Building Big Data Storage Solutions (Data Lakes) for Maximum Flexibility

July 2017

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Introduction 1

Amazon S3 as the Data Lake Storage Platform 2

Data Ingestion Methods 3

- Amazon Kinesis Firehose 4
- AWS Snowball 5
- AWS Storage Gateway 5

Data Cataloging 6

- Comprehensive Data Catalog 6
- HCatalog with AWS Glue 7

Securing, Protecting, and Managing Data 8

- Access Policy Options and AWS IAM 9
- Data Encryption with Amazon S3 and AWS KMS 10
- Protecting Data with Amazon S3 11
- Managing Data with Object Tagging 12

Monitoring and Optimizing the Data Lake Environment 13

- Data Lake Monitoring 13
- Data Lake Optimization 15

Transforming Data Assets 18

In-Place Querying 19

- Amazon Athena 20
- Amazon Redshift Spectrum 20

The Broader Analytics Portfolio 21

- Amazon EMR 21
- Amazon Machine Learning 22
- Amazon QuickSight 22
- Amazon Rekognition 23

Future Proofing the Data Lake 23
Contributors 24
Document Revisions 24

Abstract

Organizations are collecting and analyzing increasing amounts of data making it difficult for traditional on-premises solutions for data storage, data management, and analytics to keep pace. Amazon S3 and Amazon Glacier provide an ideal storage solution for data lakes. They provide options such as a breadth and depth of integration with traditional big data analytics tools as well as innovative query-in-place analytics tools that help you eliminate costly and complex extract, transform, and load processes. This guide explains each of these options and provides best practices for building your Amazon S3-based data lake.

Introduction

As 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 siloes that aren't built to work well together make storage consolidation for more comprehensive and efficient analytics difficult. This, in turn, limits an organization's agility, ability to derive more insights and value from its data, and capability to seamlessly adopt more sophisticated analytics tools and processes as its skills and needs evolve.

A *data lake*, which is a single platform combining storage, data governance, and analytics, is designed to address these challenges. It's a centralized, secure, and durable cloud-based storage platform that allows you to ingest and store structured and unstructured data, and transform these raw data assets as needed. You don't need an innovation-limiting pre-defined schema. You can use a complete portfolio of data exploration, reporting, analytics, machine learning, 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, allowing them to get all of the business insights they need, whenever they need them.

Until recently, the data lake had been more concept than reality. However, Amazon Web Services (AWS) has developed a data lake architecture that allows you to build data lake solutions cost-effectively using Amazon Simple Storage Service (Amazon S3) and other services.

Using the Amazon S3-based data lake architecture capabilities you can do the following:

- 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.

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.

Easily and securely share processed datasets and results.

The remainder of this paper provides more information about each of these capabilities. Figure 1 illustrates a sample AWS data lake platform.

