The AWS Advantage in AWS Big Data

This whitepaper helps architects, data scientists, and developers understand the big data analytics options available in the Amazon Web Services (AWS) Cloud.

Topics:

Amazon Kinesis

Amazon Kinesis is a platform for streaming data on AWS that makes it easy to load and analyze streaming data. Amazon Kinesis also enables you to build custom streaming data applications for specialized needs. With Kinesis, you can ingest real-time data such as application logs, website clickstreams, Internet of Things (IoT) telemetry data, and more into your databases, data lakes, and data warehouses, or build your own real-time applications using this data. Amazon Kinesis enables you to process and analyze data as it arrives and respond in real-time instead of having to wait until all your data is collected before the processing can begin.

Currently there are four pieces of the Kinesis platform that can be utilized based on your use case:

- Amazon Kinesis Data Streams enables you to build custom applications that process or analyze streaming data.
- Amazon Kinesis Video Streams enables you to build custom applications that process or analyze streaming video.
- Amazon Data Firehose enables you to deliver real-time streaming data to AWS destinations such as Amazon S3, Amazon Redshift, OpenSearch Service, and Splunk.
- Amazon Managed Service for Apache Flink enables you to process and analyze streaming data with standard SQL or with Java (managed Apache Flink).

Kinesis Data Streams and Kinesis Video Streams enable you to build custom applications that process or analyze streaming data in real time. Kinesis Data Streams can continuously capture and store terabytes of data per hour from hundreds of thousands of sources, such as website clickstreams, financial transactions, social media feeds, IT logs, and location-tracking events.

Kinesis Video Streams can continuously capture video data from smartphones, security cameras, drones, satellites, dashcams, and other edge devices.

With the Amazon Kinesis Client Library (KCL), you can build Amazon Kinesis applications and use streaming data to power real-time dashboards, generate alerts, and implement dynamic pricing and advertising. You can also emit data from Kinesis Data Streams and Kinesis Video Streams to other AWS services such as Amazon S3, Amazon Redshift, Amazon EMR, and AWS Lambda.

Provision the level of input and output required for your data stream, in blocks of one megabyte per second (MB/sec), using the AWS Management Console, API, or SDKs. The size of your stream can be adjusted up or down at any time without restarting the stream and without any impact on the data sources pushing data to the stream. Within seconds, data put into a stream is available for analysis.

With Amazon Data Firehose, you don't need to write applications or manage resources. You configure your data producers to send data to Firehose and it automatically delivers the data to the AWS destination or third party (Splunk) that you specified. You can also configure Firehose to transform your data before data delivery. It 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, and encrypt the data before loading it, minimizing the amount of storage used at the destination and increasing security.

Amazon Managed Service for Apache Flink is the easiest way to process and analyze real-time, streaming data. With Managed Service for Apache Flink, you just use standard SQL or Java (Flink) to process your data streams, so you don't have to learn any new programming languages. Simply point Managed Service for Apache Flink at an incoming data stream, write your SQL queries, and specify where you want to load the results. Managed Service for Apache Flink takes care of running your SQL queries continuously on data while it's in transit and sending the results to the destinations.

For complex data processing applications, Amazon Managed Service for Apache Flink provides an option to use open-source libraries such as Apache Flink, Apache Beam, AWS SDK, and AWS service integrations. It includes more than ten connectors from Apache Flink and gives you the ability to build custom integrations. It's also compatible with the AWS Glue Schema Registry, a serverless feature of AWS Glue that enables you to validate and control the evolution of streaming data using registered Apache Avro schemas.

You can use Apache Flink in Amazon Managed Service for Apache Flink to build applications whose processed records affect the results exactly once, referred to as exactly once processing. This means that even in the case of an application disruption, like internal service maintenance or user-initiated application update, the service will ensure that all data is processed and there is no duplicate data. The service stores previous and in-progress computations, or *state*, in running application storage. This enables you to compare real-time and past results over any time period and provides fast recovery during application disruptions.

The subsequent sections focus primarily on Amazon Kinesis Data Streams.

