AWS Whitepaper

Storage Best Practices for Data and Analytics Applications

Storage Best Practices for Data and Analytics Applications: AWS Whitepaper

Table of Contents

This whitepaper is for historical reference only. Some content might be outdated and some links might not be available.

Storage Best Practices for Data and Analytics Applications

Publication date: **November 16, 2021** (*Document history*)

Amazon Simple Storage Service (Amazon S3) and Amazon Glacier (Amazon Glacier) provide ideal storage solutions for data lakes. Data lakes, powered by Amazon S3, provide you with unmatched availability, agility, and flexibility required to combine different types of data and analytics approaches to gain deeper insights, in ways that traditional data silos and data warehouses cannot. In addition, data lakes built on Amazon S3 integrate with other analytical services for ingestion, inventory, transformation, and security of your data in the data lake. This guide explains each of these options and provides best practices for building, securing, managing, and scaling a data lake built on Amazon S3.

Introduction

Because organizations are collecting and analyzing increasing amounts of data, traditional on-premises solutions for data storage, data management, and analytics can no longer keep pace. Data silos that aren't built to work well together make it difficult to consolidate the storage for more comprehensive and efficient analytics. This, in turn, limits an organization's agility and ability to derive more insights and value from its data. In addition, this reduces the capability to seamlessly adopt more sophisticated analytics tools and processes, because it necessitates upskilling of the workforce.

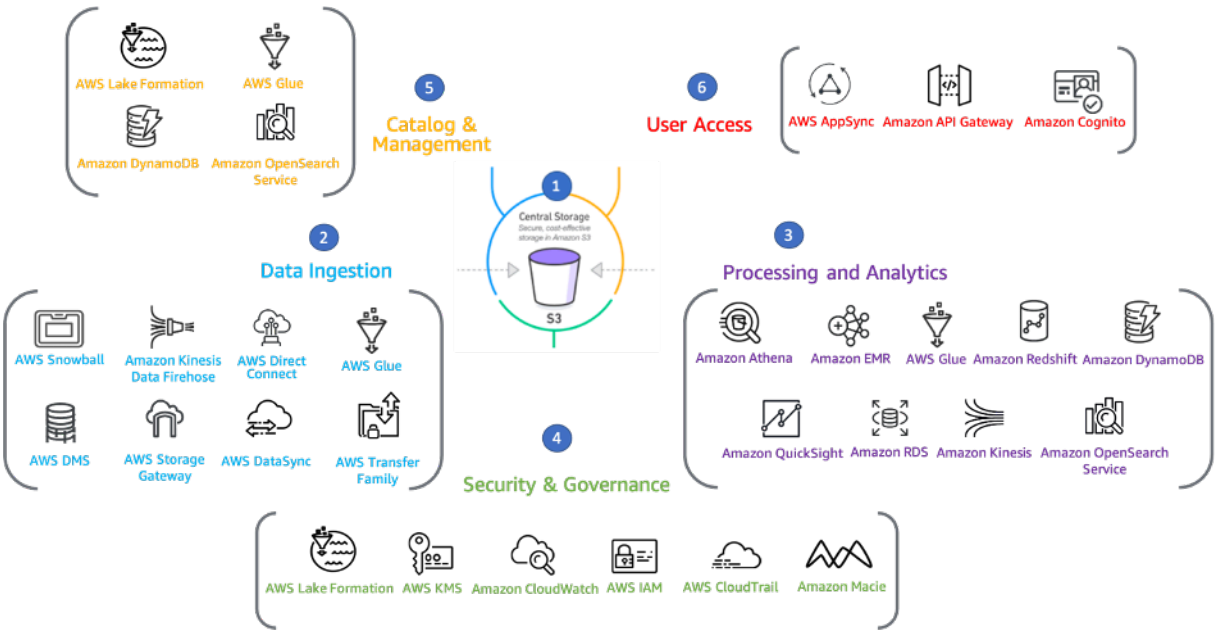
A *data lake* is an architectural approach that allows you to store all your data in a centralized repository, so that it can be categorized, catalogued, secured, and analyzed by a diverse set of users and tools. In a data lake you can ingest and store structured, semi-structured, and unstructured data, and transform these raw data assets as needed. Using a cloud-based data lake you can easily decouple the compute from storage, and scale each component independently, which is a huge advantage over an on-premises or Hadoop-based data lake. You can use a complete portfolio of data exploration, analytics, machine learning, reporting, and visualization tools on the data. A data lake makes data and the optimal analytics tools available to more users, across more lines of business, enabling them to get all of the business insights they need, whenever they need them.

More organizations are building data lakes for various use cases. To guide customers in their journey, Amazon Web Services (AWS) has developed a data lake architecture that allows you to build scalable, secure data lake solutions cost-effectively using Amazon S3 and other AWS services.

Using the data lake built on Amazon S3 architecture capabilities you can:

- Ingest and store data from a wide variety of sources into a centralized platform.
- Build a comprehensive data catalog to find and use data assets stored in the data lake.
- Secure, protect, and manage all of the data stored in the data lake.
- Use tools and policies to monitor, analyze, and optimize infrastructure and data.
- Transform raw data assets in place into optimized usable formats.
- Query data assets in place.
- Integrate the unstructured data assets from Amazon S3 with structured data assets in a data warehouse solution to gather valuable business insights.
- Store the data assets into separate buckets as the data goes through extraction, transformation, and load process.
- Use a broad and deep portfolio of data analytics, data science, machine learning, and visualization tools.
- Quickly integrate current and future third-party data-processing tools.
- Securely share processed datasets and results.
- Scale virtually to unlimited capacity.

The remainder of this paper provides more information about each of these capabilities. The following figure illustrates a sample AWS data lake platform.



High-level AWS data lake technical reference architecture

Central storage: Amazon S3 as the data lake storage platform

A data lake built on AWS uses Amazon S3 as its primary storage platform. Amazon S3 provides an optimal foundation for a data lake because of its virtually unlimited scalability and high durability. You can seamlessly and non-disruptively increase storage from gigabytes to petabytes of content, paying only for what you use. Amazon S3 is designed to provide 99.999999999% durability. It has scalable performance, ease-of-use features, native encryption, and access control capabilities. Amazon S3 integrates with a broad portfolio of AWS and third-party ISV tools for data ingestion, data processing, and data security.

Key data lake-enabling features of Amazon S3 include the following:

- **Decoupling of storage from compute and data processing** – In traditional Hadoop and data warehouse solutions, storage and compute are tightly coupled, making it difficult to optimize costs and data processing workflows. With Amazon S3, you can cost-effectively store all data types in their native formats. You can then launch as many or as few virtual servers as you need using Amazon Elastic Compute Cloud (Amazon EC2) to run analytical tools, and use services in AWS analytics portfolio, such as Amazon Athena, AWS Lambda, Amazon EMR, and Quick, to process your data.
- **Centralized data architecture** – Amazon S3 makes it easy to build a multi-tenant environment, where multiple users can run different analytical tools against the same copy of the data. This improves both cost and data governance over that of traditional solutions, which require multiple copies of data to be distributed across multiple processing platforms.
- **S3 Cross-Region Replication:** – You can use Cross-Region Replication to copy your objects across S3 buckets within the same account or even with a different account. Cross-Region Replication is particularly useful in meeting compliance requirements, minimizing latency by storing the objects closer to the user location, and improving operational efficiency.
- **Integration with clusterless and serverless AWS services** – You can use Amazon S3 with Athena, Amazon Redshift Spectrum, and AWS Glue to query and process data. Amazon S3 also integrates with AWS Lambda serverless computing to run code without provisioning or managing servers. You can process event notifications from S3 through AWS Lambda, such as when an object is created or deleted from a bucket. With all of these capabilities, you only pay for the actual amounts of data you process or for the compute time that you consume. For machine learning use cases, you need to store the model training data and the model artifacts generated during

model training. Amazon SageMaker AI integrates seamlessly with Amazon S3, so you can store the model training data and model artifacts on a single or different S3 bucket.

- **Standardized APIs** – Amazon S3 RESTful application programming interfaces (APIs) are simple, easy to use, and supported by most major third-party independent software vendors (ISVs), including leading Apache Hadoop and analytics tool vendors. This allows customers to bring the tools they are most comfortable with and knowledgeable about to help them perform analytics on data in Amazon S3.
- **Secure by default** – Amazon S3 is secure by default. Amazon S3 supports user authentication to control access to data. It provides access control mechanisms such as bucket policies and access-control lists to provide fine-grained access to data stored in S3 buckets to specific users and groups of users. You can also manage the access to shared data within Amazon S3 using S3 Access Points. More details about S3 Access Points are included in the **Securing, protecting, and managing data** section. You can also securely access data stored in S3 through SSL endpoints using HTTPS protocol. An additional layer of security can be implemented by encrypting the data-in-transit and data-at-rest using server-side encryption (SSE).
- **Amazon S3 for storage of raw and iterative data sets** – When working with a data lake, the data undergoes various transformations. With extract, transform, load (ETL) processes and analytical operations, various versions of the same data sets are created or required for advanced processing. You can create different layers for storing the data based on the stage of the pipeline, such as raw, transformed, curated, and logs. Within these layers you can also create additional tiers based on the sensitivity of the data.
- **Storage classes for cost savings, durability, and availability** – Amazon S3 provides a range of storage classes for various use cases.
  - **S3 Standard** – General purpose storage for frequently accessed data.
  - **S3 Standard Infrequent Access (S3 Standard-IA)** and **S3 One Zone Infrequent Access (S3 One Zone – IA)** – Infrequently accessed, long lived data.
  - **Amazon Glacier and S3 Glacier Deep Archive** – Long-term archival of data.

    Using S3 Lifecycle policy, you can move the data across different storage classes for compliance and cost optimization.
- **Scalability and support for structured, semi-structured, and unstructured data** – Amazon S3 is a petabyte scale object store which provides virtually unlimited scalability to store any type of data. You can store structured data (such as relational data), semi-structured data (such as JSON, XML, and CSV files), and unstructured data (such as images or media files). This feature makes Amazon S3 the appropriate storage solution for your cloud data lake.

Data ingestion methods

A core capability of a data lake architecture is the ability to quickly and easily ingest multiple types of data:

- Real-time streaming data and bulk data assets, from on-premises storage platforms.
- Structured data generated and processed by legacy on-premises platforms - mainframes and data warehouses.
- Unstructured and semi-structured data – images, text files, audio and video, and graphs).

AWS provides services and capabilities to ingest different types of data into your data lake built on Amazon S3 depending on your use case. This section provides an overview of various ingestion services.