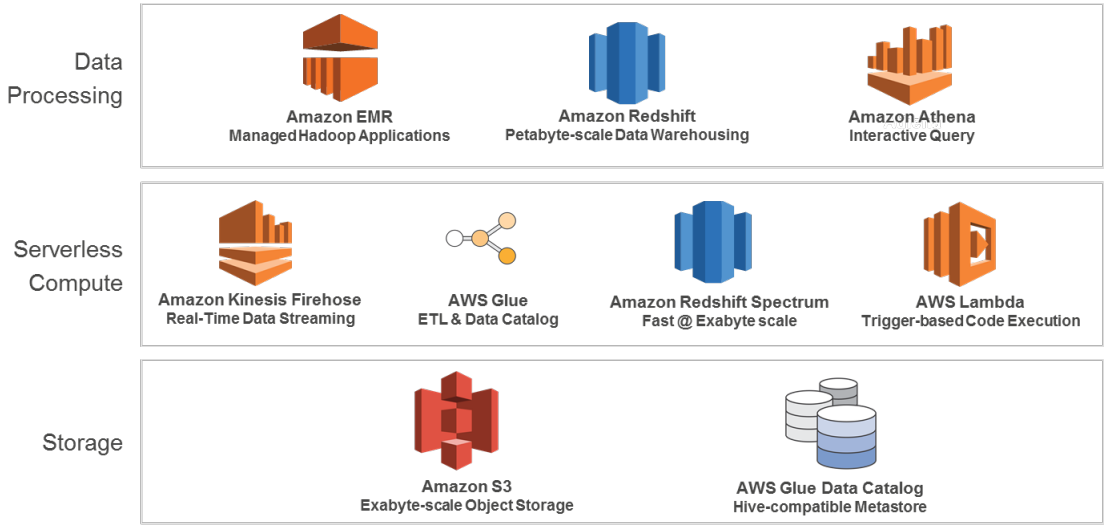

Figure 1: Sample AWS data lake platform

Amazon S3 as the Data Lake Storage Platform

The Amazon S3-based data lake solution uses Amazon S3 as its primary storage platform. Amazon S3 provides an optimal foundation for a data lake because of its virtually unlimited scalability. You can seamlessly and nondisruptively 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, and native encryption and access control capabilities. Amazon S3 integrates with a broad portfolio of AWS and third-party ISV data processing tools.

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 (EC2), and you can use AWS analytics tools to process your data. You can optimize your EC2 instances to provide the right ratios of CPU, memory, and bandwidth for best performance.

Centralized data architecture. Amazon S3 makes it easy to build a multi-tenant environment, where many users can bring their own data analytics tools to a common set of 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.

Integration with clusterless and serverless AWS services. Use Amazon S3 with Amazon Athena, Amazon Redshift Spectrum, Amazon Rekognition, 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. With all of these capabilities, you only pay for the actual amounts of data you process or for the compute time that you consume.

Standardized APIs. Amazon S3 RESTful 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.

Data Ingestion Methods

One of the core capabilities of a data lake architecture is the ability to quickly and easily ingest multiple types of data, such as real-time streaming data and bulk data assets from on-premises storage platforms, as well as data generated and processed by legacy on-premises platforms, such as mainframes and data warehouses. AWS provides services and capabilities to cover all of these scenarios.

Amazon Kinesis Firehose

Amazon Kinesis Firehose is a fully managed service for delivering real-time streaming data directly to Amazon S3. Kinesis Firehose automatically scales to match the volume and throughput of streaming data, and requires no ongoing administration. Kinesis Firehose can also be configured to transform streaming data before it's stored in Amazon S3. Its transformation capabilities include compression, encryption, data batching, and Lambda functions.

Kinesis 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. Kinesis Firehose encryption supports Amazon S3 server-side encryption with AWS Key Management Service (AWS KMS) for encrypting delivered data in Amazon S3. You can choose not to encrypt the data or to encrypt with a key from the list of AWS KMS keys that you own (see the section Encryption with AWS KMS). Kinesis 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 Amazon S3 transaction costs and transactions per second load.

Finally, Kinesis Firehose can invoke Lambda functions to transform incoming source data and deliver it to Amazon S3. Common transformation functions include transforming Apache Log and Syslog formats to standardized JSON and/or CSV formats. The JSON and CSV formats can then be directly queried using Amazon Athena. If using a Lambda data transformation, you can optionally back up raw source data to another S3 bucket, as Figure 2 illustrates.