Ideal usage patterns

Amazon Kinesis Data Steams is useful wherever there is a need to move data rapidly off producers (data sources) and continuously process it. That processing can be to transform the data before emitting it into another data store, drive real-time metrics and analytics, or derive and aggregate multiple streams into more complex streams, or downstream processing. The following are typical scenarios for using Kinesis Data Streams for analytics:

- **Real-time data analytics** – Kinesis Data Streams enables real-time data analytics on streaming data, such as analyzing website clickstream data and customer engagement analytics.
- **Log and data feed intake and processing** – With Kinesis Data Streams, you can have producers push data directly into an Amazon Kinesis stream. For example, you can submit system and application logs to Kinesis Data Streams and access the stream for processing within seconds. This prevents the log data from being lost if the front-end or application server fails an d reduces local log storage on the source. Kinesis Data Streams provides accelerated data intake because you are not batching up the data on the servers before you submit it for intake.
- **Real-time metrics and reporting** – You can use data ingested into Kinesis Data Streams for extracting metrics and generating KPIs to power reports and dashboards at real-time speeds. This enables data-processing application logic to work on data as it is streaming in continuously, rather than waiting for data batches to arrive.

Anti-patterns

Amazon Kinesis Data Streams has the following anti-patterns:

- **Small scale consistent throughput** – Even though Kinesis Data Streams works for streaming data at 200 KB per second or less, it is designed and optimized for larger data throughputs.
- **Long-term data storage and analytics** – Kinesis Data Streams is not suited for long-term data storage. By default, data is retained for 24 hours, and you can extend the retention period by up to 365 days.

AWS Lambda

AWS Lambda enables you to run code without provisioning or managing servers. You pay only for the compute time you consume – there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service – all with zero administration. Just upload your code and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.

Ideal usage patterns

AWS Lambda enables you to run code in response to triggers such as changes in data, shifts in system state, or actions by users. Lambda can be directly triggered by AWS services such as Amazon S3, DynamoDB, Amazon Kinesis Data Streams, Amazon Simple Notification Service (Amazon SNS), and Amazon CloudWatch, enabling you to build a variety of real-time data processing systems:

- **Real-time file processing –** You can trigger Lambda to invoke a process where a file has been uploaded to Amazon S3 or modified. For example, to change an image from color to gray scale after it has been uploaded to Amazon S3.
- **Real-time stream processing** – You can use Kinesis Data Streams and Lambda to process streaming data for click stream analysis, log filtering, and social media analysis.
- **Extract, transform, load (ETL)** – You can use Lambda to run code that transforms data and loads that data into one data repository to another.
- **Replace Cron** – Use schedule expressions to run a Lambda function at regular intervals as a cheaper and more available solution than running cron on an EC2 instance.

- **Process AWS Events** – Many other services, such as AWS CloudTrail, can act as event sources simply by logging to Amazon S3 and using S3 bucket notifications to trigger Lambda functions.

Anti-patterns

AWS Lambda has the following anti-patterns:

- **Long running application** — Each Lambda function must complete within 900 seconds. For long running applications that may require jobs to run longer than fifteen minutes, Amazon EC2, Amazon EKS, or Amazon ECS is recommended. Alternately, you can use Step Functions.
- **Dynamic websites** — While it is possible to run a static website with AWS Lambda, running a highly dynamic and large volume website can be performance prohibitive. Utilizing Amazon EC2, Amazon EKS, or Amazon ECS and AWS CloudFormation is a recommended use-case.
- **Stateful applications** — Lambda code must be written in a “stateless” style, meaning it should assume there is no affinity to the underlying compute infrastructure. Local file system access, child processes, and similar artifacts may not extend beyond the lifetime of the request, and any persistent state should be stored in Amazon S3, DynamoDB, or another internet-available storage service.