Amazon Kinesis Data Firehose

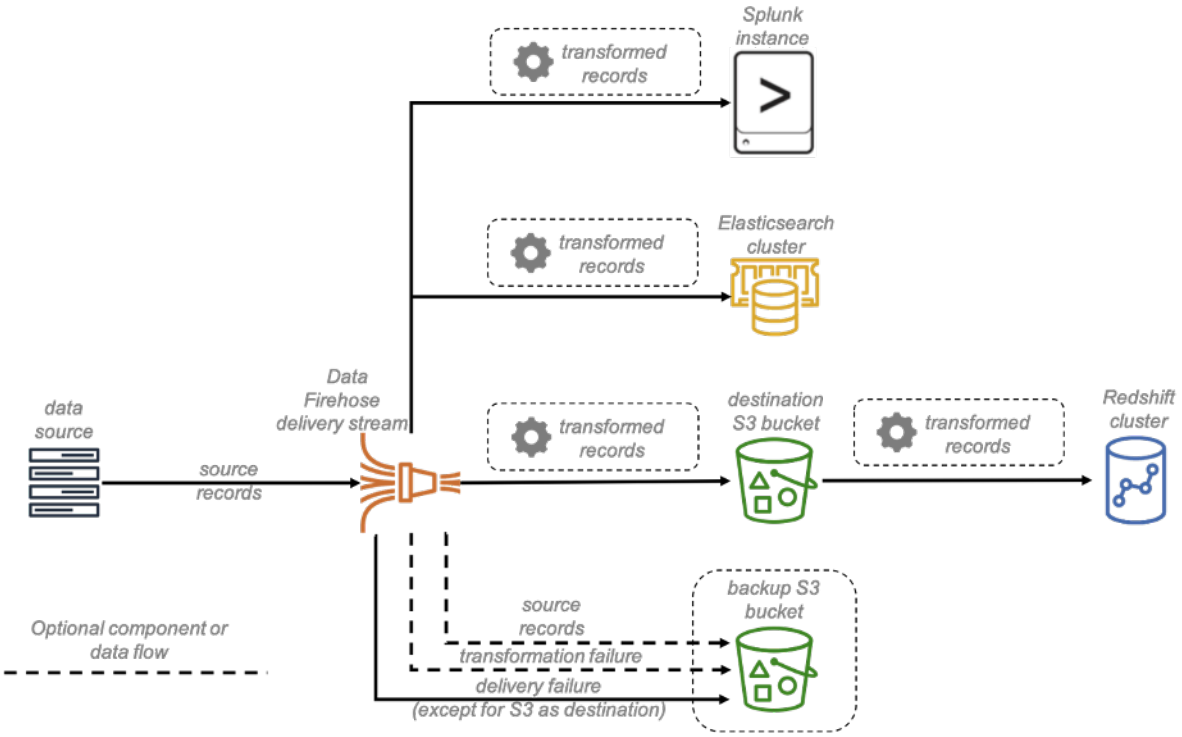
Amazon Data Firehose is part of the Kinesis family of services that makes it easy to collect, process, and analyze real-time streaming data at any scale. Firehose is a fully managed service for delivering real-time streaming data directly to data lakes (Amazon S3), data stores, and analytical services for further processing. Firehose automatically scales to match the volume and throughput of streaming data, and requires no ongoing administration. Firehose can also be configured to transform streaming data before it's stored in a data lake built on Amazon S3. Its transformation capabilities include compression, encryption, data batching, and Lambda functions. Firehose integrates with Amazon Kinesis Data Streams and Amazon Managed Streaming for Apache Kafka to deliver the streaming data into destinations, such as Amazon S3, Amazon Redshift, Amazon OpenSearch Service, and third-party solutions such as Splunk.

Firehose can convert your input JSON data to Apache Parquet and Apache ORC before storing the data into your data lake built on Amazon S3. Parquet and Orc being columnar data formats, help save space and allow faster queries on the stored data compared to row-based formats such as JSON. Firehose can compress data before it's stored in Amazon S3. It currently supports GZIP, ZIP, and SNAPPY compression formats. GZIP is the preferred format because it can be used by Amazon Athena, Amazon EMR, and Amazon Redshift.

Firehose also allows you to invoke Lambda functions to perform transformations on the input data. Using Lambda blueprints, you can transform the input comma-separated values (CSV), structured text, such as Apache Log and Syslog formats, into JSON first. You can optionally store the source

data to another S3 bucket. The following figure illustrates the data flow between Firehose and different destinations.

Firehose also provides the ability to group and partition the target files using custom prefixes such as dates for S3 objects. This facilitates faster querying by the use of the partitioning and incremental processing further with the same feature.



Delivering real-time streaming data with Kinesis Data Firehose to different destinations with optional backup

Firehose also natively integrates with Amazon Managed Service for Apache Flink which provides you with an efficient way to analyze and transform streaming data using Apache Flink and SQL applications. Apache Flink is an open-source framework and engine for processing streaming data using Java and Scala. Using Managed Service for Apache Flink, you can develop applications to perform time series analytics, feed real-time dashboards, and create real-time metrics. You can also use Managed Service for Apache Flink for transforming the incoming stream and create a new data stream that can be written back into Firehose before it is delivered to a destination.

Finally, Firehose encryption supports Amazon S3 server-side encryption with AWS Key Management Service (AWS KMS) for encrypting delivered data in your data lake built on Amazon S3. You can choose not to encrypt the data or to encrypt the data with a key from the list of AWS KMS keys that you own (refer to the **Data encryption with Amazon S3 and AWS KMS** section of

this document). Firehose can concatenate multiple incoming records, and then deliver them to Amazon S3 as a single S3 object.

This is an important capability because it reduces the load of Amazon S3 transaction costs and transactions per second. You can grant your application access to send data to Firehose using AWS Identity and Access Management (IAM). Using IAM, you can also grant Firehose access to S3 buckets, Amazon Redshift cluster, or Amazon OpenSearch Service cluster. You can also use Kinesis Data Firehose with virtual private cloud (VPC) endpoints (AWS PrivateLink). AWS PrivateLink is an AWS technology that enables private communication between AWS services using an elastic network interface with private IPs in your Amazon VPC.

AWS Snow Family

AWS Snow Family, comprised of AWS Snowball Edge, AWS Snowball Edge, and AWS Snowmobile, offers hardware devices of varying capacities for movement of data from on-premises locations to AWS. The devices also offer cloud computing capabilities at the edge for the applications that need to perform computations closer to the source of the data. Using Snowball Edge you can transfer data generated continuously from sensors, IoT devices, and machines, to the AWS Cloud. Snowball Edge features 8 TB of storage. Snowball Edge and Snowmobile are used to transfer massive amounts of data up to 100 PB.

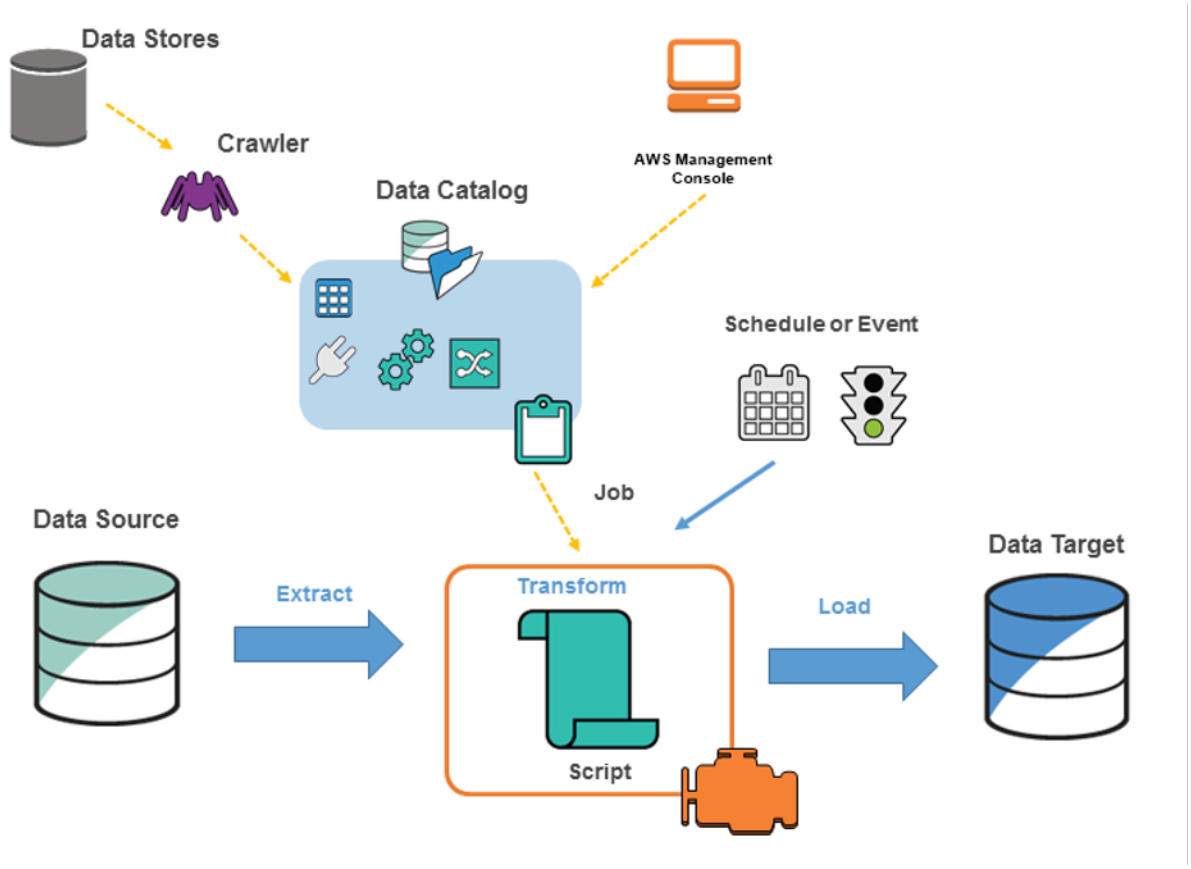
Snowball moves terabytes of data into your data lake built on Amazon S3. You can use it to transfer databases, backups, archives, healthcare records, analytics datasets, historic logs, IoT sensor data, and media content, especially in situations where network conditions hinder transfer of large amounts of data both into and out of AWS.

AWS Snow Family uses physical storage devices to transfer large amounts of data between your on-premises data centers and your data lake built on Amazon S3. You can use AWS Storage Optimized Snowball Edge to securely and efficiently migrate bulk data from on-premises storage platforms and Hadoop clusters. Snowball Edge supports encryption and uses AES-256-bit encryption. Encryption keys are never shipped with the Snowball device, so the data transfer process is highly secure.

Data is transferred from the Snowball Edge device to your data lake built on Amazon S3 and stored as S3 objects in their original or native format. Snowball Edge also has a Hadoop Distributed File System (HDFS) client, so data may be migrated directly from Hadoop clusters into an S3 bucket in its native format. Snowball Edge devices can be particularly useful for migrating terabytes of data from data centers and locations with intermittent internet access.

AWS Glue

AWS Glue is a fully managed serverless ETL service that makes it easier to categorize, clean, transform, and reliably transfer data between different data stores in a simple and cost-effective way. The core components of AWS Glue consists of a central metadata repository known as AWS Glue Data Catalog which is a drop-in replacement for an Apache Hive metastore (refer to the **Catalog and search** section of this document for more information) and an ETL job system that automatically generates Python and Scala code and manages ETL jobs. The following figure depicts the high-level architecture of an AWS Glue environment.



Architecture of an AWS Glue environment

To ETL the data from source to target, you create a job in AWS Glue, which involves the following steps:

1. Before you can run an ETL job, define a crawler and point it to the data source to identify the table definition and the metadata required to run the ETL job. The metadata and the table definitions are stored in the Data Catalog. The data source can be an AWS service, such as Amazon RDS, Amazon S3, Amazon DynamoDB, or Kinesis Data Streams, as well as a third-party

JDBC-accessible database. Similarly, a data target can be an AWS service, such as Amazon S3, Amazon RDS, and Amazon DocumentDB (with MongoDB compatibility), as well as a third-party JDBC-accessible database.

