lake Formation policies on the data lake.
- Consistent access control across analytic services: Lake Formation permissions are honored by compliant analytics services such as Glue, EMR, Athena, Redshift Spectrum and QuickSight. So you have a single pane of glass to define fine-grained access policies at table, row and column level that are honored across compliant analytic services.
- ML transforms: Lake Formation provides more sophisticated transformation jobs than Glue via ML transforms. These transforms are used to match and deduplicate records across diverse datasets.
- Monitoring: Workflow and ETL jobs are monitored and audit trails provided that can be analyzed in CloudTrail using Amazon Athena. Lake Formation UI monitors access to the data and permissions in real time (see Figure 6).

Figure 6: AWS Lake Formation Builds on AWS Glue



Lake Formation gives you a central console where you can discover data sources, set up transformation jobs to move data to an S3 data lake, remove duplicates and match records, catalog data for access by analytic tools, configure data access and security policies, and audit and control access from AWS analytic and machine learning services. But it has the following anti-patterns:

Managing unstructured data: Lake Formation doesn't manage security for unstructured data.

Business catalog: Lake Formation has decent capabilities to label data with tags and designate data owners, but it isn't as feature-rich as advanced catalogs present in the AWS Marketplace.

Managing permissions for open-source and third-party-data big data components: AWS Lake Formation is not integrated with Presto, HBase, Trino, Databricks and some EMR components.

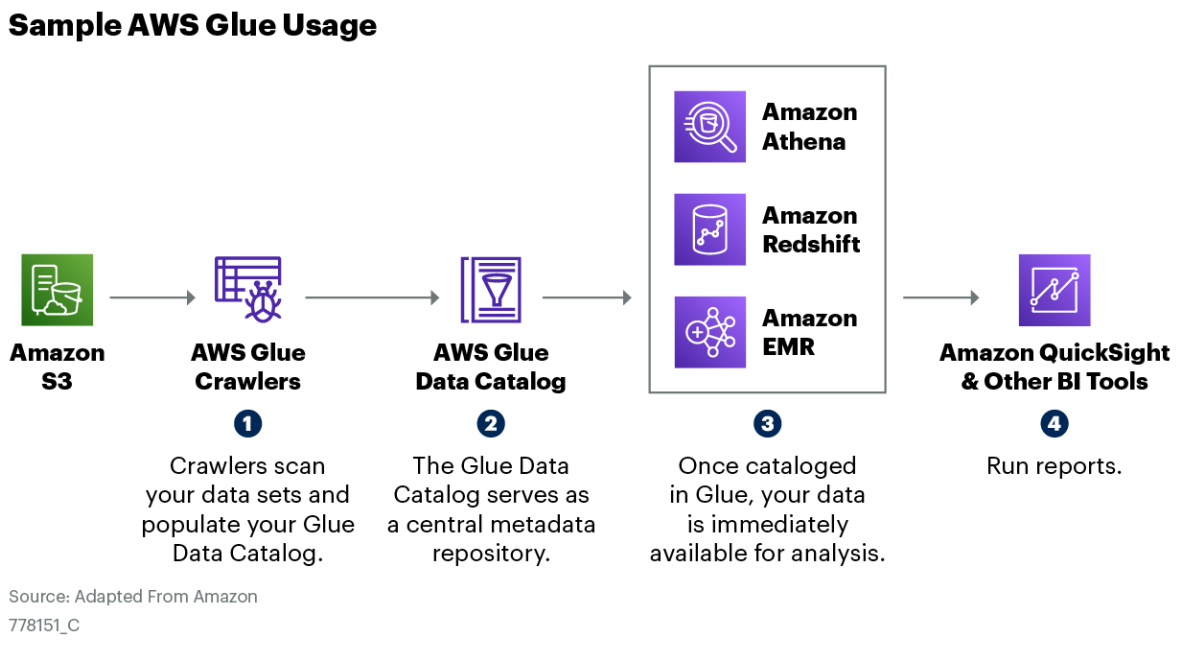
Real-time analytics: AWS Lake Formation is not integrated with Amazon Kinesis or OpenSearch service.

These limitations were true at the time of the release of this research. Please refer to Notes and Restrictions for Governed Tables for the updated list.

