Build Modern Data Streaming Analytics Architectures on AWS

AWS Whitepaper

Build Modern Data Streaming Analytics Architectures on AWS: AWS Whitepaper

Table of Contents

Build Modern Data Streaming Analytics Architectures on AWS

Publication date: **May 17, 2022** (*Document revisions* (p. 21))

Abstract

Modern data architecture is about using the right tool for the job. It acknowledges that a "one size fits all" approach leads to compromise, and a solution that is not optimized for anyone. With modern data architecture, customers can integrate a data lake, data warehouses, and purpose-built data services, together with a unified governance layer, to create a system without boundaries, enabling data-driven decisions.

When building a modern data architecture, there is sometimes the need for data to flow with low latency between components to power real-time decisions. This whitepaper helps cloud architects, data scientists, and developers to design and building modern data streaming architectures that can quickly generate insights using Amazon Web Services (AWS) streaming services such Amazon Kinesis Data Streams, Amazon Kinesis Data Firehose, Amazon Kinesis Data Analytics, and Amazon Managed Streaming for Apache Kafka (Amazon MSK).

Are you Well-Architected?

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems in the cloud. The six pillars of the Framework allow you to learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. Using the AWS Well-Architected Tool, available at no charge in the AWS Management Console (sign-in required), you can review your workloads against these best practices by answering a set of questions for each pillar.

In the Data Analytics Lens, we focus on how to design, deploy, and architect your data analytics workloads in the AWS Cloud. This lens adds to the best practices described in the Well-Architected Framework.

For more expert guidance and best practices for your cloud architecture—reference architecture deployments, diagrams, and whitepapers—refer to the AWS Architecture Center.

Introduction

In the next three years, there will be more data created than was created in the prior 30 years combined. While most people think the data explosion is occurring only in certain industries (such as social media), it actually affects everyone and, as data professionals, AWS is increasingly being asked to build systems that handle a variety of data.

On-premises tools and data stores can be used, but do not support pay-as-you-go, and can turn out to be more expensive for this kind of scale. Organizations need to easily access and analyze all types of data such as structured data, semi-structured data, unstructured data, and real-time streaming data to

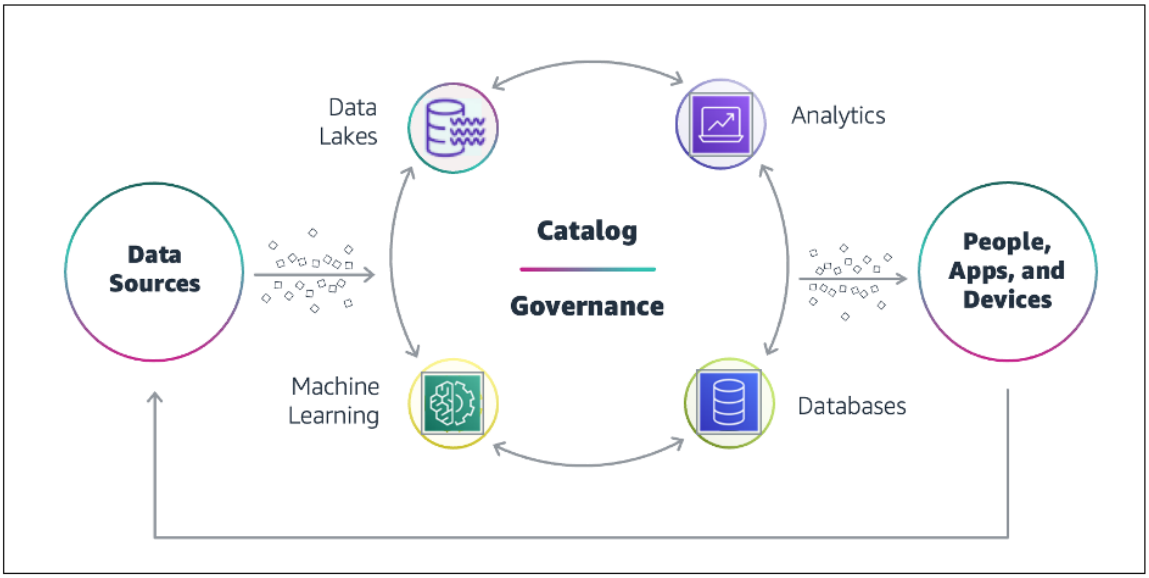
perform comprehensive and efficient analytics. Organizations need to easily break down data silos to gain new insights and build better experiences. These organizations want a modern data architecture to collect, store, organize, and process valuable data, make it available in a secure way, and enable applications to derive low latency, near real-time insights.

This whitepaper presents how customers can implement modern data architecture and realize the benefits of low latency insights with streaming technologies. Modern data architecture on AWS enables you to collect, manage, process, and analyze all your real-time streaming data in a simple and integrated fashion. Modern data architecture also allows you to use all your data for a variety of use cases, such as interactive SQL, business intelligence (BI), machine learning (ML), streaming analytics, and big data analytics.

The whitepaper first discusses the concept of the modern data architecture approach, then presents three modern data architecture data movement patterns to derive insights from your near real-time streaming data, using AWS purpose-built analytics services.

What is a modern data architecture?