2. Either provide a script to perform the ETL job, or AWS Glue can generate the script automatically.
3. Run the job on-demand or use the scheduler component that helps in initiating the job in response to an event and schedule at a defined time.
4. When the job is run, the script extracts the data from the source, transforms the data, and finally loads the data into the data target.

AWS DataSync

AWS DataSync is an online data transfer service that helps in moving data between on-premises storage systems and AWS storage services, as well as between different AWS storage services. You can automate the data movement between on-premises Network File Systems (NFS), Server Message Block (SMB), or a self-managed object store to your data lake built on Amazon S3. DataSync allows data encryption and data integrity validation to ensure safe and secure transfer of data. DataSync also has support for an HDFS connector to read directly from on-premises Hadoop clusters and replicate your data to your data lake built on Amazon S3.

AWS Transfer Family

AWS Transfer Family is a fully managed and secure transfer service that helps you to move files into and out of AWS storage services (for example, your data lake built on Amazon S3 storage and Amazon Elastic File System (Amazon EFS) Network File System (NFS)). AWS Transfer Family supports Secure Shell (SSH) File Transfer Protocol (FTP), FTP Secure (FTPS), and FTP. You can use AWS Transfer Family to ingest data into your data lakes built on Amazon S3 from third parties, such as vendors and partners, to perform an internal transfer within the organization, and distribute subscription-based data to customers.

Storage Gateway

Storage Gateway can be used to integrate legacy on-premises data processing platforms with a data lake built on Amazon S3. The File Gateway configuration of Storage Gateway offers on-premises devices and applications a network file share through an NFS connection. Files written to this mount point are converted to objects stored in Amazon S3 in their original format without any

proprietary modification. This means that you can integrate applications and platforms that don't have native Amazon S3 capabilities—such as on-premises lab equipment, mainframe computers, databases, and data warehouses—with S3 buckets, and then use analytical tools such as Amazon EMR or Amazon Athena to process this data.

Apache Hadoop distributed copy command

Amazon S3 natively supports distributed copy (DistCp), which is a standard Apache Hadoop data transfer mechanism. This allows you to run DistCp jobs to transfer data from an on-premises Hadoop cluster to an S3 bucket. The command to transfer data is similar to the following:

```
hadoop distcp hdfs://source-folder s3a://destination-bucket
```

Direct Connect

Direct Connect establishes a dedicated network connection between your on-premises internal network and AWS. Direct Connect links the internal network to an Direct Connect location over a standard Ethernet fiber optic cable. Using the dedicated connection, you can create virtual interface directly with Amazon S3, which can be used to securely transfer data from on-premises into a data lake built on Amazon S3 for analysis.

AWS Database Migration Service

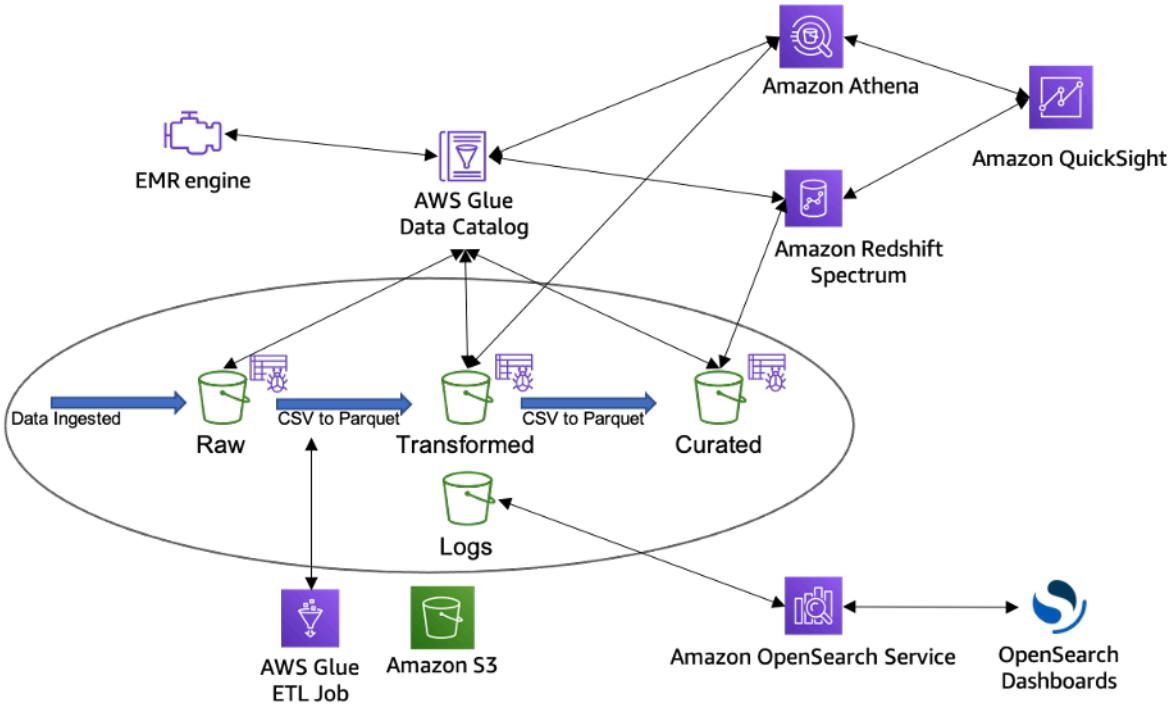
AWS Database Migration Service (AWS DMS) facilitates the movement of data from various data stores such as relational databases, NoSQL databases, data warehouses, and other data stores into AWS. AWS DMS allows one-time migration and ongoing replication (change data capture) to keep the source and target data stores in sync. Using AWS DMS, you can use Amazon S3 as a target for the supported database sources. AWS DMS task for Amazon S3 writes both full load migration and change data capture (CDC) in a comma separated value (CSV) format by default.

You can also write the data into Apache Parquet format (parquet) for more compact storage and faster query options. Both CSV and parquet formats are favorable for in-place querying using services such as Amazon Athena and Amazon Redshift Spectrum (refer to the **In-place querying** section of this document for more information). As mentioned earlier, Parquet format is recommended for analytical querying. It is useful to use AWS DMS to migrate databases from on-premises to or across different AWS accounts to your data lake built on Amazon S3 during initial data transfer or on a regular basis.

Data lake foundation

Amazon S3 provides the foundation for building a data lake, along with integration to other services that can be tailored to your business needs. A common challenge faced by users when building a data lake is the categorization of data and maintaining data across different stages as it goes through the transformation process.

The following figure depicts a sample data lake and the transformation journey data goes through in its lifecycle.



Sample data lake transformation journey

This section provides a recommended bucket strategy for building a data lake foundation.

A data lake can be broadly categorized across four distinct buckets:

- **Raw data** – Data ingested from the data sources in the raw data format, which is the immutable copy of the data. This can include structured, semi structured, and unstructured data objects such as databases, backups, archives, JSON, CSV, XML, text files, or images.
- **Transformed** – This bucket consists of transformed data normalized to a specific use case for performance improvement and cost reduction. In this stage, data can be transformed into

columnar data formats, such as Apache Parquet and Apache ORC, which can be used by Amazon Athena.

- **Curated** – The transformed data can be further enriched by blending it with other data sets to provide additional insights. This layer typically contains S3 objects which are optimized for analytics, reporting using Amazon Athena, Amazon Redshift Spectrum, and loading into massively parallel processing data warehouses such as Amazon Redshift.
- **Logs** – This bucket stores process logs for Amazon S3, and other services in the data lake architecture. The logs can include S3 access logs, CloudWatch logs, or CloudTrail logs.

The following table shows a recommended sample folder structure for your data lake per environment. Each environment can have the same folder structure with tags to segregate across each environment.

Recommended sample folder structure for your data lake per environment

Bucket	Raw	Transformed	Curated
Folder	/*<LOB>*/*<database-name>* /*<Table-name>* /Partition1/Partition2/.../data files	/*<LOB>*/*<database-name>* /*<Table-name>* /Partition1/Partition2/.../data files	/*<BU>*/*<database-name>* /*<Table-name>* /Partition1/Partition2/.../data files
Tagging	Object level	Object level	Object level
File format	Source defined	User defined (Recommended: Apache Parquet)	Final processed files (Recommended: Apache Parquet)
Compression	Can be applied for cost optimization	Snappy	Snappy
Lifecycle policy	Driven by individual object tags	Can have a bucket or object-level policy	Driven by individual object tags
Access	No user access. Use TBAC to enforce	Users can have access to the bucket via AWS	Users can have access to this data through

Bucket	Raw	Transformed	Curated
	services principals access.	Lake Formation - AWS IAM	AWS Lake Formation to a specific dataset or tables. (Column-level access is enabled and recommended for access policies.)
Partitioning	By source process date	By business date	By business date
File sizes	Source-defined	User defined (Recommended file size: 128 MB)	• 128 MB recommended for use by Athena/Spectrum querying • Amazon Redshift load: Number of files of roughly equal size, which are a multiple of the total data slices on Amazon Redshift roughly. Can range from 1 MB to 1 GB.
Encryption	AWS KMS/AWS CloudHSM/HSM	AWS KMS/AWS CloudHSM/HSM	AWS KMS/AWS CloudHSM/HSM

It is recommended you follow best practices when defining your bucket strategy for your data lake built on Amazon S3:

- Buckets names must be unique within a partition. Currently there are three partitions (`aws` – Standard Regions, `aws-cn` – China Regions, `aws-us-gov` – AWS GovCloud (US); however, names can be reused after the buckets are deleted (with several exceptions).

- Deleted bucket names are not available to reuse immediately; hence, if users want to use a particular bucket name, they should not delete the bucket.
- All bucket names across all AWS Regions should comply with DNS naming conventions.
- Buckets can store an unlimited number of objects in a bucket without impacting performance. The objects can be stored in a single bucket or across multiple buckets; however, you cannot create a bucket from within another bucket.
- Production S3 buckets should be hosted under a different AWS account, separate from non-production workloads.
- Build an automatic ingestion mechanism to catalog and create the multiple layers of data storage including Raw, Transformed, and Curated.
- Consider building automatic data classification rules based on schema and data.
- Consider additional folders within the data lakes, such as reports, downstream applications, or user folders.
- Enable versioning, if needed, for protection from accidental deletes.
- Use separate buckets for S3 data which needs to be replicated.