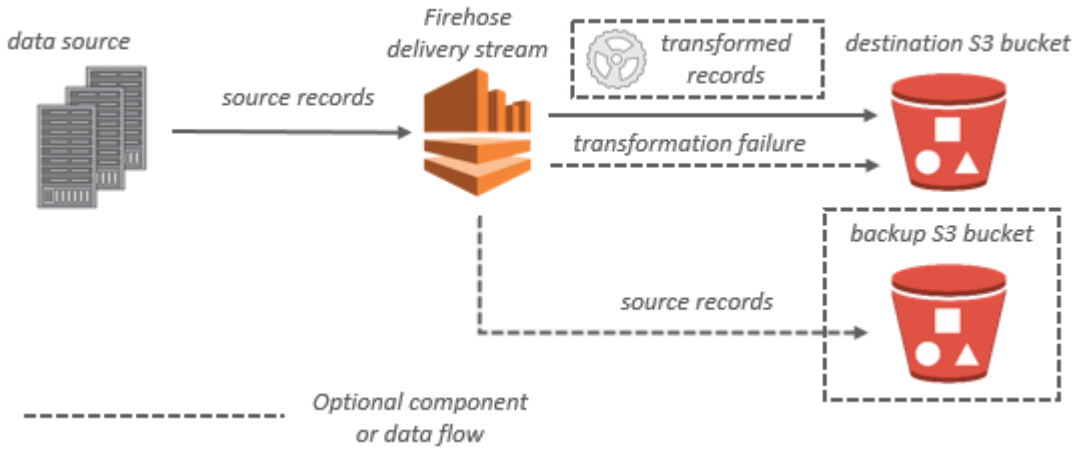

Figure 2: Delivering real-time streaming data with Amazon Kinesis Firehose to Amazon S3 with optional backup

AWS Snowball

You can use AWS Snowball to securely and efficiently migrate bulk data from on-premises storage platforms and Hadoop clusters to S3 buckets. After you create a job in the AWS Management Console, a Snowball appliance will be automatically shipped to you. After a Snowball arrives, connect it to your local network, install the Snowball client on your on-premises data source, and then use the Snowball client to select and transfer the file directories to the Snowball device. The Snowball client uses AES-256-bit encryption. Encryption keys are never shipped with the Snowball device, so the data transfer process is highly secure. After the data transfer is complete, the Snowball's E Ink shipping label will automatically update. Ship the device back to AWS. Upon receipt at AWS, your data is then transferred from the Snowball device to your S3 bucket and stored as S3 objects in their original/native format. Snowball also has an HDFS client, so data may be migrated directly from Hadoop clusters into an S3 bucket in its native format.

AWS Storage Gateway

AWS Storage Gateway can be used to integrate legacy on-premises data processing platforms with an Amazon S3-based data lake. The File Gateway configuration of Storage Gateway offers on-premises devices and applications a network file share via 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 easily 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 tools such as Amazon EMR or Amazon Athena to process this data.

Additionally, Amazon S3 natively supports 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 typically looks like the following:

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

Data Cataloging

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 an Amazon S3-based data lake is the data catalog. The data catalog provides a query-able interface of all assets stored in the data lake's S3 buckets. The data catalog is designed to provide a single source of truth about the contents of the data lake.

There are two general forms of a data catalog: a comprehensive data catalog that contains information about all assets that have been ingested into the S3 data lake, and a Hive Metastore Catalog (HCatalog) that contains information about data assets that have been transformed into formats and table definitions that are usable by analytics tools like Amazon Athena, Amazon Redshift, Amazon Redshift Spectrum, and Amazon EMR. The two catalogs are not mutually exclusive and both may exist. The comprehensive data catalog can be used to search for all assets in the data lake, and the HCatalog can be used to discover and query data assets in the data lake.

Comprehensive Data Catalog

The comprehensive data catalog can be created by using standard AWS services like AWS Lambda, Amazon DynamoDB, and Amazon Elasticsearch Service (Amazon ES). At a high level, Lambda triggers are used to populate DynamoDB

tables with object names and metadata when those objects are put into Amazon S3; then Amazon ES is used to search for specific assets, related metadata, and data classifications. Figure 3 shows a high-level architectural overview of this solution.

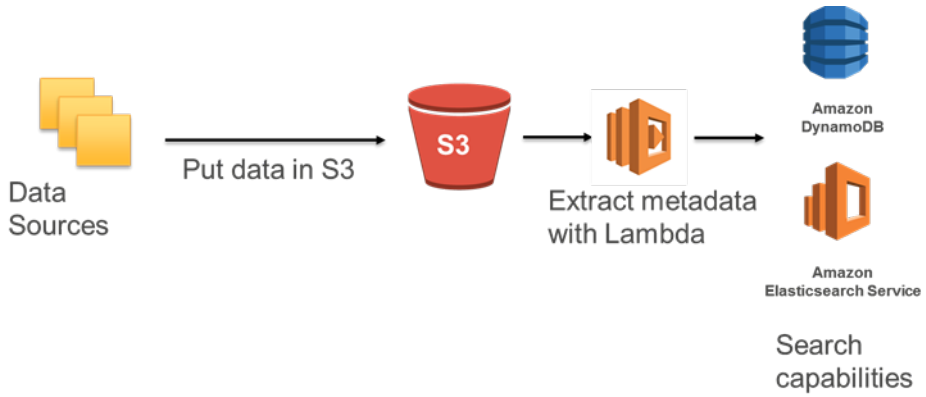

Figure 3: Comprehensive data catalog using AWS Lambda, Amazon DynamoDB, and Amazon Elasticsearch Service