Glue Data Catalog

- Glue data catalog is the central metadata repository for storing structural and operational metadata from different data assets. It is Hive metastore compatible and is a drop-in replacement for the Apache Hive Metastore for big data applications running on Amazon EMR.
- Glue data catalog provides seamless integration with Amazon Athena, Amazon EMR and Redshift Spectrum to provide schema definition and ease of use. Adding table definitions inside the Glue data catalog makes them available in all these services, providing a common view among all.
- Glue crawlers can automatically scan schemas from data sources to populate the data catalog with up-to-date information. These details can be added or updated manually by using the Glue console or calling an API. Hive DDL statements are also supported via the Amazon Athena console or a Hive client on Amazon EMR. Users with existing Apache Hive metastore can bulk import the metadata in AWS Glue Data Catalog (see Figure 7).

Figure 7: Sample AWS Glue Usage



Source: Adapted From Amazon

778151_C

AWS Identity and Access Management

AWS Identity and Access Management (IAM) allows organizations to securely manage access to its various services and resources. AWS IAM does not incur any charges because its sole purpose is to enable access to other services. Some of the best practices to govern resources include:

- Limit the number of principals (users, roles, compute resources) with privileged permissions using the IAM policies. IAM policy is attached to each user or group of users or roles. These policies are global and apply to all regions. Develop fine-grained policies that consider data sovereignty and localization requirements; for example, create policies that restrict data processing and storage to a specific jurisdiction or geographic region.
- Identify unused roles and take measures to reduce overly permissive policies. Features such as "role last used date," Access Advisor and Access Analyzer can be used to identify areas where customers can reduce the scope of permission and resource policies.
- Ensure the data locality governance requirement by defining a policy that ensures that the data stays local to the specified region.
- Control data access at the network level. IAM policy can specify which regions are allowed and which are rejected. A good recommendation is to give "least privileges." Organizations can "allow list" or "block list" IP addresses. In addition, organizations can set up disaster recovery (DR) on their data, preferably in a different region and under a different account.
- Control data access at the storage level. IAM policy can allow read-only access to certain S3 buckets and restrict writes or deletes. Another best practice is to turn on versioning on Amazon S3 objects.
- Control data access at the data security level. IAM policy can deny unencrypted writes to Amazon S3 objects. Data security policies can also be applied at the application level; for example, EMR or SageMaker. AWS organization policies can be useful to enforce this kind of practice across a customer's entire environment.
- Secure your encryption keys. AWS provides fully managed Key Management Service (KMS) with features such as key rotation. However, some Gartner clients prefer to manage keys on their own and even have it on-premises.

Orchestration

AWS Step Functions

- AWS Step Functions is a fully managed service that can be used to orchestrate data pipelines using visual workflows.
- Step Functions provides a graphical console to arrange and visualize the components of your application as a series of steps. You can change and add steps without even writing code.
- Using Step Functions, you can arrange and visualize your application's components as steps.
- In Step Functions, every step is automatically triggered and tracked, and errors are retried as necessary, ensuring that your application runs as expected and in order.
- AWS Step Functions state machines are defined in JSON using the declarative Amazon States Language.
- AWS Step Functions sends metrics to Amazon CloudWatch and AWS CloudTrail. This helps in monitoring and troubleshooting by checking logs.
- As your workload changes, AWS Step Functions automatically scales the operations and underlying compute for you.
- Supports over 200 services, including Lambda, ECS, Fargate, Batch, DynamoDB, SNS, SQS, SageMaker, EventBridge or EMR.
- Can also be used to automate IT and business processes.

Amazon Managed Workflows for Apache Airflow

- Amazon Managed Workflows for Apache Airflow (MWAA) is a managed Apache Airflow service that frees you from managing, configuring and scaling the Airflow environment.
- It provides AWS-backed logging and monitoring capabilities along with automatic scaling.
- Users can execute their existing Airflow workflows on Amazon MWAA and interact with their environment programmatically using the AWS console, API and Command Line Interface (CLI).
- It can even receive system metrics from on-premises Airflow instances. These metrics can then be viewed from Amazon CloudWatch.

- Users can directly access each Airflow environment through the Amazon MWAA management console and the Airflow UI.
- MWAA can be preferred over AWS Step Functions if users prioritize open source and portability.

Strengths

- The AWS ecosystem for data management is extremely well-tested with customers across a variety of domains. AWS provides many highly customizable services across areas like ingestion, processing, and integration with diverse features and capabilities.
- The AWS ecosystem includes best-practice out-of-the-box templates and blueprints. Many AWS services are based on open-source projects where skills are readily available, and to a large extent can ease the high upskilling and learning associated with onboarding.
- AWS holds the edge for organizations with web-scale applications that support a lot of users. When looking for a platform that is feature-rich and scalable, AWS is a good choice and continues to offer new features and updates to attract more customers.