Modern data architecture on AWS allows you to build a scalable data lake, and use a broad and deep collection of purpose-built data services that provide the performance required for use cases such as low latency streaming analytics, interactive dashboards, log analytics, big data processing, and data warehousing. It enables you to easily move data between the data lake and purpose-built data services, and to set up governance and compliance in a unified way to secure, monitor, and manage access to your data. AWS calls this modern, cloud-based analytics architecture the *modern data architecture*.



Modern data architecture on AWS

AWS provides a broad platform of managed services to help you build, secure, and seamlessly scale end-to-end data analytics applications quickly, using the modern data architecture approach. There is no hardware to procure, no infrastructure to maintain and scale—only what you need to collect, store, process, and analyze your data. AWS offers analytical solutions specifically designed to handle this growing amount of data, and provide insight into your business.

Working with streaming data on AWS

Customers want the freedom to move data between their centralized data lakes and the surrounding purpose-built data services in a seamless, secure, and compliant way, to get insights with speed and agility.

For example, many organizations store streaming data in a data lake for offline analytics, and a portion of that data lake data can be moved out to a data warehouse for daily reporting. Think of this concept as *inside-out data movement*.

You can also move data in the other direction: from the outside-in. For example, you can move streaming data from non-relational databases into the data lake for product recommendation by using ML algorithms. Think of this concept as *outside-in data movement*.

In other situations, you may want to move data from one purpose-built data store to another. For example, you may copy the product catalog data stored in your database to your search service to make it easier to look through your product catalog, and offload the search queries from the database. Think of this concept as data movement *around the perimeter*.

The volume of data produced is increasing rapidly, and the data is coming from a wide variety of sources, in a variety of forms. The data is coming at lightning speeds due to an explosive growth of real-time data sources. Organizations create value by making decisions from their data. The faster they can make decisions and take action, the better they perform against their competitors. The value of data diminishes over time. To get the most value from the data, it must be processed at the velocity in which it is created at the source. Organizations need to work with the real-time data to achieve a better customer experience and to improve customer engagement.

Streaming data includes a wide variety of data, such as log files generated by customers using your mobile or web applications, ecommerce purchases, in-game player activity, information from social networks, financial trading floors, geospatial services, and telemetry from connected devices or instrumentation in data centers.

For example, sensors in transportation vehicles, industrial equipment, and farm machinery send data to a streaming application. The application monitors performance, detects any potential defects in advance, and places a spare part order automatically preventing equipment down time.

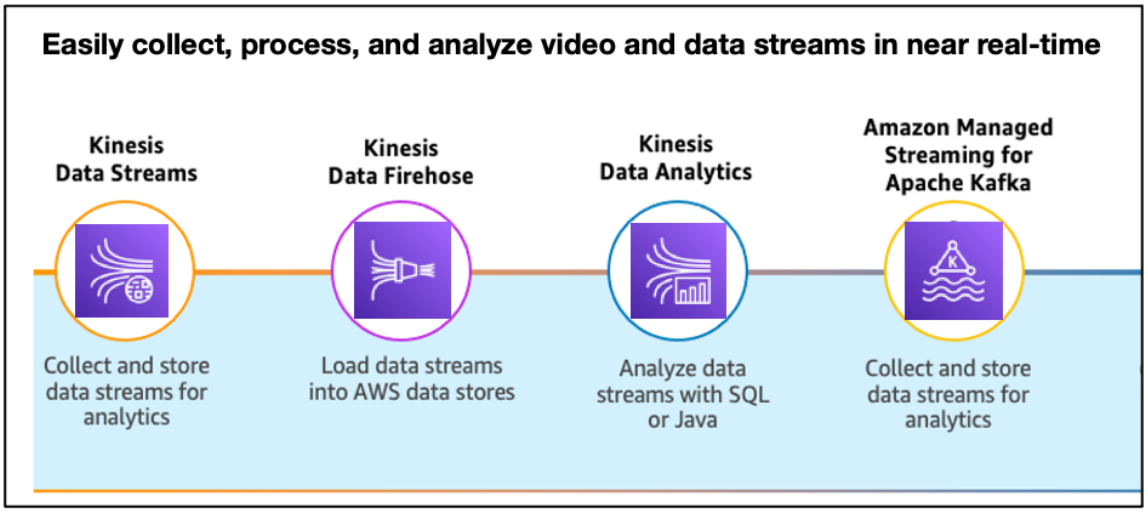
AWS provides a number of options to work with streaming data. You can take advantage of the managed streaming data services offered by Amazon Kinesis, Amazon MSK, Amazon EMR Spark streaming, or deploy and manage your own streaming data solution in the cloud on Amazon Elastic Compute Cloud (Amazon EC2).

Amazon Kinesis is a platform for streaming data on AWS, offering powerful services to make it easy to load and analyze streaming data. It also enables you to build custom streaming data applications for specialized needs. If you have a streaming use case and you want to use an AWS native, fully managed service, consider Amazon Kinesis. It offers four services: Amazon Kinesis Data Streams, Amazon Kinesis Data Firehose, Amazon Kinesis Data Analytics, and Amazon Kinesis Video Streams.