HCatalog with AWS Glue

AWS Glue (now in beta) can be used to create a Hive-compatible Metastore Catalog of data stored in an Amazon S3-based data lake. To use AWS Glue to build your data catalog, register your data sources with AWS Glue in the AWS Management Console. AWS Glue will then crawl your S3 buckets for data sources and construct a data catalog using pre-built classifiers for many popular source formats and data types, including JSON, CSV, Parquet, and more. You may also add your own classifiers or choose classifiers from the AWS Glue community to add to your crawls to recognize and catalog other data formats. The AWS Glue-generated catalog can be used by Amazon Athena, Amazon Redshift, Amazon Redshift Spectrum, and Amazon EMR, as well as third-party analytics tools that use a standard Hive Metastore Catalog. Figure 4 shows a sample screenshot of the AWS Glue data catalog interface.

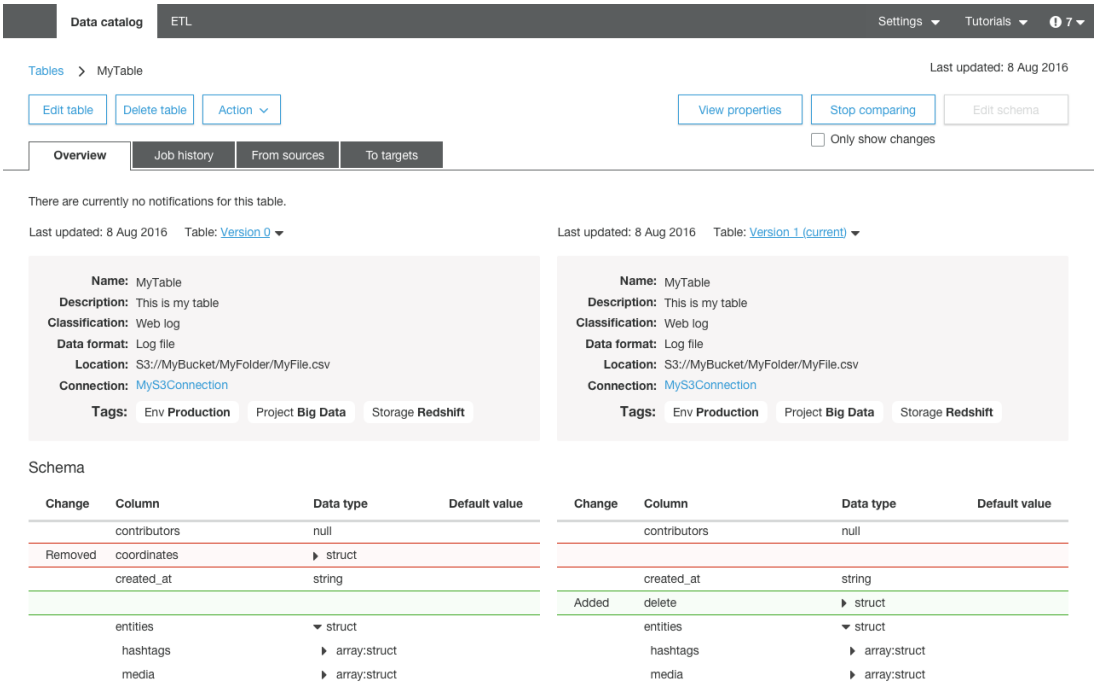

Figure 4: Sample AWS Glue data catalog interface

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 Amazon 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 Amazon S3 data management features enable automation and scaling of data lake storage management, even when it contains billions of objects and petabytes of data assets.

Securing your data lake begins with implementing very fine-grained controls that allow authorized users to see, 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 to secure an Amazon S3-based data lake.