Weaknesses

- AWS does not include enterprise support by default. It can (and should) be purchased as an add-on service.
- While AWS does offer several mature options in other categories, it still lacks capabilities and tools in the governance, orchestration, data ops and data observability space.
- Most AWS services are region-oriented, and when AWS provides multiregion services like DynamoDB global tables and S3 Cross-Region Replication, you need to handle eventual consistency yourself.
- AWS has “poor cohesion” across its massive array of services, including inconsistencies among products and a tendency to demand an “application builder’s mindset” that can be daunting to enterprises.

Guidance

- Winning the cloud strategy depends on not just cloud hardware but embracing managed services, serverless, and the entirety of cloud services so that the infrastructure stack functions reliably without managing virtual servers, storage and networking. Simple infrastructure as a service (IaaS) alone on the cloud is legacy now.
- Leverage AWS-native tools for ingestion, processing, data formats, data sources and network boundaries before investing in third-party tools. AWS tools are optimized in terms of cost, bandwidth utilization and speed to work across network boundaries and provide almost all possible solutions required by most organizations to ingest, integrate and process data to the cloud.
- Data governance and data cataloging tools within the AWS ecosystem are still evolving compared to more mature commercial offerings. While metadata management features and fine-grained security controls are available within AWS Glue and AWS Lake Formation respectively, organizations should look to invest in third-party tools, especially for the business data catalog and to track data lineage.
- Keep an eye on the costs. Use the right tools within the AWS ecosystem to monitor costs and budget. Organizations can get carried away by the plethora of choices in the cloud world. This can easily lead to a cluttered architecture and lack of governance in the architecture. Organizations need to ensure they have the right governance in place to manage the diverse services. It is highly recommended that organizations use the tools provided by AWS to monitor usage, resources and costs, and plan their budgets accordingly to prevent paying high costs for suboptimal system usage in the cloud.

Recommended by the Authors

Some documents may not be available as part of your current Gartner subscription.

Solution Scorecard for AWS Cloud Analytical Data Stores

The Impacts of Emerging Cloud Data Ecosystems: An Architectural Perspective

Innovation Insight: Data Ecosystems Will Reshape the Data Management Market

Solution Path for Building Modern Analytics and BI Architectures

Vendor Rating: Amazon

Table 1: AWS Data Ingestion Capabilities

Capability	*Glue Features*
Volume	Maximum capacity is expressed in terms of the number of AWS Glue data processing units (DPUs) that can be allocated when a job runs. For more info go to AWS Glue Jobs API.

Capability	*Glue Features*
Source Types	AWS Glue natively supports data stored in Amazon Aurora, Amazon RDS for MySQL, Amazon RDS for Oracle, Amazon RDS for PostgreSQL, Amazon RDS for SQL Server, Amazon Redshift, Amazon DynamoDB and Amazon S3, as well as MySQL, Oracle, Microsoft SQL Server, and PostgreSQL databases in your Virtual Private Cloud (Amazon VPC) running on Amazon EC2. AWS Glue also supports data streams from Amazon Managed Streaming for Apache Kafka, Amazon Kinesis Data Streams, and Apache Kafka. You can use a crawler to populate the AWS Glue Data Catalog with tables. This is the primary method used by most AWS Glue users. A crawler can crawl multiple data stores in a single run. Upon completion, the crawler creates or updates one or more tables in your Data Catalog. Extract, transform, and load (ETL) jobs that you define in AWS Glue use these Data Catalog tables as sources and targets. The ETL job reads from and writes to the data stores that are specified in the source and target Data Catalog tables. For a list of file-based and table-based data stores that crawlers can crawl, go to Which Data Stores Can I Crawl? You can also write custom Scala or Python code and import custom libraries and Jar files into your AWS Glue ETL jobs to access data sources not natively supported by AWS Glue. For more details on importing custom libraries, refer to our AWS Glue Documentation.

Capability	*Glue Features*
Source Formats	AWS Glue supports many file formats, including flat files, TXT, JSON, CSV, Parquet, ORC, Avro, and XML. AWS Glue differentiates its capability with its extensions to Apache Spark called DynamicFrame, which allows developers to handle frequently changing schemas gracefully. AWS Glue also supports modern transactional formats such as Apache Hudi, Apache Iceberg and Delta Lake formats for running transactional workloads in data lakes. Additional file formats can be supported by a number of third-party libraries and offerings from the AWS Marketplace.
Targets	AWS Glue supports the following data targets: Amazon S3, Amazon Relational Database Service (Amazon RDS), third-party JDBC-accessible databases, and MongoDB and Amazon DocumentDB (with MongoDB compatibility).
Extensible Connectors	AWS Glue supports connecting to many relational and nonrelational database systems via Glue's own connectors, connectors offered in the AWS Marketplace, and the ability to use publicly available connection libraries. For example, RDBMS support includes PostgreSQL, MySQL, Oracle, Microsoft SQL Server, Google BigQuery, Vertica, Snowflake, IBM Db2 Database, and more. AWS Glue natively supports reading from and writing to nonrelational databases including Amazon DynamoDB, Amazon DocumentDB, and MongoDB, and with connectors from the AWS Marketplace to Apache Cassandra, Apache HBase, Azure Cosmos DB, and more. Connectors available in the AWS Marketplace also offer connectivity to legacy systems, mainframes, and ERP/CRM systems.