Amazon EMR

Amazon EMR is the industry-leading 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. Amazon EMR makes it easy to set up, operate, and scale your big data environments by automating time-consuming tasks like provisioning capacity and tuning clusters. With EMR you can run petabyte-scale analysis at less than half of the cost of traditional on-premises solutions and over 3x faster than standard Apache Spark. You can run workloads on Amazon EC2 instances, on Amazon EKS clusters, or on-premises using EMR on AWS Outposts.

Amazon EMR does all the work involved with provisioning, managing, and maintaining the infrastructure and software of a Hadoop cluster. Amazon EMR now supports Amazon EC2 M6g instances to deliver the best price performance for cloud workloads. Amazon EC2 M6g instances are powered by AWS Graviton2 processors that are custom designed by AWS, utilizing 64-bit Arm Neoverse cores. Amazon EMR provides up to 35% lower cost and up to 15% improved performance for Spark workloads on Graviton2-based instances versus previous generation instances.

Ideal usage patterns

Amazon EMR’s flexible framework reduces large processing problems and data sets into smaller jobs and distributes them across many compute nodes in a Hadoop cluster. This capability lends itself to many usage patterns with big data analytics. Here are a few examples:

- Log processing and analytics
- Large ETL data movement
- Risk modeling and threat analytics
- Ad targeting and click stream analytics
- Genomics
- Predictive analytics
- Ad hoc data mining and analytics

For more information, see the documentation for Amazon EMR.

Anti-patterns

Amazon EMR has the following anti-pattern:

- **Small datasets** – Amazon EMR is built for massive parallel processing; if your data set is small enough to run quickly on a single machine, in a single thread, the added overhead to map and reduce jobs may not be worth it for small datasets that can easily be processed in memory on a single system. Amazon EMR or a relational database running on Amazon EMR may be a better option for workloads with stringent requirements.

AWS Glue

AWS Glue is a serverless data integration service that makes it easy to discover, prepare, and combine data for analytics, machine learning, and application development. AWS Glue provides all of the capabilities needed for data integration. It uses both visual and code-based interfaces to make data integration easier.

Users can easily find and access data using the AWS Glue Data Catalog. Data engineers and ETL developers can visually create, run, and monitor ETL workflows with a few clicks in AWS Glue

Studio. Data analysts and data scientists can use AWS Glue DataBrew to visually enrich, clean, and normalize data without writing code.

Ideal usage patterns

AWS Glue is designed to easily prepare data for extract, transform, and load (ETL) jobs. Using AWS Glue gives you the following benefits:

- **Data discovery**
    - Automatic schema discovery, using AWS Glue crawlers.
    - Manage and enforce schemas for data streams with AWS Glue Schema Registry.
- **Data Catalog**
    - The AWS Glue Data Catalog is a central repository to store structural and operational metadata for all your data assets. For a given dataset, you can store its table definition, physical location, add business relevant attributes, as well as track how this data has changed over time.
    - The AWS Glue Data Catalog is Apache Hive Metastore-compatible and is a drop-in replacement for the Apache Hive Metastore for Big Data applications running on Amazon EMR, and third-party applications such as Databricks.
- **Data transformation**
    - Visually transform data with a drag and drop interface using AWS Glue Studio. AWS Glue automatically generates reusable and portable code using familiar technology – Python (or Scala) and Spark.
    - Serverless streaming ETL jobs in AWS Glue continuously consume data from streaming sources including Amazon Kinesis and Amazon MSK, clean and transform it in-flight, and make it available for analysis in seconds in your target data store.
- **Data replication**
    - AWS Glue Elastic Views enables you to create views over data stored in multiple types of AWS data stores and materialize the views in a target data store of your choice by writing queries in PartiQL, an open-source SQL-compatible query language.
- **Data preparation**
    - Deduplicate and cleanse data with built-in machine learning. The FindMatches feature deduplicates and finds records that are imperfect matches of each other.
    - Normalize data without code using a visual interface. AWS Glue DataBrew provides an interactive, point-and-click visual interface for users like data analysts and data

    scientists to clean and normalize data without writing code. Choose from over 250 built-in transformations.

- **Integration**
    - Integration with data access services like Amazon Athena, Amazon EMR, and Amazon Redshift. Also, with third parties.
- **Serverless**
    - No infrastructure to provision or manage.