Catalog and search

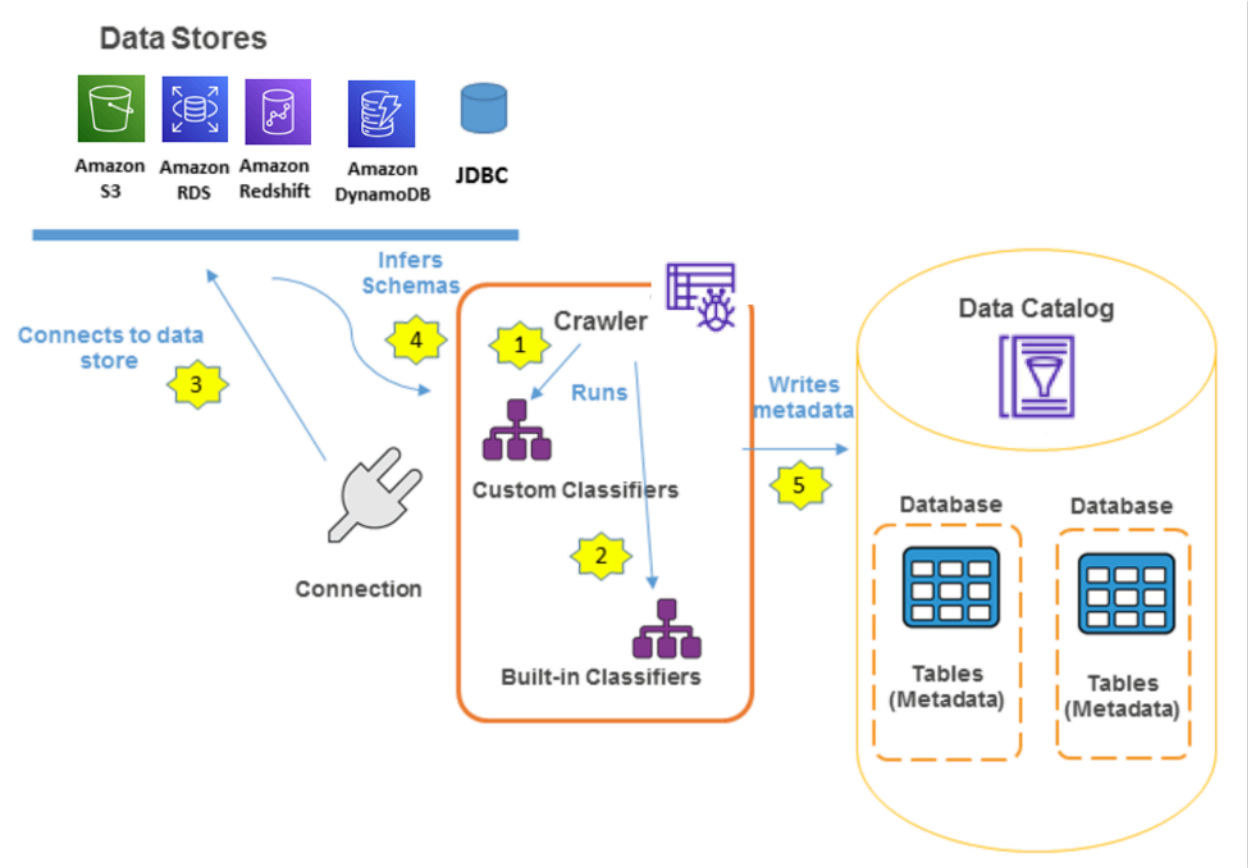
The earliest challenges that inhibited building a data lake were keeping track of all of the raw assets as they were loaded into the data lake, and then tracking all of the new data assets and versions that were created by data transformation, data processing, and analytics. Thus, an essential component of a data lake built on Amazon S3 is the Data Catalog. The Data Catalog provides an interface to query all assets stored in data lake S3 buckets. The Data Catalog is designed to provide a single source of truth about the contents of the data lake.

AWS Glue

AWS Glue is a fully managed ETL service that makes it easier to categorize, clean, transform, and reliably transfer data between different data stores. The AWS Glue Data Catalog, a component of AWS Glue, provides a unified metadata repository for performing analytical operations across various data sources, such as Amazon EMR, Amazon Athena, Amazon Redshift, and Amazon Redshift Spectrum, and any application that is compatible with a Hive metastore.

To create a Data Catalog, use AWS Glue crawlers that crawl the data from the data sources as registered with AWS Glue in the AWS Management Console. The data source can be an S3 bucket, a table in Amazon RDS, Amazon DynamoDB, or Amazon Redshift, or any external database that supports JDBC connectivity.

You can define custom classifiers or use the built-in classifiers provided by AWS Glue to classify the data. AWS Glue provides classifiers for common file types, such as CSV, JSON, AVRO, or XML. AWS Glue also provides classifiers for common relational database management systems using a JDBC connection. You can create a custom classifier using a grok pattern, an XML tag, JavaScript Object Notation (JSON), or comma-separated values (CSV). The following figure depicts the working of AWS Glue in building a Data Catalog.



AWS Glue Data Catalog

Data Catalog is a database that stores metadata in tables consisting of data schema, data location, and runtime metrics. Data Catalog is also Apache Hive metastore compatible that can be used as a central repository for storing structural and operational metadata. AWS Glue also provides out-of-box integration with Amazon EMR that allows you to use Data Catalog as an external Hive metastore. Data Catalog is recommended, especially when you need a persistent metastore or a metastore shared between different applications, services, clusters, or AWS accounts. Data Catalog can also be used to create an external table for Athena or Amazon Redshift.

AWS Lake Formation

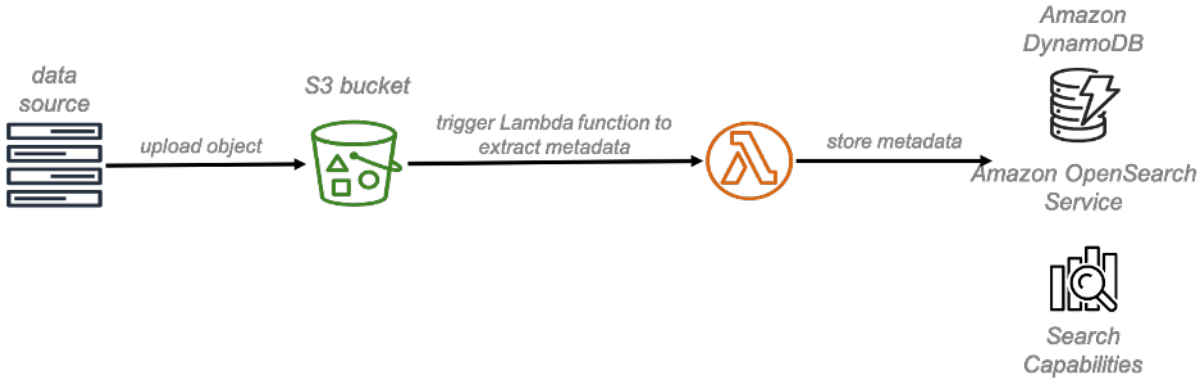
AWS Lake Formation helps to easily build, secure, and manage data lakes. Lake Formation provides centralized governance and access control for the data in a data lake built on S3, and controls access to the data through various services, such as AWS Glue, Athena, Amazon Redshift Spectrum, Quick, and Amazon EMR. AWS Lake Formation can connect to an S3 bucket and orchestrate a dataflow that can ingest, clean, transform, and organize the raw data.

Lake Formation uses AWS Glue Data Catalog to automatically classify data in data lakes, data sources, transforms, and targets. Apart from the metadata, the Data Catalog also stores information consisting of resource links to shared databases and tables in external accounts for cross account access to the data in a data lake built on S3.

Lake Formation provides you with a grant/revoke permission model to control access to Data Catalog resources (consisting of database and metadata tables), S3 buckets and underlying data in these buckets. Lake Formation permissions along with IAM policies provide granular access to the data stored in data lakes built on S3. These permissions can be used to share Data Catalog resources with external AWS accounts. The users from these accounts can run jobs and queries by combining data from multiple data catalogs across multiple accounts.

Comprehensive data catalog

You have the flexibility to create a comprehensive data catalog using standard AWS services such as AWS Lambda, DynamoDB, and Amazon OpenSearch Service. At a high level, AWS Lambda triggers are used to populate DynamoDB tables with object names and metadata when those objects are put into S3; Amazon OpenSearch Service is used to search for specific assets, related metadata, and data classifications. The following figure shows a high-level architectural overview of this solution.



Comprehensive data catalog using AWS Lambda, DynamoDB, and Amazon OpenSearch Service

Transforming data assets

A core value of a data lake is that it is the collection point and repository for all of an organization's data assets, regardless of their native formats. This allows quick ingestion, elimination of data duplication and data sprawl, and centralized governance and management. After the data assets are collected, they need to be transformed into normalized formats to be used by a variety of data analytics and processing tools.

The key to 'democratizing' the data and making the data lake available to the widest number of users of varying skill sets and responsibilities is to transform data assets into a format that allows for efficient one-time SQL querying. Data transformation or ETL process prepares the data to be consumed by downstream systems for advanced analytics, visualizations and business reporting, or machine learning.

The first step involves extracting information from the different data sources which was discussed in the Data ingestion methods section. The data is stored in a raw bucket (refer to the **Data lake foundation** section of this document). After you have the data, you need to transform it. Transformations might involve aggregating the data from different sources or changing the file format of the data received, such as from CSV to Parquet format to reduce the file size and optimize the Athena query.

You can compress the files or convert the data format from string to double, or number to date format. After the transformation is complete, the data can be written into transformed bucket (refer to the **Data lake foundation** section of this document) which is then further used by the downstream systems.

There are multiple ways to transform data assets, and the "best" way often comes down to the nature of the analytics application, individual preference, skill sets, and the tools available. When a data lake is built on AWS, there is a wide variety of tools and services available for data transformation, so you can pick the methods and tools that best suits your purpose. Because the data lake is inherently multi-tenant, multiple data transformation jobs using different tools can run concurrently.

There are two common and straightforward methods to transform data assets into Parquet in a data lake built on S3 using Amazon EMR clusters. The first method involves creating an Amazon EMR cluster with Hive installed using the raw data assets in Amazon S3 as input, transforming those data assets into Hive tables, and then writing those Hive tables back out to Amazon S3 in

Parquet format. The second method is to use Spark on Amazon EMR. With this method, a typical transformation can be achieved with only 20 lines of PySpark code.

A third data transformation method on a data lake built on S3 is to use AWS Glue. AWS Glue simplifies and automates difficult and time-consuming data discovery, conversion, mapping, and job scheduling tasks. AWS Glue guides you through the process of transforming and moving your data assets with an easy-to-use console that helps you understand your data sources, transform and prepare these data assets for analytics, and load them reliably from S3 data sources back into S3 destinations.