Capability	*Glue Features*
Batch Data Capture	AWS Glue provides a number of connectors, data transformations, and load utilities to process data in batch/microbatch and streaming modes using AWS Glue Studio (a visual IDE for ETL development), AWS Glue Interactive Sessions (for notebook-based interface for data interaction development), and AWS Glue DataBrew (for no-code data wrangling experience). AWS Glue offers a rich set of transformation capabilities to process both structured and semistructured data.
Incremental Data (CDC)	AWS Glue streaming ETL allows customers to process real-time data from Amazon Kinesis Data Streams, Apache Kafka, or Amazon Managed Streaming for Apache Kafka (Amazon MSK) for both real-time and near-real-time streaming ETL. AWS Glue Streaming supports autoscaling to keep costs low and in line with data volume. In addition, AWS Glue integrates with AWS Database Migration Service (AWS DMS) to process and apply change data capture (CDC) records. Customers can use Amazon DMS to set up CDC on sources and use Amazon Simple Storage Service (Amazon S3) to store data or Amazon Kinesis Data Streams to stream the changes. AWS Glue then crawls these data sources and processes the data in real-time or batch mode, depending on customer use case. Users can incorporate the capabilities delivered by the Job Bookmarks feature to track data that has been previously processed.
Velocity of Data	AWS Glue helps customers build extract, transform, and load (ETL) pipelines operating at varying data ingestion frequencies.

Source: Gartner (November 2022)

Table 2: Comparing AWS Streaming Options

Feature	*Kinesis*	*MSK*	*SQS*
Message Size Limit	1 MB	Dependent on instance type, replication factor, throughput, etc. 8 MB (for serverless)	256 KB
Retention Period	7 days (365 with long-term retention, optional cost)	Unlimited	14 days
Decision Complexity	Easy (with On-Demand mode, capacity scales automatically)	Medium (need to choose between serverless that requires minimal setup and provisioned that requires deciding between broker types and numbers)	Easy (need to create queues, no decision to make)
Multiple Consumer Support	Yes	Yes	No
Replay Support	Yes	Yes	No
Analytics Capability	Yes	Yes	No
Message Ordering	Yes	Yes	Only for FIFO
Cloud Agnostic	No	Yes	No
Granularity Level	Shards	Kakfa partition	Messages
Checkpointing	With Amazon DynamoDB	Yes	Yes

Feature	*Kinesis*	*MSK*	*SQS*
Managed/Serverless	Serverless	Both	Managed
Customization	Minimum	Maximum	Not required

Source: Gartner (November 2022)

Table 3: Differences Between EMR Serverless and Amazon EMR on EC2

Feature	*EMR Serverless*	*Amazon EMR on EC2*
Resilience to Availability Zone failures	Y	N
Open-source frameworks supported	Apache Spark and Apache Hive	Apache Hive, Apache Pig, Apache Sqoop, Apache Spark, Apache Tez, Apache HBase, Apache Oozie, Presto, Trino, etc.
Support for fine-grained authorization using AWS Lake Formation	N	Y
Integration with Apache Ranger for table- and column-level permission control	N	Y
Customize operating system images	N	Y
Customize and load additional libraries and dependencies	Y	Y
Run workloads from SageMaker Studio as part of machine learning (ML) workflow	N	Y
Connect to self-hosted Jupyter Notebooks	N	Y
Build pipelines using Apache Airflow and Amazon Managed Workflows for Apache Airflow (MWAA)	Y	Y

Feature	*EMR Serverless*	*Amazon EMR on EC2*
Build and orchestrate pipelines using AWS Step Functions	N	Y

Source: Gartner (November 2022)