Anti-patterns

AWS Glue has the following anti-patterns:

- **Multiple ETL engines** – AWS Glue ETL jobs are Spark-based. If your use case requires you to use an engine other than Apache Spark or if you want to run a heterogeneous set of jobs that run on a variety of engines like Hive or Pig, Amazon EMR is a better choice.
- **Configurable Spark environment** – AWS Glue is a fully managed service. While you can change some configuration parameters in your cluster, if your use case requires extensive configuration changes, Amazon EMR or Amazon EMR on EKS is a better choice.

AWS Lake Formation

AWS Lake Formation is an integrated data lake service that makes it easy for you to ingest, clean, catalog, transform, and secure your data and make it available for analysis and machine learning. 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.

Lake Formation automatically manages access to the registered data in S3 via services including AWS Glue, Amazon Athena, Amazon Redshift, Amazon EMR Notebooks and Zeppelin notebooks with Apache Spark, and Amazon QuickSight to ensure compliance with your defined policies. If you’ve set up transformation jobs spanning AWS services, Lake Formation configures the flows, centralizes their orchestration, and lets you monitor the running of your jobs. With Lake Formation, you can configure and manage your data lake without manually integrating multiple underlying AWS services.

Ideal usage patterns

AWS Lake Formation helps you collect and catalog data from databases and object storage, move the data into your new S3 data lake, clean and classify your data using machine learning algorithms, and secure access to your sensitive data. Your users can access a centralized data catalog which describes available datasets and their appropriate usage. Using AWS Lake Formation gives you the following benefits:

- **Build data lakes quickly**
  - Lake Formation blueprints simplify the deployment of ingestion workflow via a simple interface.
  - Simply point Lake Formation at your data sources, and Lake Formation crawls those sources and moves the data into your new S3 data lake.
  - Lake Formation has built-in machine learning to deduplicate and find matching records (two entries that refer to the same thing) to increase data quality.
- **Simplify security management**
  - Centrally define security, governance, and auditing policies in one place.
  - You can define security policy-based rules for your users and applications by role in Lake Formation, and integration with AWS Identity and Access Management (AWS IAM) authenticates those users and roles.
  - With tag-based access control, data stewards can define a tag ontology based on data classification, and grant access based on tags to IAM principals and SAML principals or groups.
  - Enforce encryption leveraging the encryption capabilities of S3 for data in your data lake.
  - Provides comprehensive audit logs with CloudTrail to monitor access and show compliance with centrally defined policies.
- **Provide self-service access to data**
  - Label your data with business metadata. Designate data owners, such as data stewards and business units, by adding a field in table properties as “custom attributes”.
  - Specify, grant, and revoke permissions on tables defined in the central Data Catalog. The same Data Catalog is available for multiple accounts, groups, and services.
  - Search for relevant data by name, contents, sensitivity, or other any other custom labels you have defined.
- **Integration**
  - Integration with data access services like Amazon Athena, Amazon EMR, Amazon Redshift, and Amazon QuickSight.

- **Serverless**
  - No infrastructure to provision or manage.

Anti-patterns

AWS Lake Formation has the following anti-patterns:

- **Managing unstructured data** – AWS Lake Formation does not manage security for unstructured objects as documents (PDF, Word), images, or videos.
- **Business catalog** – Lake Formation helps label your data with business metadata and designate data owners such as data stewards and business units by adding a field in table properties as "custom attributes". For advanced use cases, consider integrating a Partner solution from the AWS Marketplace.
- **Managing permission for open source and third-party big data components** — AWS Lake Formation is not integrated with Presto, HBase and other EMR components not specified in the previous **Interface** section of this chapter. It is also not integrated with third party applications such as Trino or Databricks.
- **Real-time analytics** – AWS Lake Formation is not integrated with Amazon Kinesis or OpenSearch Service. If you need to manage a real-time analytics pipeline or dashboard, you will have to manage privileges in a traditional way.

Credit to: AWS Documentation