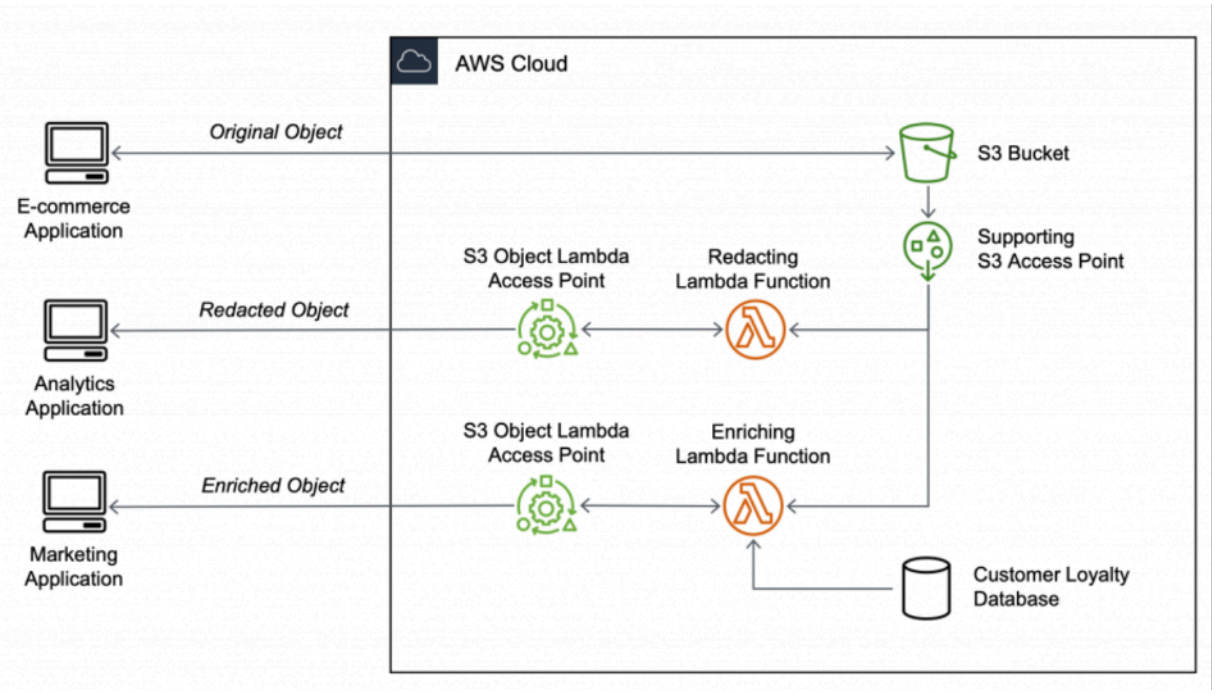
AWS Glue automatically crawls raw data assets in your data lake's S3 buckets, identifies data formats, and then suggests schemas and transformations so that you don't have to spend time hand coding data flows. You can then edit these transformations, if necessary, using the tools and technologies you already know, such as Python, Spark, Git, and your favorite integrated developer environment, and then share them with other AWS Glue users of the data lake. AWS Glue's flexible job scheduler can be set up to run data transformation flows on a recurring basis, in response to triggers, or even in response to AWS Lambda events.

AWS Glue automatically and transparently provisions hardware resources, and distributes ETL jobs on Apache Spark nodes so that ETL run times remain consistent as data volume grows. AWS Glue coordinates the implementation of data lake jobs in the right sequence, and automatically retries failed jobs. With AWS Glue, there are no servers or clusters to manage, and you pay only for the resources consumed by your ETL jobs.

AWS Lambda functions can also be used for transforming the data stored in your data lake built on Amazon S3. Lambda functions can respond to event notifications from Amazon S3 when an object is created or deleted. You can configure a Lambda function to perform an action asynchronously, based on the event.

When data is stored in a data lake built on S3, you can share the data with multiple applications. These applications can vary in nature and purpose (for example, ecommerce applications, analytics applications, and marketing applications) and necessitate a different view for each application. S3 Object Lambda can be used to add your own code to process the data retrieved before returning it to an application (for example, masking the personally identifiable information (PII) data or masking credit card information before transferring it to the application).

S3 Object Lambda uses a Lambda function to automatically process and transform the data as it is being retrieved from an S3 bucket. The following figure shows an example of an S3 Object Lambda.



Using S3 Object Lambda to transform data before retrieval by applications

S3 Object Lambda can be very useful in redacting PII data for analytics applications, format conversion, enriching data from other data sources, resizing objects, or even implementing custom authorization rules to access the data stored in your data lake built on Amazon S3.

Another transformation feature that S3 supports natively is batch operations. Using S3 Batch Operations, you can define an operation to be performed on the exabytes of objects stored in your bucket. S3 Batch Operations manages the entire lifecycle of the batch operation by tracking the progress, sending notifications, and storing a detailed completion report for all the operations performed on the objects.

By using S3 Batch Operations, you can copy objects from a bucket into another bucket in the same or different Region, or invoke a Lambda function to transform the object. S3 Batch Operation can also be used to manage the tags defined for the objects and restoring a large number of objects from Amazon Glacier. S3 Batch Operation, along with S3 Object Lock, can be used to manage retention dates and legal holds to apply compliance and governance rules for multiple objects at the same time. S3 Object Lock works on the write once read many (WORM) model that can help in prevention of accidental deletion of objects. It can also help in adhering to regulatory controls.

In-place querying

One of the most important capabilities of a data lake on AWS is the ability to perform in-place transformation and querying of data assets. This allows users to run sophisticated analytical queries directly on their data stored in S3, without having to copy and load data into separate analytics platforms or data warehouses.

There are various tools to perform in-place querying for data stored in a data lake, such as Presto on Amazon EMR and various partner tools. This section provides an overview of serverless services that not only helps perform in-place querying, but also avoids the procurement and management of servers.

Users can query S3 data without any additional infrastructure, and only pay for the queries that they run. This makes the ability to analyze vast amounts of unstructured data accessible to any data lake user who can use SQL, and makes it far more cost effective than the traditional method of performing an ETL process, creating a Hadoop cluster or data warehouse, loading the transformed data into these environments, and then running query jobs. AWS Glue, as described in the previous sections, provides the data discovery and ETL capabilities, and Amazon Athena and Amazon Redshift Spectrum provides the serverless in-place querying capabilities.

In addition to in-place querying using Athena and Redshift Spectrum, S3 also provides capabilities to retrieve subset of your data through S3 Select and Amazon Glacier Select, that improves the performance of accessing large amounts of data from your data lake built on S3. Using S3 Select, users can run SQL statements to filter and retrieve only a subset of data stored in their data lake. S3 Select operates on objects stored in CSV, JSON, or Apache Parquet format, and other compression formats such as GZIP or BZIP2. Users can also delimit the result set, thus, reducing latency to retrieve the data and optimizing cost.

Amazon Athena

Amazon Athena is an interactive query service that makes it easier for you to analyze data directly in S3 using standard SQL. With a few actions in the AWS Management Console, you can use Athena directly against data assets stored in the data lake built on S3 and begin using standard SQL to run one-time queries and get results in a matter of seconds.

Athena is serverless, so there is no infrastructure to set up or manage, and you only pay for the volume of data assets scanned during the queries you run.

Athena scales automatically—running queries in parallel—so results are fast, even with large datasets and complex queries. You can use Athena to process unstructured, semi-structured, and structured data sets. Supported data asset formats include CSV, JSON, or columnar data formats such as Apache Parquet and Apache ORC. You can also use Athena to run one-time queries using ANSI SQL without first aggregating or loading the data into Athena. Athena integrates with Quick for easy data visualization. It can also be used with third-party reporting and business intelligence tools by connecting these tools to Athena with a JDBC driver.

Athena also natively integrates with AWS Glue Data Catalog which provides a persistent metadata store for the data stored in S3. You can create the table and use Athena to query the data based on a metadata store that integrates with the ETL and data discovery features of AWS Glue.

When querying an existing table, Athena uses Presto under the hood, a distributed SQL engine. Athena can also be used to query S3 inventory using standard SQL. Amazon S3 Inventory is an S3 tool to help manage storage. You can use the tool to audit and report on the replication and encryption status of the objects for business, compliance, and regulatory needs. Athena supports querying S3 inventory files in ORC, Parquet, or CSV format. AWS recommends using ORC-formatted or Parquet-formatted inventory files because these formats provide faster query performance and lower query costs. ORC and Parquet formats are columnar formats that allows the reader to read, decompress, and process the columns that are only required for the current query.

Amazon Redshift Spectrum

Another way to perform in-place querying of data assets in a data lake built on Amazon S3 is to use Amazon Redshift Spectrum. Amazon Redshift is a large-scale, managed data warehouse service that supports massive parallel processing. By contrast, Amazon Redshift Spectrum allows you to run Redshift SQL queries directly against massive amounts of data—up to exabytes—stored in a data lake built on Amazon S3. Amazon Redshift Spectrum applies sophisticated query optimization, scaling processing across thousands of nodes, so results are fast—even with large data sets and complex queries. Amazon Redshift Spectrum can directly query a wide variety of data assets stored in the data lake, including CSV, TSV, Parquet, Sequence, and RCFile. Because Amazon Redshift Spectrum supports the SQL syntax of Amazon Redshift, you can run sophisticated queries using the same business intelligence tools that you use today. You also have the flexibility to run queries that span frequently accessed data assets that are stored locally in Amazon Redshift, and your full data sets stored in S3.

Because Amazon Athena and Amazon Redshift share a common data catalog and common data formats, you can use both Athena and Amazon Redshift Spectrum against the same data assets. You would typically use Athena for one-time data discovery and SQL querying, and then use Amazon Redshift Spectrum for more complex queries and scenarios where a large number of data lake users want to run concurrent business intelligence and reporting workloads.

The broader analytics portfolio

With a data lake built on AWS, data assets get ingested and stored in one massively scalable, low cost, performant platform. Data discovery, transformation, and SQL querying can all be done in place using innovative AWS services such as AWS Glue, Amazon Athena, and Amazon Redshift Spectrum. In addition, there are a wide variety of other AWS services that can be directly integrated with S3 to create any number of sophisticated analytics, machine learning, and artificial intelligence data processing pipelines.

This allows you to quickly solve a wide range of analytics business challenges on a single platform, against common data assets, without having to worry about provisioning hardware and installing and configuring complex software packages before loading data and performing analytics. Plus, you only pay for what you consume. Some of the most common AWS services that can be used with data assets in a data lake built on S3 are describedin this section.

Amazon EMR

Amazon EMR is a highly distributed computing framework used to quickly and easily process data in a cost-effective manner. Amazon EMR uses Apache Hadoop, an open-source framework, to distribute data and processing across an elastically resizable cluster of EC2 instances and allows you to use all the common Hadoop tools such as Hive, Pig, Spark, Flink, Hudi, Hue, Livy, MXNet, Presto, TensorFlow, HBase, or Zeppelin. Amazon EMR does all the heavily lifting involved with provisioning, managing, and maintaining the infrastructure and software of a Hadoop cluster, and is integrated directly with S3.

With Amazon EMR, you can launch a persistent cluster that stays up indefinitely or a temporary cluster that ends after the analysis is complete. In either scenario, you only pay for the hours the cluster is up. Amazon EMR supports a variety of EC2 instance types encompassing general purpose, compute, memory, storage I/O optimized, and GPU instances (for example, M6g, C5, R5, Z1, I3, D2, G4, and P3), and all Amazon EC2 pricing options (On-Demand, Reserved, and Spot). The latest version of EMR 6.1.0 (at the time of writing this document) also supports ARM based instance types.

AWS Graviton processors are custom built by Amazon Web Services and use 64-bit Arm Neoverse cores to deliver the best price performance. When you launch an Amazon EMR cluster (called a *job flow*), you choose how many and what type of EC2 instances to provision. Organizations with many different lines of business and a large number of users can build a single data lake solution,

store their data assets in S3, and then spin up multiple EMR clusters to share data assets in a multi-tenant fashion.

Amazon EMR integrates with S3 for input, output, and intermediate data storage. You can choose the HDFS which runs on a cluster consisting of primary and core nodes for processing data that does not need to persist after the lifecycle of the cluster. You can use Amazon EMR file system (EMRFS) using Amazon S3 as a data layer for applications running on your cluster. The data can be stored on EMRFS to persist even after the cluster lifecycle. This mechanism helps separate compute and storage, thus enabling you to independently scale the compute and storage depending on the workload. You can scale your compute needs by resizing your cluster, and you can scale your storage needs by using Amazon S3.