Table 4: A Comparison of Data Transformation and Processing Services on AWS

Feature	*Lambda*	*Glue*	*EMR*	*Amazon Data Pipeline*
User Interface	AWS Lambda console (Code interface), AWS CLI, AWS SDK	Glue console (Visual- and Code-based)	EMR console (Code interface), API, SDK or CLI	Console (Visual), API and CLI
Managed/Serverless	Serverless	Serverless	Managed and serverless	Managed
Ease of Use	Code extensive	Easy to use	Code extensive	Easy to use
Batch/Streaming	Both	Both	Both	Batch
Orchestration	Built-in, Also can be invoked from AWS Step Functions and Apache Airflow	Built-in	Can be achieved with another service like AWS Step Functions or Airflow (but may not work for EMR Serverless)	Built-in
Built-In Transformations	No	Yes	No	Limited
Compute Engine	Any	Spark	Apache Spark, Apache Hive and Presto	Spark, Hive, Pig, etc.

Feature	*Lambda*	*Glue*	*EMR*	*Amazon Data Pipeline*
Specialization	Run any type of code without the need of provisioning any server	Easy-to-use ETL and cataloging service in AWS with loads of features and integration with other services	Interactive analysis and machine learning using Apache Spark, Hive and Presto. Provides more control over configurations and is ideal for cloud migration of on-premises Hadoop environments	Data movement and processing between AWS services
Anti-Patterns	Has a 15-minute time limit on functions	Doesn't offer extensive changes to configuration parameters since it's a fully managed service	Not as easy to use as Glue	Suitable for batch workloads only

Source: Gartner (November 2022)

Table 5: AWS Options for Decision Tree

Number Key	*AWS -Native*
1	Amazon RDS
2	Amazon Aurora
3	Amazon Redshift
4	Amazon DynamoDB
5	Amazon DocumentDB
6	Amazon Keyspaces
7	Amazon Neptune
8	Amazon EMR
9	Amazon S3
10	Kinesis
11	Amazon ElastiCache, Amazon MemoryDB for Redis

Source: Gartner (November 2022)

Table 6: Redshift Versus Athena

Differentiating Factor	*Redshift*	*Athena*
Use Case	Data warehousing — ideal for collecting data from multiple repositories into a single place for long periods of time, to build sophisticated business reports from historical data	Data lake engine — easy to run interactive queries against data directly in Amazon S3
Schema Declaration	Schema-on-write	Schema-on-read
Management	Managed service (serverless for Redshift Serverless)	Serverless
Data Storage	Data stored primarily in Redshift nodes (for better performance) or S3	Data stored primarily in S3 (federated query for more sources)
Data Structure	Structured Data* (can support semistructured using SUPER datatype or via Redshift Spectrum)	Both structured and unstructured
Foundation	PostgreSQL	Presto
Maintenance Downtime	Yes (no for Redshift Serverless)	No
Stored Procedure	Supported	Not Supported

Source: Gartner (November 2022)

Table 7: Lake Formation Use Cases

Use Case	*Description*
Data Discovery	■ Crawls (using Glue Crawlers) and reads your data sources to extract technical metadata (such as schema definitions) and creates an accessible searchable catalog ■ Discovers data sources provided access by your IAM policies ■ Can also access on-premises data sources using JDBC connections for Oracle, MySQL, PostgreSQL, SQL Server and MariaDB
Data Ingestion	■ Blueprints can be used to ingest data into data lake ■ Glue workflows are created by Lake Formation to crawl source tables and migrate the data to S3
Cataloging	■ Central catalog for all your data ■ Single unified experience for accessing data ■ Provides the ability to manually add labels at table or column level apart from the automated ones created by the crawlers ■ Searches data by name, contents, sensitive or other custom label

Use Case	*Description*
Data Transformation:	■ Allows setup of transformation jobs ■ Allows removal of duplicates and linking of matching records through machine learning (ML) algorithms ■ Job orchestration among AWS services and monitoring capabilities
Data Governance and Security	■ Catalog data for access by analytics tools ■ Allows configuration of data access and security policies along with integration with IAM ■ Audit and control access from AWS analytic and ML services ■ Allows enforcement of data encryption leveraging the S3 capabilities ■ Automatic server-side encryption with keys managed by the AWS Key Management Service (KMS) ■ Comprehensive audit logs with CloudTrail ■ Central catalog that can specify, grant and revoke permissions on tables; the same information is available to multiple accounts, groups and services ■ Column-level, row-level and cell-level security ■ To protect against malicious insider deletions, S3 encrypts data in transit when replicating across regions

Use Case	*Description*
Self-Service	■ Allows labeling of data with business metadata ■ Allows designation of data owners like data stewards and business units by adding a custom field

Source: Gartner (November 2022)