Access Policy Options and AWS IAM

You can manage access to your Amazon S3 resources using access policy options. By default, all Amazon S3 resources—buckets, objects, and related subresources—are private: only the resource owner, an AWS account that created them, can access the resources. The resource owner can then grant access permissions to others by writing an access policy. Amazon 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*. Example resource-based policies include bucket policies and access control lists (ACLs). 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.

For most data lake environments, we recommend using user policies, so that permissions to access data assets can also be tied to user roles and permissions for the data processing and analytics services and tools that your data lake users will use. User policies are associated with AWS Identity and Access Management (IAM) service, which allows you to securely control access to AWS services and resources. With IAM, you can create IAM users, groups, and roles in accounts and then attach access policies to them that grant access to AWS resources, including Amazon S3. The model for user policies is shown in Figure 5. For more details and information on securing Amazon S3 with user policies and AWS IAM, please reference: [Amazon Simple Storage Service Developers Guide] and [AWS Identity and Access Management User Guide].

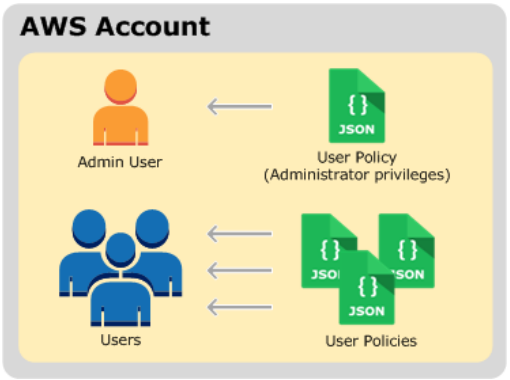

Figure 5: Model for user policies

Data Encryption with Amazon S3 and AWS KMS

Although user policies and IAM control who can see and access data in your Amazon S3-based data lake, it's also important to ensure that users who might inadvertently or maliciously manage to gain access to those data assets can't see and use them. This is accomplished by using encryption keys to encrypt and de-encrypt data assets. Amazon S3 supports multiple encryption options. Additionally, 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 (SSE) and client-side encryption. SSE provides data-at-rest encryption for data written to Amazon S3. With SSE, Amazon 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 written into Amazon S3. For example, a data lake user could specify client-side encryption before transferring data assets into Amazon S3 from the Internet, or could specify that services like Amazon EMR, Amazon Athena, or Amazon Redshift use client-side encryption with Amazon S3. SSE 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 like a data lake, we strongly recommend using AWS KMS to coordinate keys across client- and server-side encryption and across multiple data processing and analytics services.

For even greater levels of data lake data protection, other services like 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 data assets. This architecture has been created for the Amazon S3-based data lake solution reference architecture, which can be found, downloaded, and deployed at

https://aws.amazon.com/answers/big-data/data-lake-solution/

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 16 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.

Beyond core data protection, another key element is to protect data assets 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 data. This becomes especially important in a large multi-tenant data lake, which will have a large number of users, many applications, and constant ad hoc data processing and application development. Amazon S3 provides versioning to protect data assets against these scenarios. When enabled, Amazon S3 versioning will keep 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, and can be combined with other Amazon S3 capabilities such as lifecycle management for long-term

retention of versions on lower cost storage tiers such as Amazon Glacier, and Multi-Factor-Authentication (MFA) Delete, which requires a second layer of authentication—typically via an approved external authentication device—to delete data asset versions.

Even though Amazon 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 (DR) architectures in a second location. Amazon S3 cross-region replication (CRR) 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.

All of these Amazon S3 features and capabilities—when combined with other AWS services like IAM, AWS KMS, Amazon Cognito, and Amazon API Gateway—ensure that a data lake using Amazon S3 as its core storage platform will be able to 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 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. Amazon 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 an example of object tagging, 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-grain 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 Amazon S3 data lifecycle policies, which is discussed in the next section of this whitepaper. 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 next section of this paper—to display monitoring and action audit data by specific data asset tag filters.

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 SLAs, 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 Amazon 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 trigger alarms. This allows you to automatically react to changes in your AWS resources.

CloudWatch can monitor AWS resources such as Amazon EC2 instances, Amazon S3, Amazon EMR, Amazon Redshift, Amazon DynamoDB, and Amazon Relational Database Service (RDS) database instances, as well as custom metrics generated by other data lake applications and services. CloudWatch provides system-wide visibility into resource utilization, application performance, and operational health. You can use these insights to proactively react to issues and keep your data lake applications and workflows running smoothly.