Amazon SageMaker AI

Machine learning is another important data lake use case. Amazon SageMaker AI is a fully managed machine learning service that provides a set of features to build, train, and deploy machine learning models. SageMaker AI provides a suite of built-in algorithms and mechanisms to build custom algorithms based on user requirements. SageMaker AI also provides an integrated Jupyter notebook instance to prepare and transform the data, develop code for model training, and subsequently deploy, test, and validate the models.

SageMaker AI provides visualization tools and wizards to guide you through the process of creating machine learning models without having to learn complex algorithms and technology. SageMaker AI can create machine learning models based on datasets stored in your data lake built on S3. SageMaker AI greatly enhances machine learning capabilities by combining with your data lake built on S3. For training machine learning models, a large amount of data is required, and a data lake backed by S3 and Amazon Glacier provides cost-effective storage.

Quick

Quick is a fast business analytics service that makes it easy for you to build visualizations, perform targeted analysis, and quickly get business insights from your data assets stored in the data lake, any time, on any device. You can use Quick to seamlessly discover AWS data sources such as Amazon Redshift, Amazon RDS, Amazon Aurora, Amazon Athena, and Amazon S3, connect to any of these data sources and data assets, and get insights from this data in minutes. Quick allows organizations using the data lake to seamlessly scale their business analytics capabilities to hundreds of thousands of users. It delivers fast and responsive query performance by using a Super-fast, Parallel, In-memory Calculation Engine (SPICE).

Amazon OpenSearch Service

Amazon OpenSearch Service is a managed service where you can deploy and manage OpenSearch clusters at scale. Common use cases include log analytics, real-time application monitoring, and clickstream analytics. Amazon OpenSearch Service can be used to add a search box to a website, analyze metrics, security event data, application and infrastructure logs, or store data to automate business workflows.

Amazon OpenSearch Service manages the infrastructure provisioning and maintenance, thus reducing the overhead of managing the Amazon OpenSearch Service cluster yourself. Amazon OpenSearch Service integrates with various open source and native AWS services to provide a seamless experience. Amazon OpenSearch Service integrates with open source Logstash to conveniently upload bulk data into Amazon OpenSearch Service domain and with open source Kibana for visualization.

Amazon OpenSearch Service also runs with Firehose for ingestion of streaming data, AWS CloudTrail, and Amazon CloudWatch for logging and monitoring, respectively. You can also use your data lake built on Amazon S3 to store the streaming data before it is ingested into the Amazon OpenSearch Service domain. Ultrawarm storage and Cold storage, types of Amazon OpenSearch Service storage, use Amazon S3 for storing infrequently-accessed read-only data in a cost-efficient manner.

Securing, protecting, and managing data

Building a data lake, and making it the centralized repository for assets that were previously duplicated and placed across many siloes of smaller platforms and groups of users, requires implementing stringent and fine-grained security and access controls along with methods to protect and manage the data assets. A data lake solution on AWS—with S3 as its core—provides a robust set of features and services to secure and protect your data against both internal and external threats, even in large, multi-tenant environments. Additionally, innovative S3 data management features allow automation and scaling of data lake storage management, even when it contains billions of objects and petabytes of dataassets.

Securing your data lake begins with implementing very fine-grained controls that allow authorized users to view, access, process, and modify particular assets, and ensure that unauthorized users are blocked from taking any actions that would compromise data confidentiality and security. A complicating factor is that access roles may evolve over various stages of a data asset's processing and lifecycle. Fortunately, Amazon has a comprehensive and well-integrated set of security features, such as access policy, resource-based policies, bucket policies, and data encryption, to secure a data lake built on S3.

Access policy options

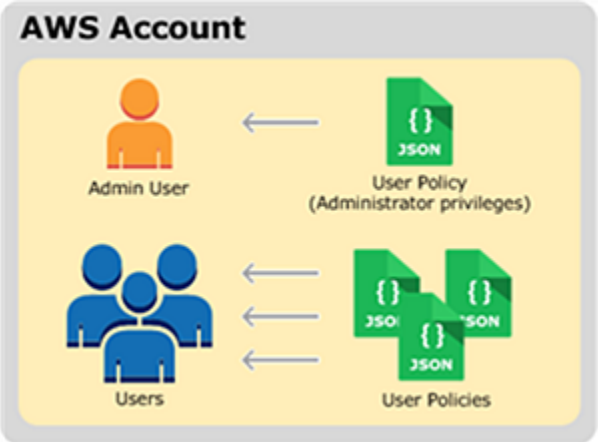
You can manage access to your S3 resources using access policy options. By default, all S3 resources—buckets, objects, and related sub-resources—are private (only the resource owner, an AWS account that created them, can access the resource). The resource owner can then grant access permissions to other users by writing an access policy. S3 access policy options are broadly categorized as resource-based policies and user policies. Access policies that are attached to resources are referred to as *resource-based policies*. Examples of resource-based policies include bucket policies and access-control lists. Access policies that are attached to users in an account are called *user policies*. Typically, a combination of resource-based and user policies are used to manage permissions to S3 buckets, objects, and other resources.

When you are creating a centralized data lake, you want to provide access to various different accounts and users from different accounts. You can define a bucket policy to manage permissions for cross-account access, and manage permissions for users in another account. Bucket policies can be defined based on various factors, such as S3 operations, type of requestor, type of resources, and the nature of the request. A bucket policy defined for a bucket is applicable for all the objects

in the bucket. This way you can manage the permissions for all the objects centrally instead of managing access for individual objects. Bucket policies can also be customized as per the user requirements, for example, a bucket policy can be defined to allow access to a particular S3 bucket for a time interval when the request is originated from a particular Classless Inter-Domain Routing (CIDR) block.

Similar to bucket policies, access-control lists are also used for managing permissions to a bucket or an object. Access-control lists differ from bucket policies in multiple ways. Access-control lists can only provide read/write permissions to an object or bucket. Additionally, access-control lists can be used when granting bucket permission to other accounts or users in other accounts. Object access-control lists are helpful when you want to manage access at an object level. The bucket owner can grant permission to another account to upload the object using object access-control lists. However, bucket access-control lists can be used to provide access to the S3 Log Delivery group to write access logs to the S3 bucket.

For most data lake environments, AWS recommends using user policies, so that permissions to access data assets can also be linked to user roles and permissions for the data processing and analytics services and tools that your data lake users will use. User policies are also recommended to be used if you want to provide access to a user for objects in a bucket. User policies are associated with IAM, which allows you to securely control access to AWS services and resources. With IAM, you can create users, groups, and roles in accounts and then attach access policies to them that grant access to AWS resources, including S3. The model for user policies is shown inthe following figure. For more details and information on securing S3 with userpolicies and IAM, refer to Amazon S3 security and Identity and access management in Amazon S3.



Model for user policies

Data Encryption with Amazon S3 and AWS KMS

Although user policies and IAM control who can review and access data in your data lake built on S3, it's also important to ensure that users who might inadvertently or maliciously gain access to those data assets can't review and use them. This is accomplished by using encryption keys to encrypt and decrypt data assets. S3 supports multiple encryption options.

AWS KMS helps scale and simplify management of encryption keys. AWS KMS gives you centralized control over the encryption keys used to protect your data assets. You can create, import, rotate, disable, delete, define usage policies for, and audit the use of encryption keys used to encrypt your data. AWS KMS is integrated with several other AWS services, making it easy to encrypt the data stored in these services with encryption keys. AWS KMS is integrated with AWS CloudTrail, which provides you with the ability to audit who used which keys, on which resources, and when.

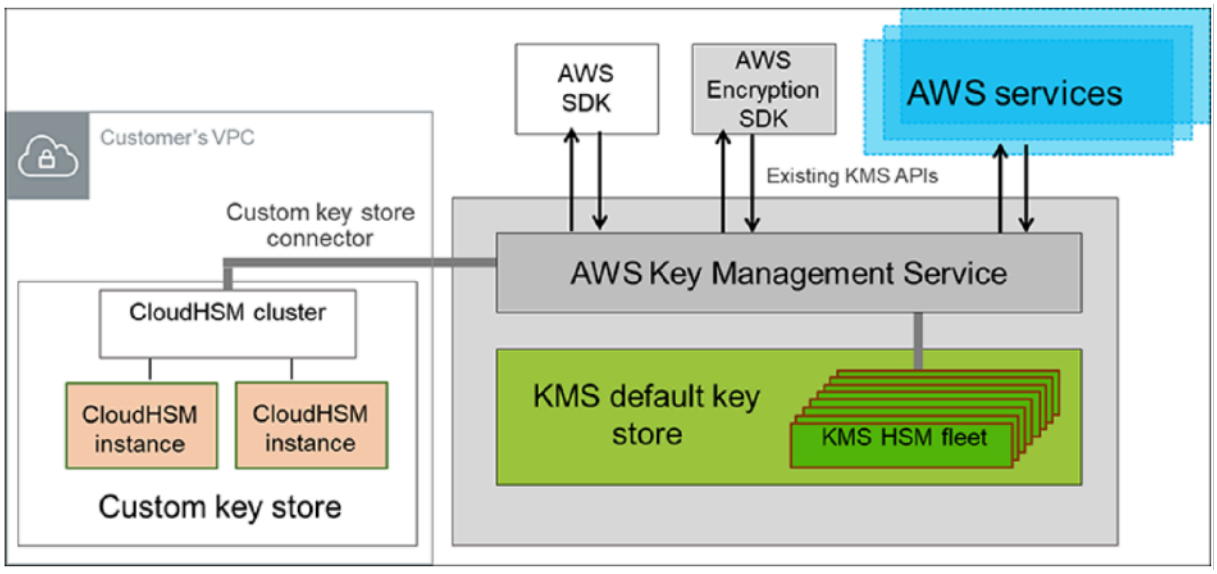
Data lakes built on AWS primarily use two types of encryption: server-side encryption and client-side encryption. Server-side encryption provides data-at-rest encryption for data written to S3. With server-side encryption, S3 encrypts user data assets at the object level, stores the encrypted objects, and then decrypts them as they are accessed and retrieved. With client-side encryption, data objects are encrypted before they are written into S3. For example, a data lake user can specify client-side encryption before transferring data assets into S3 from the internet, or can specify that services such as Amazon EMR, Amazon Athena, or Amazon Redshift use client-side encryption with S3.

Server-side encryption and client-side encryption can be combined for the highest levels of protection. Given the intricacies of coordinating encryption key management in a complex environment, such as a data lake, AWS strongly recommends using AWS KMS to coordinate keys across client-side and server-side encryption and across multiple data processing and analytics services.

For even greater levels of data lake protection, other services, such as Amazon API Gateway, Amazon Cognito, and IAM, can be combined to create a "shopping cart" model for users to check in and check out data lake assets. This data Lake on AWS solution architecture is built on S3 solution reference architecture.