Apache Kafka has been around for ten+ years and good number of customers have been using Kafka for a while. To enhance and reduce the overhead of managing services, AWS has introduced Amazon MSK. In addition, you can run other streaming data platforms such as Apache Flume, Apache Spark Streaming,

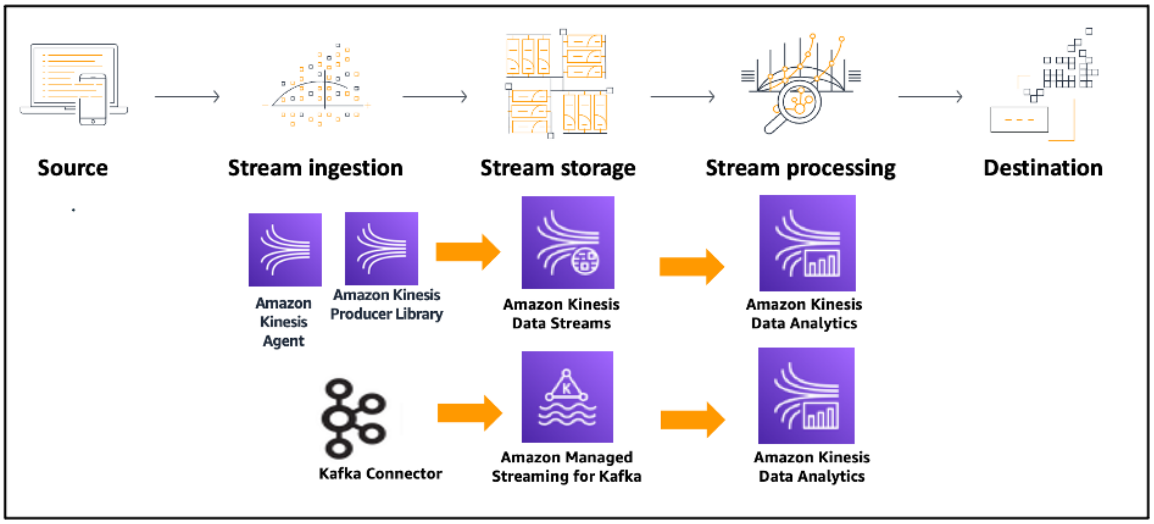
and Apache Storm on Amazon EC2 and Amazon EMR. If open-source technology is critical for your data processing strategy, you are familiar with Apache Kafka, you intend to have multiple consumers per stream, and you are looking for real-time latency less than 200ms, AWS recommends you to choose Amazon MSK rather than Amazon Kinesis.

The following diagram illustrates the various streaming services available on AWS.



Near real-time streaming on AWS

Refer to the following diagram for an example of streaming data lifecycle.



Streaming data lifecycle

Amazon Kinesis Data Streams

Amazon Kinesis Data Streams enables you to build your own custom applications that process or analyze streaming data for specialized needs. It can continuously capture and store terabytes of data per hour from hundreds of thousands of sources. You can then build applications that consume the data from Kinesis Data Streams to power near real-time dashboards, generate alerts, implement dynamic pricing

and advertising, and more. Kinesis Data Streams supports your choice of stream processing framework including Kinesis Client Library (KCL), Apache Storm, and Apache Spark Streaming.

With Kinesis Data Streams, you can ingest real-time data such as application logs, website clickstreams, and Internet of Things (IoT) telemetry data for ML, analytics, and other applications. In addition to streaming ingestion use cases, you can also use Kinesis Data Streams as part of a modern data architecture to facilitate the low latency movement of data.

Kinesis Streams services offer integration with modern data architecture in following ways to unlock new value from your data, such as improving operational efficiency, optimizing processes, developing new products and revenue streams, and building better customer user experiences.

- Use AWS Database Migration Service (AWS DMS) to capture near real-time transactions from relational databases and push data to an Amazon Kinesis data stream as the outside-in data movement approach.
- Capture real-time events from Amazon DynamoDB Streams in a near real-time stream using and AWS Lambda function to Amazon Kinesis Data Firehose as the inside-out data movement approach.
- Create AWS Glue spark streaming extract, transform, load (ETL) jobs that run nearly continuously and consume data from Amazon Kinesis Data Streams. This job cleans and transforms the data, then loads the results into Amazon S3 data lakes or Java Database Connectivity (JDBC) data stores as outside-in data movement approach.

Amazon Kinesis Data Firehose

Amazon Kinesis Data Firehose is the easiest way to load streaming data into AWS. It can capture, transform, and deliver streaming data to Amazon S3, Amazon Redshift, OpenSearch Service, generic HTTP endpoints, and service providers such as Datadog, New Relic, MongoDB, and Splunk.

Kinesis Data Firehose is a fully managed service that automatically scales to match the throughput of your data and requires no ongoing administration. It can also batch, compress, transform, and encrypt your data streams before loading, minimizing the amount of storage used and increasing security.

Following are few use cases that our customers tackle using Amazon Kinesis Data Firehose.

- With Kinesis Data Firehose, you can capture data continuously from connected devices such as consumer appliances, embedded sensors, and TV set-top boxes. Kinesis Data Firehose loads the data into your specified destinations, enabling near real-time access to metrics, insights, and dashboards.
- You can also detect application errors as they happen and identify root cause by collecting, monitoring, and analyzing log data. You can easily install and configure the Amazon Kinesis Agent on your servers to automatically watch application and server log files and send the data to Kinesis Data Firehose. Kinesis Data Firehose continuously streams the log data to your destinations so you can visualize and analyze the data.
- You can use Amazon Kinesis Data Firehose to ingest near real-time clickstream data, enabling marketers to connect with their customers in the most effective way. You can stream billions of small messages that are compressed, encrypted, and delivered to your destinations. From there, you can aggregate, filter, and process the data, and refresh content performance dashboards in near real-time.

