Contents

Athena

Athena workgroups: We recommend using workgroups to isolate queries for teams, applications, or different workloads. For example, you may create separate workgroups for two different teams in your organization.

You can also have separate workloads. For example, you can create two independent workgroups, one for automated scheduled applications, such as report generation, and another for ad-hoc usage by analysts.

30-90% cost savings by using columnar data format.

For Athena query, small number of large files performs better than large number of small files.

Amazon Athena is built on open-source engines and frameworks such as Spark, Presto, and Apache Iceberg, giving customers the flexibility to use Python or SQL or work on open data formats.

Avro is not columnar, but splittable. Athena can handle unstructured/semi structured data.

When Athena sees a new data catalog, it creates a table based on that. The metadata in the table tells Athena where the data is located in Amazon S3, and specifies the structure of the data (column names, data types, the name of the table and so on.)

Athena can only query the latest version of data on a versioned Amazon S3 bucket and cannot query previous versions of the data. Athena does not support querying the data in GLACIER.

Using a row-based data format is not suitable for aggregating large amounts of data.

Athena creates metadata only when a table is created.

You can use MSCK REPAIR TABLE or ALTER TABLE ADD PARTITION to load the partition information into the catalog. The MSCK REPAIR TABLE scans S3 and automatically adds new partitions to the catalog. Run 'MSCK REPAIR TABLE' after partitioning the data. Works only with hive style partitions.

It is certainly possible to query the entire data in S3 using Athena, however, it will not be able to match the high performance offered by Redshift.

External schema of PostgreSQL? An external schema is also known as view schema. Each view schema describes the database part that a particular user group is interested and hides the remaining parts from that user group.

Federated query: Allows you to query data sources other than S3 buckets using a data connector. A data connector is implemented in a Lambda function that uses Athena Query Federation SDK. Prebuilt connectors are available for most sources. You can write your own connector or customize prebuilt connectors.

Federated query across regions is better solution. Redshift in Sydney, PostgreSQL in Singapore. Athena can do federated query.

Athena uses presto, an open source, distributed SQL query engine. Automatically executes queries in parallel. Integrates out of the box with Glue. Integrates with QuickSight. Retains query history for 45 days. Keeps query results in S3.

Athena does not support user-defined functions(UDF), INSERT INTO statements, and stored procedures.

Athena performance improvements – data partitioning, convert data to columnar, compressing data, make the files splitable so that parallel scans are possible.

If the files in the target S3 bucket are encrypted, you can perform Athena queries on the encrypted data itself.

Cost controls - Per-query limit, Per-workgroup limit. You can create up to 1000 workgroups per Region in your account. Each account has a primary workgroup and the default permissions allow all authenticated users access to this workgroup.

You pay only for the queries that you run. You are charged based on the amount of data scanned by each query.

Amazon Athena usually just utilizes a single S3 bucket as a source. Having multiple S3 buckets won't improve the overall performance nor achieve parallel processing.

Athena can query data in objects that are stored in different Storage classes (Standard, Standard-IA and Intelligent-Tiering) in Amazon S3. But can't query data in Glacier.

S3 Select is not suitable for analytic workloads. You should use Amazon Athena instead.

Amazon Athena is a query service and is not primarily utilized for loading data into the cluster.

By partitioning your data, you can restrict the amount of data scanned by each query, thus improving performance and reducing cost. Athena leverages Hive for partitioning data. You can partition your data by any key. A common practice is to partition the data based on time, often leading to a multi-level partitioning scheme.

Athena can use Hive style partitions, whose data paths contain key value pairs connected by equal signs (for example, country=us/... or year=2021/month=01/day=26/...).

Amazon Athena allows you to set two types of cost controls: per query limit, per workgroup limit. Per-query control limit automatically cancels queries.