Based on the compliance requirements, PII data must be handled separately and setup encryption with dedicated hardware. However, using a dedicated hardware hosted in the security account will introduce additional latency during data processing. You can use AWS CloudHSM or HashiCorp Vault to store the keys. By using CloudHSM you can configure a CloudHSM cluster to store your custom keys and authorize AWS KMS to use it as a dedicated key store. The following figure shows

the primary components of CloudHSM and a cluster of two CloudHSM instances connected to AWS KMS to create a customer-controlled key store.



Customer-controlled key store using AWS CloudHSM cluster and AWS KMS

Protecting data with Amazon S3

A vital function of a centralized data lake is data asset protection—primarily protection against corruption, loss, and accidental or malicious overwrites, modifications, or deletions. Amazon S3 has several intrinsic features and capabilities to provide the highest levels of data protection when it is used as the core platform for a data lake.

Data protection rests on the inherent durability of the storage platform used. Durability is defined as the ability to protect data assets against corruption and loss. Amazon S3 provides 99.999999999% data durability, which is 4 to 6 orders of magnitude greater than that which most on-premises, single-site storage platforms can provide. Put another way, the durability of Amazon S3 is designed so that 10,000,000 data assets can be reliably stored for 10,000 years.

Amazon S3 achieves this durability in all 24 of its global Regions by using multiple Availability Zones. Availability Zones consist of one or more discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities. Availability Zones offer the ability to operate production applications and analytics services, which are more highly available, fault tolerant, and scalable than would be possible from a single data center. Data written to Amazon S3 is redundantly stored across three Availability Zones and multiple devices within each Availability Zone to achieve 99.9999999% durability. This means that even in the event of an entire data center failure, data would not be lost.

In addition to core data protection, another key element for data assets is to protect against unintentional and malicious deletion and corruption, whether through users accidentally deleting data assets, applications inadvertently deleting or corrupting data, or rogue actors trying to tamper with the data. This becomes especially important in a large multi-tenant data lake, which will have a large number of users, many applications, and constant necessary data processing and application development.

S3 provides versioning to protect data assets against these scenarios. When enabled, S3 versioning will retain multiple copies of a data asset. When an asset is updated, prior versions of the asset will be retained and can be retrieved at any time. If an asset is deleted, the last version of it can be retrieved.

Data asset versioning can be managed by policies, to automate management at large scale. These policies can be combined with other Amazon S3 capabilities, such as lifecycle management for long-term retention of versions on lower cost storage tiers. Examples of such storage tiers include Amazon Glacier, and Multi-Factor-Authentication (MFA) Delete. These lower cost tiers require a second layer of authentication—typically through an approved external authentication device—to delete data asset versions. Even though S3 provides 99.999999999% data durability within an AWS Region, many enterprise organizations may have compliance and risk models that require them to replicate their data assets to a second geographically distant location and build disaster recovery architectures in a second location.

S3 Cross-Region Replication is an integral S3 capability that automatically and asynchronously copies data assets from a data lake in one AWS Region to a data lake in a different AWS Region. The data assets in the second Region are exact replicas of the source data assets that they were copied from, including their names, metadata, versions, and access controls. All data assets are encrypted during transit with SSL to ensure the highest levels of data security.

Another Amazon S3 feature is S3 Object Lock. This feature allows you to write objects using the WORM model. You can use Object Lock for scenarios which make it imperative that the data is not updated or deleted after it is written. You can also use Object Lock to satisfy compliance regulations in the financial and healthcare sectors, or retain the original copies of the data for reconciliation and auditing purposes. Object Lock has been assessed for SEC Rule 17a-4(f), FINRA Rule 4511, and CFTC Regulation 1.31 by Cohasset Associates. You can download a copy of the Cohasset Associates Assessment report.

When creating a data lake on S3, you store millions of objects in the data lake, necessitating an information asset register to keep track of the different objects, their replication status,

updates, and versions. You can use Amazon S3 Inventory to audit and report on the replication and encryption status of the objects for business, compliance, and regulatory needs.

All of these S3 features and capabilities—when combined with other AWS services such as IAM, AWS KMS, Amazon Cognito, and Amazon API Gateway—ensure that a data lake using Amazon S3 as its core storage platform will meet the most stringent data security, compliance, privacy, and protection requirements. Amazon S3 includes a broad range of certifications, including PCI-DSS, HIPAA/HITECH, FedRAMP, SEC Rule 17-a-4, FISMA, EU Data Protection Directive, and many other global agency certifications. These levels of compliance and protection allow organizations to build a data lake on AWS that operates more securely and with less risk than one built in their own on-premises data centers.

Managing data with object tagging

Because data lake solutions are inherently multi-tenant, with many organizations, lines of businesses, users, and applications using and processing data assets, it becomes very important to associate data assets to all of these entities and set policies to manage these assets coherently. S3 has introduced a new capability—object tagging—to assist with categorizing and managing S3 data assets. An object tag is a mutable key-value pair. Each S3 object can have up to 10 object tags. Each tag key can be up to 128 Unicode characters in length, and each tag value can be up to 256 Unicode characters in length. For example, suppose an object contains protected health information (PHI) data. A user, administrator, or application that uses object tags might tag the object using the key-value pair PHI=True or Classification=PHI.

In addition to being used for data classification, object tagging offers other important capabilities. Object tags can be used in conjunction with IAM to enable fine-grained controls of access permissions. For example, a particular data lake user can be granted permissions to only read objects with specific tags.

Object tags can also be used to manage S3 data lifecycle policies, which is discussed in the **Monitoring and optimizing the data lake environment** section of this document. A data lifecycle policy can contain tag-based filters. Finally, object tags can be combined with Amazon CloudWatch metrics and AWS CloudTrail logs—also discussed in the **Monitoring and optimizing the data lake environment** section of this document—to display monitoring and action audit data by specific data asset tag filters. In addition to object tags, you can also enable cost allocation tags for your S3 buckets to track storage cost across projects. Cost allocation tags are used for itemized costs for resources in the cost allocation report.

The following tags are recommended based on the requirements and specific use cases.

Table 1 — Recommended tags

Key	Value	Description
Owner	Team	Can be used in different stages of the bucket, and in user buckets to create data ownership.
Security	Private, Public, or Role based	Helps in creating access control and data inventory.
Environment	Non-Production (Development, Testing) or Production	Helps in sharing the bucket across environments, minimizing data duplication, and helps in complete testing.
Privileged	Yes or No	Helps in finer access control without the need to replicate data in separate buckets.
Retention	Time frame	Helps lifecycle policy to move data across different storage classes (S3 Standard-IA, S3 One Zone-IA, and S3 Glacier) for cost optimization.

AWS Lake Formation: Centralized governance and access control

Using AWS Lake Formation, you can centrally govern your data lake and control access to your data by defining granular data access policies to the metadata and data through grant or revoke permissions model. Lake Formation integrates with IAM to define access control across the following two areas:

- **Metadata access control** – Permissions for Data Catalog resources. These permissions enable principals to create, read, update, and delete metadata databases and tables in the Data Catalog.

- **Underlying data access control** – Permissions on locations in Amazon S3. Data access permissions enable principals to read and write data to underlying S3 locations. Data location permissions enable principals to create metadata databases and tables that point to specific S3 locations.

Lake Formation uses the same metadata catalog used by AWS Glue and you can use AWS Glue crawlers to create data catalogs for the data stored in your data lakes. Because Lake Formation combines with IAM permissions, you should have the necessary IAM permissions and Lake Formation permissions to use Lake Formation and the integrated services (AWS Glue for Data Catalog and crawlers, Amazon Athena, Amazon Redshift, and Amazon EMR).

Lake Formation also controls access of integrated services to data stored in data lakes using fine-grained policies. You can define named resources access control to grant permissions to specific databases or tables by specifying database or table names. The recommended method to grant permissions is tag-based access control (TBAC) because it simplifies access control while managing a large number of Data Catalog resources and principals.

TBAC allows you to grant permissions based on one or more Lake Formation tags (LF-tags). You can create LF-tags; assign LF-tags to Data Catalog resources such as databases, tables, and columns; and grant LF-tags permissions to principals optionally with grant option. The tag-based access control method is recommended when you want to manage a large number of Data Catalog resources.

Monitoring and optimizing the data lake environment

Beyond the efforts required to architect and build a data lake, your organization must also consider the operational aspects of a data lake, and how to cost-effectively and efficiently operate a production data lake at large scale. Key elements you must consider are monitoring the operations of the data lake, making sure that it meets performance expectations and service-level agreements, analyzing utilization patterns, and using this information to optimize the cost and performance of your data lake. AWS provides multiple features and services to help optimize a data lake that is built on AWS, including S3 storage analytics, Amazon CloudWatch metrics, AWS CloudTrail, and Amazon Glacier.

Data lake monitoring

A key aspect of operating a data lake environment is understanding how all of the components that comprise the data lake are operating and performing, and generating notifications when issues occur or operational performance falls below predefined thresholds.

Amazon CloudWatch

As an administrator, you need to look at the complete data lake environment holistically. This can be achieved using Amazon CloudWatch. CloudWatch is a monitoring service for AWS Cloud resources and the applications that run on AWS. You can use CloudWatch to collect and track metrics, collect and monitor log files, set thresholds, and initiate alarms. This allows you to automatically react to changes in your data lake built on S3.

You can also use CloudWatch metrics to understand and improve the performance of applications that are using Amazon S3. You can use CloudWatch for generating daily storage metrics for your data lake built on CloudWatch by collecting and processing storage data for your S3 buckets. You can also monitor the requests to your data lake built on S3 and identify and act upon operational issues quickly. In addition, you can also monitor the S3 APIs that are pending replication, total size, and the maximum time required for replication to the destination Region.

Amazon Macie

Amazon Macie is a fully managed data security and data privacy service that uses machine learning and pattern matching to discover, monitor, and protect your sensitive data stored in your data lake.

Macie can be used to scan your data lakes and discover sensitive information such as PII or financial data, and identify and report overly permissive or unencrypted buckets.

Using Macie, you can run a sensitive data discovery job to identify sensitive information using built-in criteria and techniques, such as machine learning and pattern matching called as managed data identifiers, to analyze objects in your data lake. You can also define your own criteria called as custom data identifiers, using regular expressions defining a text pattern to match and, optionally, character sequences and a proximity rule that refines the results.

Additionally, after enabled, Macie maintains a complete inventory of your S3 buckets and evaluates and monitors the bucket for security and access control. If Macie detects a potential issue, the service creates a policy finding. Policy findings are generated when policies or settings for S3 bucket are changed in a way that reduces the security of the bucket and its objects. Macie integrates with other AWS services, such as Amazon EventBridge, which is a serverless event bus service that can send findings data to services, such as AWS Lambda and Amazon Simple Notification Service (Amazon SNS), to act on the policy findings.

AWS CloudTrail

An operational data lake has many users and multiple administrators, and may be subject to compliance and audit requirements, so it's important to have a complete audit trail of actions taken and who has performed these actions. AWS CloudTrail is an AWS service that enables governance, compliance, operational auditing, and risk auditing of AWS accounts.

CloudTrail continuously monitors and retains events related to API calls across AWS services that comprise a data lake. CloudTrail provides a history of AWS API calls for an account, including API calls made through the AWS Management Console, AWS SDKs, command line tools, and most data lakes built on S3 services. You can identify which users and accounts made requests or took actions against AWS services that support CloudTrail, the source IP address the actions were made from, and when the actions occurred.