Kinesis Data Firehose offers integration with modern data architecture in following ways to derive new deeper insights from your data.

- Kinesis Data Firehose can capture, transform, and load streaming data into Amazon S3, enabling near real-time analytics as the outside-in data movement approach.
- You can also connect Kinesis Data Firehose with Kinesis Data Streams to automatically convert the incoming data to open and standard-based formats like Apache Parquet and Apache ORC before the data is delivered as the inside-out data movement approach.

Amazon Kinesis Data Analytics

Amazon Kinesis Data Analytics is the easiest way to transform and analyze streaming data in real-time with SQL or Apache Flink. Apache Flink is an open-source framework and engine for processing data streams. Kinesis Data Analytics reduces the complexity of building, managing, and integrating Apache Flink applications with other AWS services.

Kinesis Data Analytics takes care of everything required to run streaming applications nearly continuously, and scales automatically to match the volume and throughput of your incoming data. With Kinesis Data Analytics, there are no servers to manage, no minimum fee or setup cost, and you only pay for the resources your streaming applications consume.

Kinesis Data Analytics has the following integration with other AWS services for seamless data movement.

- You can develop streaming ETL applications with Kinesis Data Analytics built-in operators to transform, aggregate, and filter streaming data. You can easily deliver your data in seconds to Amazon Kinesis Data Streams, Amazon MSK, Amazon OpenSearch Service, Amazon S3, custom integrations, and more using built-in connectors.
- You can interactively query and analyze data streams in real time and continuously produce insights and results for time sensitive use cases such as click stream analytics.
- You can develop applications that process events from one or more data streams and trigger conditional processing and external actions. You can identify patterns like anomaly detection in your data streams using standard SQL and Apache Flink libraries for complex event processing and then, store proceed event into data lake for offline analysis.

Amazon Managed Streaming for Apache Kafka (Amazon MSK)

Amazon MSK is a fully managed service that makes it easy for you to build and run applications that use Apache Kafka to process streaming data. Apache Kafka is an open-source platform for building real-time streaming data pipelines and applications. With Amazon MSK, you can use native Apache Kafka APIs to populate data lakes, stream changes to and from databases, and power ML and analytics applications. Amazon MSK uses AWS Glue Schema Registry for validating and controlling the evolution of schemas used by Apache Kafka applications.

Apache Kafka clusters are challenging to set up, scale, and manage in production. When you run Apache Kafka on your own, you need to provision servers, configure Apache Kafka manually, replace servers when they fail, orchestrate server patches and upgrades, architect the cluster for high availability, ensure data is durably stored and secured, setup monitoring and alarms, and carefully plan scaling events to support load changes. Amazon MSK makes it easy for you to build and run production applications on Apache Kafka without needing Apache Kafka infrastructure management expertise. That means you spend less time managing infrastructure and more time building applications.

Amazon MSK has following integration with AWS modern data architecture for seamless data movement, to build purpose-built analytics from your data:

- Amazon MSK integrates AWS IoT for IoT event sourcing using IoT rule action to deliver messages from your devices directly to your Amazon MSK for data analysis and visualization, without writing a single line of code as the outside-in data movement approach.
- AWS DMS can capture data from online transaction processing (OLTP) database systems, and push data to Amazon MSK as producer using the outside-in data movement approach.

- Amazon MSK integrates with Amazon Kinesis Data Analytics Apache Flink applications and the EMR spark streaming application to process streaming data in near real-time using the inside-out data movement approach.

Amazon MSK can be integrated with Kinesis Data Firehose using a Lambda function that processes process records in a Kafka topic. And deliver it to Kinesis Data Firehose delivery stream that buffers data before delivering it to the destination such an Amazon S3 bucket that stores all events from the MSK cluster for offline analysis.

Streaming analytics architecture patterns using a modern data architecture

Organizations perform streaming analytics to build better customer experiences in near real-time to stay ahead of their competitors as the value of data diminishes over time. To be near real time, data needs to be produced, captured, and processed with low latency. Organizations need a system that scales to support the modern data architecture needs of their business, but also allows them to build their own applications on top of the data collected. The order is critical, because applications need to be able to tell the story of what happened, when it happened, and how it happened, relative to other events in the pipeline.

The modern data architecture on AWS provides a strategic vision of how multiple AWS data and analytics services can be combined into a multi-purpose data processing and analytics environment to address these challenges.