Athena natively supports querying datasets and data sources that are registered with the AWS Glue Data Catalog. When you run Data Manipulation Language (DML) queries in Athena with the Data Catalog as your source, you are using the Data Catalog schema to derive insight from the underlying dataset. When you run Data Definition Language (DDL) queries, the schema you define are defined in the AWS Glue Data Catalog.

From within Athena, you can also run an AWS Glue crawler on a data source to create schema in the AWS Glue Data Catalog.

Use Amazon Athena for querying data in the Amazon S3 bucket using ODBC drivers.

You can use a JDBC (Java Database Connectivity) connection to connect Athena to BI tools and other applications, such as SQL Workbench. Through that connection, data in S3 can be queried.

Amazon Athena is primarily used to query data stored in S3 and not in HDFS.

JSON SerDe Serializes and de-serializes JSON files. (Athena)

Athena supports column names that contain alphanumeric characters and underscores but does not support special characters such as forward slashes, periods, or spaces. If otherwise, you will get

HIVE_METASTORE_ERROR. Permission-related issues usually throw a different error message, such as '*Access denied*' or '*Unauthorized*'.

When you run a CTAS query, Athena writes the results to a specified location in Amazon S3. If you specify partitions, it creates them and stores each partition in a separate partition folder in the same location. (CTAS = Create Table AS)

Bucketing CTAS query results works well when you bucket data by the column that has high cardinality and evenly distributed values. Partitioning CTAS query results works well when the number of partitions you plan to have is limited and the partitioned columns have low cardinality.

Typically, the columns you use for bucketing differ from those you use for partitioning.

Athena query returns zero records if your table location is similar to the following: s3://doc-example-bucket/table1.csv s3://doc-example-bucket/table2.csv. To resolve this issue, create individual S3 prefixes for each table similar to the following: s3://doc-example-bucket/table1/table1.csv s3://doc-example-bucket/table2/table2.csv

If the files in your S3 path have names that start with an underscore or a dot, then Athena considers these files as placeholders. Athena ignores these files when processing a query. (ex: s3://doc-example-bucket/athena/inputdata/_file1)

Athena allows you to set two types of cost controls: per-query limit and per-workgroup limit. For each workgroup, you can set only one per-query limit and multiple per-workgroup limits.

The workgroup-wide data usage control limit specifies the total amount of data scanned for all queries that run in this workgroup during the specified time period. You can create multiple limits per workgroup.

The workgroup-wide query limit allows you to set multiple thresholds on hourly or daily aggregates on data scanned by queries running in the workgroup. If the aggregate amount of data scanned exceeds the threshold, you can push a notification to an Amazon SNS topic.

If any query that runs in the workgroup exceeds the limit, query is canceled. You can create only one per-query control limit in a workgroup and it applies to each query that runs in it.

You might see more tables in the AWS Glue console than in the Athena console for the following reasons: 1) You've created tables that point to different data sources. The Athena console displays tables that point to S3 paths only. AWS Glue lists tables that point to different data sources, such as RDS and DynamoDB. 2) You've created tables in formats that aren't supported by Athena, such as XML. These tables appear in the AWS Glue Data Catalog, but not in the Athena console.

MSCK REPAIR TABLE command scans a file system such as Amazon S3 for Hive compatible partitions that were added to the file system after the Athena table was created.

Partitioning and bucketing are two ways to reduce the amount of data Athena must scan when you run a query. Partitioning and bucketing are complementary. In data bucketing, records that have the same value for a property go into the same bucket. Records are distributed as evenly as possible among buckets so that each bucket has roughly the same amount of data. Bucketing is like an additional sort key within partitions?

Objects in S3 buckets that are not from the same region are not accessible. If the data is encrypted in Amazon S3, it must be stored in the same Region.

Athena supports queries of objects that are stored with multiple storage classes in the same S3 bucket.

Athena workgroups give you the ability to track costs and set limits on the amount of data each query or the entire workgroup can process.

For Athena to scan the Amazon Redshift data, additional costs that are related to the federated query engine would occur.