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 the 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 Amazon S3-based data lake 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 an S3 bucket for durability and deeper analysis.

Data Lake Optimization

Optimizing a data lake environment includes minimizing operational costs. By building a data lake on Amazon 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, and Amazon Glacier.

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-Infrequent Access or Amazon Glacier—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.

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 could negate any potential savings. Amazon S3 provides S3 storage class analysis to help you understand how data lake data assets are used. Amazon S3 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 in an important capability for 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 ad hoc querying and on-going business intelligence (BI) querying via SQL. However, they will 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 storage layer.

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, which is an AWS storage service optimized for infrequently used cold data, and for storing write once, read many (WORM) data.

Amazon Glacier

Amazon Glacier is an extremely low-cost storage service that provides durable storage with security features for data archiving and backup Amazon Glacier has the same data durability (99.999999999%) as Amazon S3, the same integration with AWS security features, and can be integrated with S3 by using S3 lifecycle management on data assets stored in S3, so that data assets can be seamlessly migrated from S3 to Glacier. Amazon Glacier is a great storage choice when low storage cost is paramount, 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 relatively small in size but needs to be made available in minutes when auditors request data, while 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 easily deploy and enforce compliance controls on individual Glacier vaults via a lockable policy. Administrators can specify controls such as Write Once Read Many (WORM) in

a Vault Lock policy and lock the policy from future edits. Once 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 the data lake because its enormous scale provides virtually limitless throughput and extremely high transaction rates. Using Amazon S3 best practices for data asset naming ensures high levels of performance. These best practices can be found in the Amazon Simple Storage Service Developers 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 Amazon S3-based data lake 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. One of the key best practices to reduce storage and analytics processing costs, as well as 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 like 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 the CSV and Parquet formats using 1 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 an Amazon S3-based data lake, particularly Amazon Athena, Amazon EMR, Amazon Redshift, and Amazon Redshift Spectrum.

Transforming Data Assets

One of the core values of a data lake is that it is the collection point and repository for all of an organization's data assets, in whatever their native formats are. This enables 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 ad hoc SQL querying. As discussed earlier, when a data lake is built on AWS, we recommend transforming log-based data assets into Parquet format. AWS provides multiple services to quickly and efficiently achieve this.

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

The two most common and straightforward methods to transform data assets into Parquet in an Amazon S3-based data lake use Amazon EMR clusters. The first method involves creating an 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, related 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, simpler data transformation method on an Amazon S3-based data lake is to use AWS Glue. AWS Glue is an AWS fully managed extract, transform, and load (ETL) service that can be directly used with data stored in Amazon S3. 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 (IDE), 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 execution of data lake jobs in the right sequence, and automatically re-tries 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.

In-Place Querying

One of the most important capabilities of a data lake that is built on AWS is the ability to do in-place transformation and querying of data assets without having to provision and manage clusters. This allows you to run sophisticated analytic queries directly on your data assets stored in Amazon S3, without having to copy and load data into separate analytics platforms or data warehouses. You

can query S3 data without any additional infrastructure, and you only pay for the queries that you 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 provide the in-place querying capabilities.

Amazon Athena

Amazon Athena is an interactive query service that makes it easy for you to analyze data directly in Amazon 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 and begin using standard SQL to run ad hoc 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—executing 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. Athena integrates with Amazon QuickSight for easy visualization. It can also be used with third-party reporting and business intelligence tools by connecting these tools to Athena with a JDBC driver.