CloudTrail can be used to simplify data lake compliance audits by automatically recording and storing activity logs for actions made within AWS accounts.

Integration with Amazon CloudWatch logs provides a convenient way to search through log data, identify out-of-compliance events, accelerate incident investigations, and expedite responses to auditor requests. CloudTrail logs are stored in a separate logs bucket within your data lake (refer to the **Data lake foundation** section of this document) for durability and deeper analysis.

Data lake optimization

Optimizing a data lake environment includes minimizing operational costs. By building a data lake on S3, you only pay for the data storage and data processing services that you actually use, as you use them. You can reduce costs by optimizing how you use these services. Data asset storage is often a significant portion of the costs associated with a data lake. Fortunately, AWS has several features that can be used to optimize and reduce costs. These include: S3 Lifecycle management, S3 storage class analysis, S3 Intelligent-Tiering, S3 Storage Lens, and Amazon Glacier storage class.

Amazon S3 Lifecycle management

Amazon S3 Lifecycle management allows you to create lifecycle rules, which can be used to automatically migrate data assets to a lower cost tier of storage—such as S3 Standard-IA storage class or Amazon Glacier storage class—or let them expire when they are no longer needed. A lifecycle configuration, which consists of an XML file, comprises a set of rules with predefined actions that you want Amazon S3 to perform on data assets during their lifetime. Lifecycle configurations can perform actions based on data asset age and data asset names, but can also be combined with S3 object tagging to perform very granular management of data assets. If the access pattern for your S3 buckets is constantly changing or is evolving, you can use S3 Intelligent-Tiering storage class for automatic cost savings.

Amazon S3 Storage class analysis

One of the challenges of developing and configuring lifecycle rules for the data lake is gaining an understanding of how data assets are accessed over time. It only makes economic sense to transition data assets to a more cost-effective storage or archive tier if those objects are infrequently accessed. Otherwise, data access charges associated with these more cost-effective storage classes can negate any potential savings. Amazon S3 provides storage class analysis to help you understand how data lake data assets are used. Storage class analysis uses machine learning algorithms on collected access data to help you develop lifecycle rules that will optimize costs.

Seamlessly tiering to lower cost storage tiers is an important capability of a data lake, particularly as its users plan for, and move to, more advanced analytics and machine learning capabilities. Data lake users will typically ingest raw data assets from many sources,and transform those assets into harmonized formats that they can use for one-time querying and on-going business intelligence querying through SQL. However, users also want to perform more advanced analytics using streaming analytics, machine learning,and artificial intelligence. These more advanced analytics

capabilities consist of building data models, validating these data models with data assets, and then training and refining these models with historical data.

Keeping more historical data assets, particularly raw data assets, allows for better training and refinement of models. Additionally, as your organization's analytics sophistication grows, you may want to go back and reprocess historical data to look for new insights and value. These historical data assets are infrequently accessed and consume a lot of capacity, so they are often well suited to be stored on an archival storagelayer.

Another long-term data storage need for the data lake is to keep processed data assets and results for long-term retention for compliance and audit purposes, to be accessed by auditors when needed. Both of these use cases are well served by Amazon Glacier storage class, which is an S3 storage class optimized for infrequently used cold data, and for storing write once read many (WORM) data. You can also use S3 Object Lock to adhere to regulatory compliance.

S3 Intelligent-Tiering

S3 Intelligent-Tiering is designed to optimize your storage cost by automatically moving the data within your buckets by monitoring your access patterns. S3 Intelligent-Tiers makes use of two low latency high throughout access tiers: one tier for frequently accessed data and another tier for infrequently accessed data. S3 Intelligent Tiering monitors access patterns for the data for a period of 30 months and automatically moves the data to one of the access tiers without operational overhead or performance impact.

S3 Intelligent-Tiering is recommended for data that has unpredictable access patterns regardless of object type, size and retention period such as data lakes, data analytics applications or new applications. If the data from infrequently accessed tier is accessed, the data will be moved to frequently accessed tier. Additionally, you can also configure S3 Intelligent-Tiering to automatically move data from infrequently accessed tier that has not been accessed for consecutive 90 days to archive access tier and if not accessed for consecutive 180 days to deep archive access tier.

You can also configure the last access time for archiving up to a maximum of 730 days for both Archive access tier and Deep Archive access tier. S3 Intelligent-Tiering Archive access tier provides same performance as Amazon Glacier storage class and S3 Intelligent Tiering Deep Archive access tier provides same performance as Amazon Glacier Deep Archive storage class.

S3 Storage Lens

As your data lake becomes more popular and you start expanding to accommodate data from multiple applications across multiple accounts and S3 buckets. It can become increasingly

complicated to understand usage of the data across the organization, optimize cost and understand the security posture. S3 Storage Lens gives you the visibility into your object storage across your organization with point-in-time metrics, trend lines and actionable insights.

You can generate insights at organization, account, region, bucket, and prefix level. S3 aggregates your usage and metrics across all the accounts and provides an account snapshot on the S3 console (Bucket) home page. You can use the Storage Lens dashboard to visualize insights and trends, identify outliers, receive recommendations for storage cost optimization, and so on.

You can use Storage Lens dashboard to identify your largest buckets and take necessary actions to optimize the cost since the rate charged depends on the object size, duration of storage and storage class. In case you are uploading objects using multi-part upload, there might be a case when the uploads fail or not completed. The incomplete uploads remain in your buckets and are chargeable.

You can identify these incomplete multipart uploads using Storage Lens dashboard. Additionally, storage lens can also help identify multiple versions of the objects. Finally, you can also use Storage Lens to uncover cold buckets from your account. Cold buckets are the buckets which are not accessed for a long period of time. All these insights can be accessed from the Storage Lens dashboard.

Amazon Glacier

Amazon Glacier is a low-cost Amazon S3 storage class that provides durable storage with security features for data archiving and backup. Amazon Glacier has the same data durability (99.999999999%) and supports lifecycle management on data assets stored in S3, so that data assets can seamlessly migrate from Amazon S3 to Amazon Glacier. Amazon Glacier storage class is a great storage choice when low storage cost is essential, data assets are rarely retrieved, and retrieval latency of several minutes to several hours is acceptable.

Different types of data lake assets may have different retrieval needs. For example, compliance data may be infrequently accessed and be relatively small in size but needs to be made available in minutes when auditors request data, whereas historical raw data assets may be very large but can be retrieved in bulk over the course of a day when needed.

Amazon Glacier allows data lake users to specify retrieval times when the data retrieval request is created, with longer retrieval times leading to lower retrieval costs. For processed data and records that need to be securely retained, Amazon Glacier Vault Lock allows data lake administrators to deploy and enforce compliance controls on individual Amazon Glacier vaults by a lockable policy.

Administrators can specify controls such as WORM in a Vault Lock policy and lock the policy from future edits. After locked, the policy becomes immutable and Amazon Glacier will enforce the prescribed controls to help achieve your compliance objectives, and provide an audit trail for these assets using AWS CloudTrail.

Cost and performance optimization

You can optimize your data lake using cost and performance. Amazon S3 provides a very performant foundation for a data lake because its enormous scale provides virtually limitless throughput and extremely high transaction rates. Using S3 best practices for data asset naming ensures high levels of performance. These best practices can be found in the Amazon Simple Storage Service Developer Guide.

Another area of optimization is to use optimal data formats when transforming raw data assets into normalized formats, in preparation for querying and analytics. These optimal data formats can compress data and reduce data capacities needed for storage, and also substantially increase query performance by common data lake built on S3 analytic services.

Data lake environments are designed to ingest and process many types of data, and store raw data assets for future archival and reprocessing purposes, as well as store processed and normalized data assets for active querying, analytics, and reporting. A key best practice to reduce storage and analytics processing costs, and improve analytics querying performance, is to use an optimized data format, particularly a format like Apache Parquet.

Parquet is a columnar compressed storage file format that is designed for querying large amounts of data, regardless of the data processing framework, data model, or programming language. Compared to common raw data log formats such as CSV, JSON, or TXT format, Parquet can reduce the required storage footprint, improve query performance significantly, and greatly reduce querying costs for AWS services, which charge by amount of data scanned.

Amazon tests comparing CSV and Parquet formats using one TB of log data stored in CSV format to Parquet format showed the following:

- Space savings of 87% with Parquet (1 TB of log data stored in CSV format compressed to 130 GB with Parquet)
- A query time for a representative Athena query was 34x faster with Parquet (237 seconds for CSV versus 5.13 seconds for Parquet), and the amount of data scanned for that Athena query was 99% less (1.15TB scanned for CSV versus 2.69GB for Parquet)
- The cost to run that Athena query was 99.7% less ($5.75 for CSV versus $0.013 for Parquet)

Parquet has the additional benefit of being an open data format that can be used by multiple querying and analytics tools in a data lake built on Amazon S3, particularly Amazon Athena, Amazon EMR, Amazon Redshift, and Amazon Redshift Spectrum.

Additional options for performance optimization include rightsizing of the S3 objects to 128 MB, partitioning based on business dates which are typically used while querying.

Future proofing the data lake

A data lake built on AWS can immediately solve a broad range of business analytics challenges and quickly provide value to your business. However, business needs are constantly evolving, AWS and the analytics partner ecosystem are rapidly evolving and adding new services and capabilities, as businesses and their data lake users achieve more experience and analytics sophistication over time. Therefore, it's important that the data lake can seamlessly and non-disruptively evolve as needed.

AWS future proofs your data lake with a standardized storage solution that grows with your organization by ingesting and storing all of your business's data assets on a platform with virtually unlimited scalability and well-defined APIs, and integrates with a wide variety of data processing tools. This allows you to add new capabilities to your data lake as you need them without infrastructure limitations or barriers. Additionally, you can perform agile analytics experiments against data lake assets to quickly explore new processing methods and tools, and then scale the promising ones into production without the need to build new infrastructure, duplicate and/or migrate data, and have users migrate to a new platform.

Conclusion and contributors

Conclusion

Amazon S3 is a scalable, highly durable, and reliable service to build and manage a secure data lake at scale. You can ingest and store structured, semi-structured and unstructured data from wide variety of sources into a centralized platform. With a data lake built on S3, you can use native AWS services to run big data analytics, artificial intelligence (AI) and machine learning (ML) applications to gain insights from your unstructured data sets. You also have the flexibility to use your preferred analytics, AI and ML solutions from the Amazon Partner Network (APN). S3 provides a wide variety of service features to empower IT managers, storage administrators, and data scientists to enforce access policies, manage objects at scale, audit activities and secure data across their data lake built on S3.

In closing, a data lake built on AWS allows you to evolve your business around your data assets, and to use these data assets to quickly and agilely drive more business value and competitive differentiation without limits.

Contributors

The following individuals and organizations contributed to this document:

- John Mallory, Business Development Manager, AWS Storage
- Robbie Wright, Product Marketing Manager, AWS Storage
- Sarang Kamble, Senior Global Solutions Architect

Resources

- AWS Architecture Center
- AWS Whitepapers & Guides
- AWS Documentation

Document history

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Minor update	Fix non-inclusive language.	April 6, 2022
Updated	Updated for technical accuracy.	November 16, 2021
Initial publication	Whitepaper first published.	July 1, 2017

Note

To subscribe to RSS updates, you must have an RSS plug-in enabled for the browser you are using.