Low latency modern data streaming applications

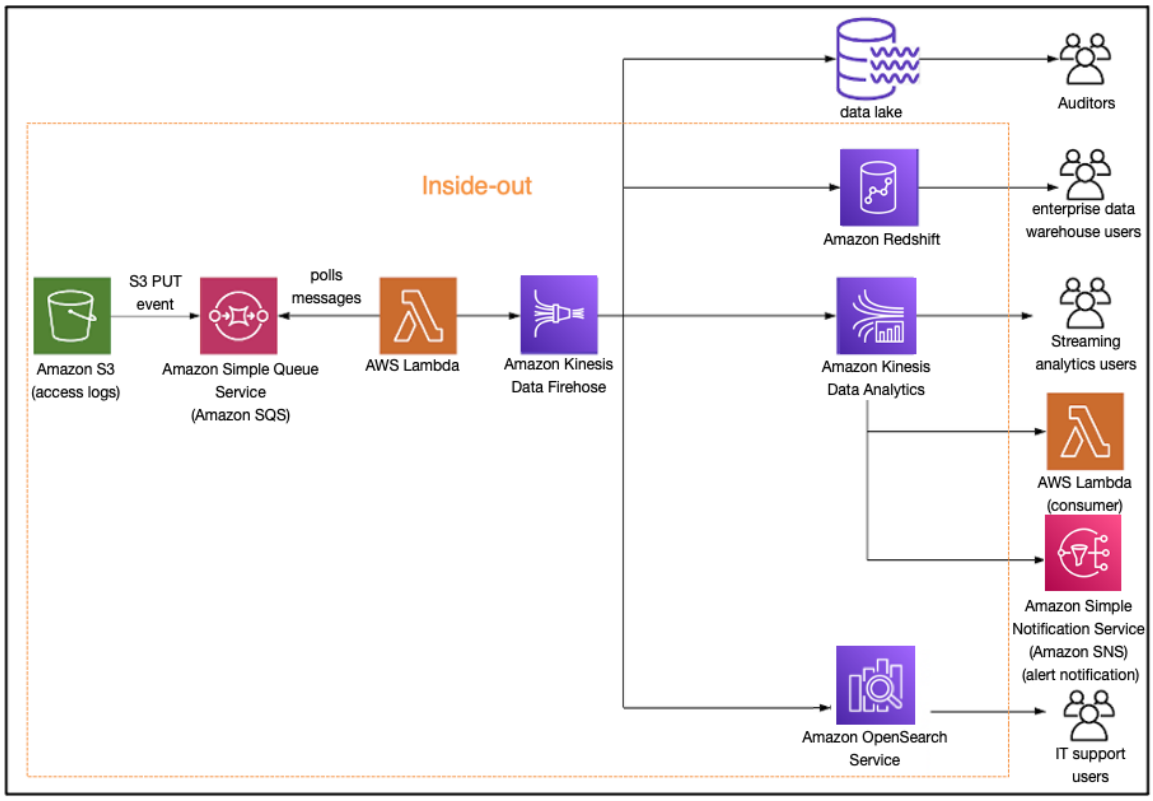
Following are a few use cases for when you need to move data around the purpose-built data services with low latency for faster insights, and how to build these streaming application architectures with AWS streaming technologies.

Build access logs streaming applications using Kinesis Data Firehose and Kinesis Data Analytics

Customers perform log analysis that involves searching, analyzing, and visualizing machine data generated by your IT systems and technology infrastructure. It includes logs and metrics such as user transactions, customer behavior, sensor activity, machine behavior, and security threats. This data is complex, but also the most valuable, as it contains operational intelligence for IT, security, and business.

In this use case, customers have collected log data in an Amazon S3 data lake. They need to access log data and analyze it in a variety of ways, using the right tool for the job for various security and compliance requirements. There are several data consumers including auditors, streaming analytics users, enterprise data warehouse users, and so on. They need to keep a copy of the data for regulatory purposes.

The following diagram illustrates the modern data architecture inside-out data movement with input access logs data, to derive near real-time dashboards and notifications.



Access logs streaming applications for anomaly detection using Amazon Kinesis Data Analytics and Amazon OpenSearch Service

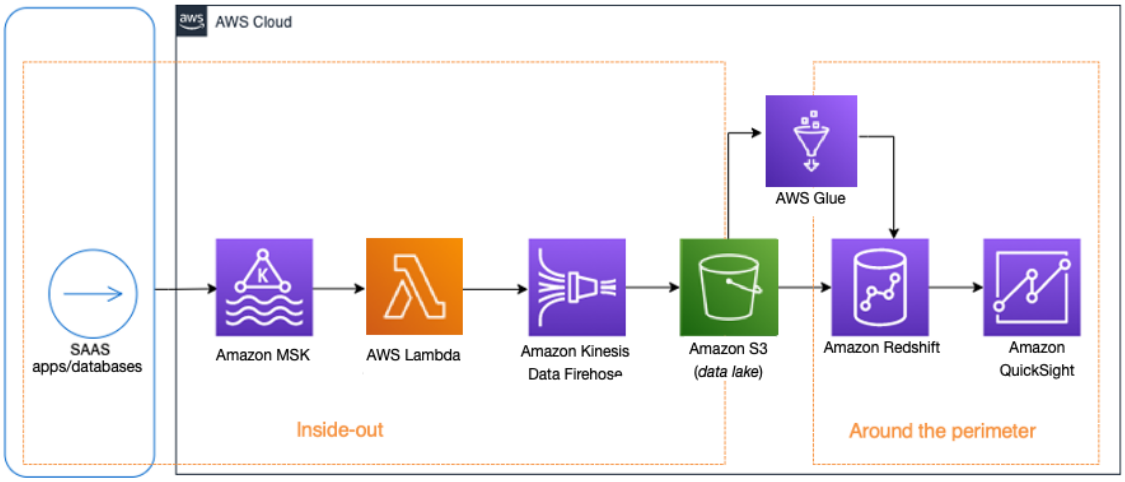
The steps that data follows through the architecture are as follows:

1. Logs from multiple sources such as Amazon CloudFront access logs, VPC Flow Logs, API logs, and application logs are pushed into the data lake.
2. Publish S3 PUT events to Amazon SQS events. AWS Lambda polls the events from SQS and invokes a Lambda function to move data into multiple sources such as Amazon S3, Amazon Redshift, Amazon OpenSearch Service, or Kinesis Data Analytics using Amazon Kinesis Data Firehose.
3. You can build low latency modern data streaming applications by creating a near real-time OpenSearch dashboard, and then processing streaming analytics out with AWS Lambda and Amazon SNS automatic notifications.
4. You can also store access log data into Amazon S3 for archival, and load sub-access log summary data into Amazon Redshift, depending on your use case.