Amazon Redshift Spectrum

A second way to perform in-place querying of data assets in an Amazon S3-based data lake is to use Amazon Redshift Spectrum. Amazon Redshift is a large-scale, managed data warehouse service that can be used with data assets in Amazon S3. However, data assets must be loaded into Amazon Redshift before queries can be run. By contrast, Amazon Redshift Spectrum enables you to run Amazon Redshift SQL queries directly against massive amounts of data—up to exabytes—stored in an Amazon S3-based data lake. 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. Redshift Spectrum can directly query a wide variety of data assets stored in the data lake, including CSV, TSV, Parquet, Sequence, and RCFile. Since Redshift Spectrum supports the SQL syntax of Amazon Redshift, you can run sophisticated queries using the same BI tools that you use today. You also have the flexibility to run queries that span both frequently accessed data assets that are stored locally in Amazon Redshift and your full data sets stored in Amazon S3. Because Amazon Athena and Amazon Redshift share a common data catalog and common data formats, you can use both Athena and Redshift Spectrum against the same data assets. You would typically use Athena for ad hoc data discovery and SQL querying, and then use Redshift Spectrum for more complex queries and scenarios where a large number of data lake users want to run concurrent BI and reporting workloads.

The Broader Analytics Portfolio

The power of a data lake built on AWS is that data assets get ingested and stored in one massively scalable, low cost, performant platform—and that data discovery, transformation, and SQL querying can all be done in place using innovative AWS services like 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 Amazon S3 to create any number of sophisticated analytics, machine learning, and artificial intelligence (AI) 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 an Amazon S3-based data lake are described next.

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, and HBase. 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 Amazon S3. With Amazon EMR, you can launch a persistent cluster that stays

up indefinitely or a temporary cluster that terminates 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 and storage I/O optimized (e.g., T2, C4, X1, and I3) instances, and all Amazon EC2 pricing options (On-Demand, Reserved, and Spot). When you launch an EMR cluster (also called a *job flow*), you choose how many and what type of EC2 instances to provision. Companies with many different lines of business and a large number of users can build a single data lake solution, store their data assets in Amazon S3, and then spin up multiple EMR clusters to share data assets in a multi-tenant fashion.

Amazon Machine Learning

Machine learning is another important data lake use case. Amazon Machine Learning (ML) is a data lake service that makes it easy for anyone to use predictive analytics and machine learning technology. Amazon ML provides visualization tools and wizards to guide you through the process of creating ML models without having to learn complex algorithms and technology. After the models are ready, Amazon ML makes it easy to obtain predictions for your application using API operations. You don't have to implement custom prediction generation code or manage any infrastructure. Amazon ML can create ML models based on data stored in Amazon S3, Amazon Redshift, or Amazon RDS. Built-in wizards guide you through the steps of interactively exploring your data, training the ML model, evaluating the model quality, and adjusting outputs to align with business goals. After a model is ready, you can request predictions either in batches or by using the low-latency real-time API. As discussed earlier in this paper, a data lake built on AWS greatly enhances machine learning capabilities by combining Amazon ML with large historical data sets than can be cost effectively stored on Amazon Glacier, but can be easily recalled when needed to train new ML models.

Amazon QuickSight

Amazon QuickSight is a very fast, easy-to-use, business analytics service that makes it easy for you to build visualizations, perform ad hoc analysis, and quickly get business insights from your data assets stored in the data lake, anytime, on any device. You can use Amazon QuickSight to seamlessly discover AWS data sources such as Amazon Redshift, Amazon RDS, Amazon Aurora, Amazon Athena, and Amazon S3, connect to any or all of these data sources and

data assets, and get insights from this data in minutes. Amazon QuickSight enables 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 robust in-memory engine (SPICE).

Amazon Rekognition

Another innovative data lake service is Amazon Rekognition, which is a fully managed image recognition service powered by deep learning, run against image data assets stored in Amazon S3. Amazon Rekognition has been built by Amazon's Computer Vision teams over many years, and already analyzes billions of images every day. The Amazon Rekognition easy-to-use API detects thousands of objects and scenes, analyzes faces, compares two faces to measure similarity, and verifies faces in a collection of faces. With Amazon Rekognition, you can easily build applications that search based on visual content in images, analyze face attributes to identify demographics, implement secure face-based verification, and more. Amazon Rekognition is built to analyze images at scale and integrates seamlessly with data assets stored in Amazon S3, as well as AWS Lambda and other key AWS services.

These are just a few examples of powerful data processing and analytics tools that can be integrated with a data lake built on AWS. See the AWS website for more examples and for the latest list of innovative AWS services available for data lake users.

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 futureproofs 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. 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

Document Revisions

Date	Description
July 2017	First publication