Stream data from diverse source systems into the data lake using MSK for near real-time reports

Customers want to stream near real-time data from diverse source systems such as Software as a Service (SaaS) applications, databases, and social media into S3, and to online analytical processing (OLAP) systems such as Amazon Redshift, to derive user behavior insights and to build better customer experiences. Hopefully these experiences will drive more reactive, intelligent, near real-time experiences. This data in Amazon Redshift can be used to develop customer-centric business reports to improve overall customer experience.

The following diagram illustrates the modern data architecture outside-in data movement, with input stream data to derive near real-time dashboards.



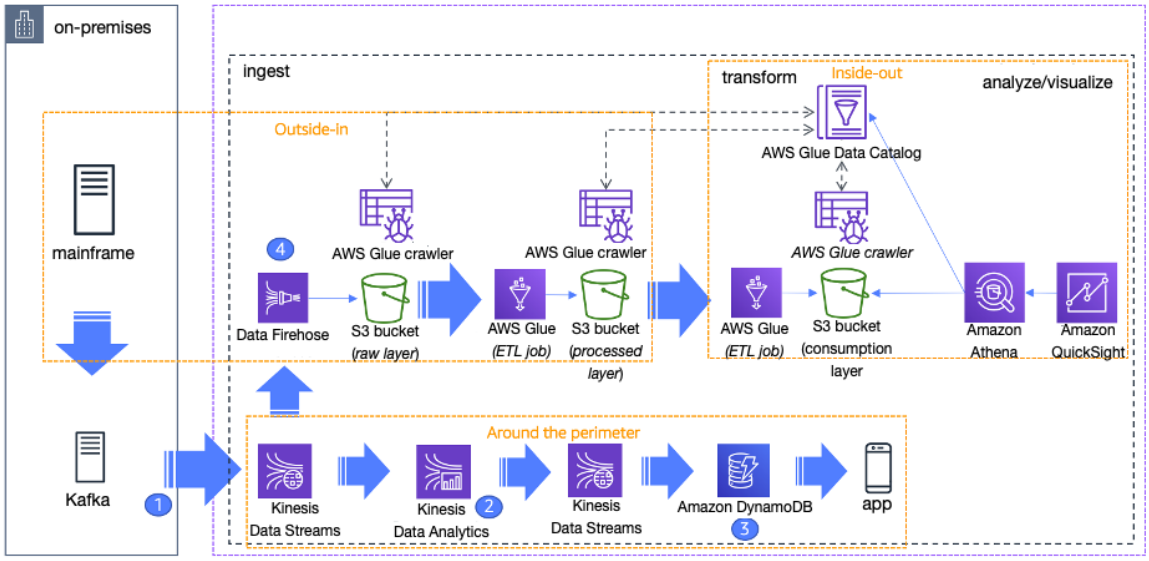
Derive insights from input data coming from diverse source systems for near real-time dashboards with Amazon QuickSight.

The steps that data follows through the architecture are as follows:

1. Stream near real-time data from source systems such as social media using Amazon MSK, Lambda, and Kinesis Data Firehose into Amazon S3.
2. You can use AWS Glue for data processing, and load transformed data into Amazon Redshift using a Glue development endpoint such as Amazon SageMaker notebook instances.
3. Once data is in Amazon Redshift, you can create a customer-centric business report using Amazon QuickSight.

Build a serverless streaming data pipeline using Amazon Kinesis and AWS Glue

Customers want low-latency near real-time analytics to process users' behavior and respond back almost instantaneously with relevant offers and recommendations. The customer's attention will be lost if these recommendations are not available for days, hours, or even minutes – they need to happen in near real-time. The following diagram illustrates a typical modern data architecture for a streaming data pipeline to keep the application up to date, and to store streaming data into a data lake for offline analysis.



Build a serverless streaming data pipeline

The steps that data follows through the architecture are as follows:

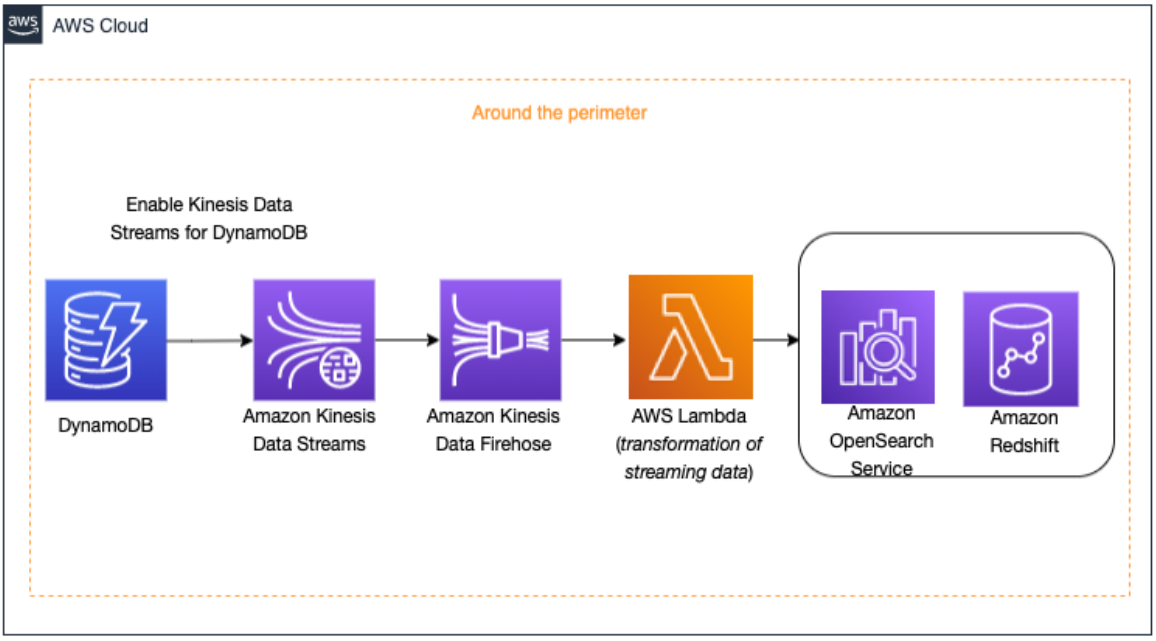
1. Extract data in near real-time from an on-premises legacy system to a streaming platform such as Apache Kafka. From Kafka, you can move the data to Kinesis Data Streams.
2. Use Kinesis Data Analytics to analyze streaming data, gain actionable insights, and respond to your business and customer needs in near real-time.
3. Store analyzed data in cloud scale databases such as Amazon DynamoDB, and push to your end users in near real-time.
4. Kinesis Data Streams can use Kinesis Data Firehose to send the same streaming content to the data lake for non-real-time analytics use cases.

Set up near real-time search on DynamoDB table using Kinesis Data Streams and OpenSearch Service

Organizations want to build a search service for their customers to find the right product, service, document, or answer to their problem as quickly as possible. Their searches will be across both semi-structured and unstructured data, and across different facets and attributes. Search results have to be relevant and delivered in near real-time. For example, if you have an ecommerce platform, you want customers to find the product they are looking for quickly in near real-time.

You can use both DynamoDB and OpenSearch Service for building a near real-time search service. You can use DynamoDB as a durable store, and OpenSearch Service to extend its search capabilities. When you set up your DynamoDB tables and streams to replicate your data into OpenSearch Service, you can perform near real-time, full-text search on your data. You can also load part of the data into Amazon Redshift, depending on your use case.

The following diagram illustrates the modern data architecture around the perimeter data movement with Amazon DynamoDB and Amazon OpenSearch Service.



Derive insights from Amazon DynamoDB data by setting up near real-time search using Amazon OpenSearch Service

The steps that data follows through the architecture are as follows:

1. In this design, the DynamoDB table is used as the primary data store. An Amazon OpenSearch Service cluster is used to serve all types of searches by indexing the table.
2. Using DynamoDB streams with Kinesis Data Streams, any update, deletion, or new item on the main table is captured and processed using AWS Lambda. Lambda makes appropriate calls to OpenSearch Service for indexing the data in near real-time.
3. For more details about this architecture, refer to Indexing Amazon DynamoDB Content with Amazon OpenSearch Service Using AWS Lambda and Loading Streaming Data into Amazon OpenSearch Service from Amazon DynamoDB.
4. You can also use streaming functionality to send the changes to OpenSearch Service or Amazon Redshift via a Data Firehose delivery stream. Before you load data into OpenSearch Service or Amazon Redshift, you might need to perform transforms on the data. You can use Lambda functions to perform this task. For more information, refer to Amazon Kinesis Data Firehose Data Transformation.

Key considerations while building streaming analytics

When you are building a streaming data pipeline using modern data architecture for analytics and ML, you must first understand the ideal usage patterns of AWS streaming data solutions, your user personas, and your specific use case so you can choose the right service for the job.

Choosing the right Kinesis service for your use case

The following table illustrates the ideal usage patterns of various Kinesis data streaming and processing services.

Table 1: Amazon Kinesis usage patterns

	Kinesis Data Streams	**Kinesis Data Firehose**	**Kinesis Data Analytics**
Usage	Collect and store data streams for analytics	Load data streams into AWS data stores	Analyze data streams with Data Analytics Studio or Apache Flink
Data sources	Mobile apps, application logs, web clickstream/social, IoT sensors, connected products, smart buildings	Connected devices such as consumer appliances, embedded sensors, TV set-top boxes, clickstream data, application logs	Analyze streaming data from Kinesis Data Streams, Amazon MSK, Amazon MQ, custom connectors
Stream ingestion	AWS SDKs, Amazon Kinesis Producer Library, AWS Mobile SDKs, Kinesis Agent, AWS IoT, Amazon CloudWatch Events, Amazon DynamoDB, AWS DMS	AWS SDKs, Kinesis Producer Library, Kinesis Data Streams, Kinesis Agent, AWS IoT, Amazon CloudWatch Events	Analyze streaming data from Kinesis Data Streams, Amazon MSK, Amazon MQ, custom connectors

Choosing the right streaming for your use case

The following table Illustrates the comparison between Apache Kafka, Kinesis Data Streams, and Amazon MSK.

Table 2 — Streaming services

Attribute	Apache Kafka	Kinesis Streams	MSK
Ease of use	Advanced setup required	Get started in minutes	Get started in minutes
Management Overhead	High	Low	Medium
Scalability	Difficult to scale	Scale in seconds with one click	Scale in minutes with one click
Throughput	Infinite	Scale with shards, supports up to 1MB payloads	Very large
Infrastructure	You manage	AWS manages	AWS manages
Write-to-read latency	<100 ms is achievable	<100 ms (with HTTP/2)	<100 ms is achievable
Open-sourced?	Yes	No	Yes

Streaming data processing technologies

Streaming data processing technologies support many use cases that include event-driven applications, data analytics applications, and data pipeline applications. Commonly used frameworks include Apache Kafka Streams, Apache Flink, KSQL, and Kinesis Data Analytics for Flink. Apache Kafka Streams, Apache Flink, and KSQL are open-source options, while Amazon Kinesis Data Analytics offers a fully managed Apache Flink and SQL solution.

The following table Illustrates the comparison between Apache Kafka Streams, Kinesis Data Analytics for Apache Flink, and Kinesis Data Analytics SQL.

Table 3 — Comparison between data stream processing technologies

Feature	Apache Kafka Streams	Kinesis Data Analytics for Apache Flink	Kinesis Data Analytics SQL
Open source	Yes	Based on open-source Apache Flink	No, based on proprietary engine
Sources	Kafka only	Kinesis Data Streams, Amazon MSK for Apache Kafka, DynamoDB streams, RabbitMQ	Kinesis Data Streams, Kinesis Data Firehose, S3 reference source
Destination/sinks	Kafka only; >10 connectors supported with Kafka connect	Amazon MSK for Kafka, Kinesis Data Streams, Kinesis Data Firehose, S3, Apache Cassandra, Amazon DynamoDB, OpenSearch Service, custom sinks supported by open-source Flink	Kinesis Data Streams, Kinesis Data Firehose, S3, OpenSearch Service, Amazon Redshift
Development languages	Java and Scala	Java, Scala and Python	SQL only

Feature	Apache Kafka Streams	Kinesis Data Analytics for Apache Flink	Kinesis Data Analytics SQL
Development process	Develop on any integrated development environment (IDE) using Java/Scala. The application is separate from the Kafka broker and needs to be scaled independently.	Develop on any IDE and build a JAR file. Create a Kinesis Data Analytics Flink application and upload application JAR.	Kinesis Data Analytics SQL editor on the AWS Management Console allows customers to develop and run queries as applications.
Exactly once processing support	Yes	Yes	No, support at least once
Per record processing latency	Sub-second	Sub-second	Seconds
Batch support	No	Yes, supported by Flink	No

Key benefits

Modern data streaming analytics architecture on AWS provides the following key benefits:

- A comprehensive set of integrated tools enables every user equally.
- Empowers all personas — use best-fit analytics services.
- Build seamless low latency analytics architectures for near real-time personalization, and tailoring customer experience in near real-time.
- Simplified streaming ingestion and cleaning enables data engineers to build streaming applications faster.
- Unified low latency streaming analytics across operational databases, data warehouse, and data lake.
- Democratizes ML with SQL.
- Security, compliance, and audit capabilities across the data lake.
- Cost-effective, durable storage with global replication capabilities.
- Centralized management of fine-grained permissions empowers security officers.

Conclusion

AWS modern data architecture provides various AWS purpose-built and managed streaming analytics services to satisfy low-latency use cases. Each of these streaming analytics services has its own characteristics in terms of scalability, latency, cost, and supported data sources. The seamless data movement empowered by AWS modern data architecture allows organizations to gain full insights from their data using various data analytics services.

With streaming analytics applications built with AWS modern data architecture, organizations can get low latency insights quickly with near real-time analytics without worrying about the underlining infrastructures. The flexibility and extensibility of AWS modern data architecture makes it easy to support new use cases by using new analytics services as they become available.

Contributors

Contributors to this document include:

- Raghavarao Sodabathina, Enterprise Solutions Architect, Amazon Web Services
- Changbin Gong, Principal Solutions Architect, Amazon Web Services
- Niyati Upadhyay, Solutions Architect, Amazon Web Services
- Soujanya Konka, Solutions Architect, Amazon Web Services

Further reading

For detailed architectural patterns, refer to the following resources:

- Modern data architecture on AWS
- Harness the power of your data with AWS Analytics (blog post)
- *Derive Insights from AWS Modern Data* (AWS whitepaper)
- Design a data mesh architecture using AWS Lake Formation and AWS Glue (blog post)
- *AWS Cloud Data Ingestion Patterns and Practices* (AWS whitepaper)
- Creating a source to Lakehouse data replication pipe using Apache Hudi, AWS Glue, AWS DMS, and Amazon Redshift (blog post)
- Best practices for consuming Amazon Kinesis Data Streams using AWS Lambda (blog post)
- Best practices for Kinesis Data Analytics from the *Amazon Kinesis Data Analytics for SQL Applications Developer Guide*
- Security Best Practices for Kinesis Data Firehose from the *Amazon Kinesis Data Firehose Developer Guide*
- Best practices from Delhivery on migrating from Apache Kafka to Amazon MSK (blog post)

Document revisions

Date	Description
May 17, 2022	First publication

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS glossary

For the latest AWS terminology, see the AWS glossary in the *AWS General Reference*.