User Defined Functions (UDF) in Athena allow you to create custom functions to process records. To use a UDF in Athena, you write a USING EXTERNAL FUNCTION clause before a SELECT statement in an SQL query.

Athena is not ETL; Athena is not visualization tool;

Partitioning by the session ID will allow a single processor to process all the actions for a user session in order.

Athena v2 based on presto; Athena v3 based on Trino

Athena stores table definitions in Glue data catalog. Athena uses Apache Hive DDL to define tables.

Athena uses Hive only for DDL and creation/modification and deletion of tables or partitions. Athena uses Trino and Presto when you run SQL queries on S3.

Athena uses SerDes to interpret the data read from S3.

Data Pipeline

You can schedule an AWS data pipeline for incremental copy of RDS data to S3.

You can use AWS data pipeline to automate the movement and transformation of data. Based on a schedule you define, your pipeline regularly performs processing activities such as distributed data copy, SQL transforms, MapReduce applications, or custom scripts against destinations such as S3, Amazon RDS, Amazon Redshift or Amazon DynamoDB.

You can use AWS data pipeline to set up a one-time export of RDS data to S3 as well as an ongoing incremental copy of RDS data to S3. You can also use AWS data pipeline to load data from S3 to Redshift.

Data Pipeline Native integration with S3, DynamoDB, RDS, EMR, EC2, and Redshift. Supports JDBC. Task Runner is installed and runs automatically on resources created by your pipeline definitions. Data nodes in data pipeline.

Data Pipeline: is a web service that helps you reliably process and move data between different AWS compute and storage services, as well as on-premises data sources, at specified intervals. With AWS Data Pipeline, you can regularly access your data where it's stored, transform and process it, and efficiently transfer the results to AWS services such as S3, RDS, DynamoDB, and EMR.

Amazon AppFlow - Fully managed integration service that enables you to securely transfer data between Software as a Service (SaaS) applications and AWS.

Data Science

Pandas - This library is used for structured data operations, like import CSV files, create dataframes, and data preparation

Numpy - This is a mathematical library. Has a powerful N-dimensional array object, linear algebra, Fourier transform, etc.

Matplotlib - This library is used for visualization of data. SciPy - This library has linear algebra modules; When we load a data set using Pandas, all blank cells are automatically converted into "NaN" values. seaborn library (sns) for showing heatmap.

AWS Data Exchange makes it more efficient for AWS customers to securely exchange and use third-party data in AWS. When a data provider publishes a data product on AWS Data Exchange, it is also listed in AWS Marketplace.

Once subscribed to a data product, you can use the AWS Data Exchange API to load data directly into Amazon S3 and then analyze it with a wide variety of AWS analytics and machine learning services.

Apache MXNet on AWS is a fast and scalable training and inference framework with an easy-to-use, concise API for machine learning.

For ML, if cost is the primary optimization criteria, then SageMaker does not represent the best option, as the SageMaker instances are estimated to be 40% costlier than Amazon EC2 instances. Run your ML app in EC2 instead.

DynamoDB

DDB – ProvisionedCapacityExceed – exponential backoff; If RCU issue, we can use DynamoDB Accelerator (DAX)

Ondemand – 2.5 times more expensive than provisioned. RRU/WRU (read Request Unit)

If the writes are throttled on the GSI, then the main table will be throttled! (Though main table uses separate WCU capacity)

DynamoDB Streams are made of shards, just like Kinesis Data Streams. Records are not retroactively populated in a stream after enabling it. Retained for 24 hours. Data from DDB stream can be sent to Kinesis Data Streams, AWS Lambda, Kinesis Client Library applications.

DDB - Parallel Scan increases speed; batchOperations reduce latency

DynamoDB can be used for indexing S3 object metadata!

NoSQL databases don't perform aggregations such as "SUM", "AVG"

If you have to load data from DynamoDB to anywhere, use DynamoDB streams.

EMR is natively integrated with DynamoDB.

EMR

Transient vs Long running clusters.

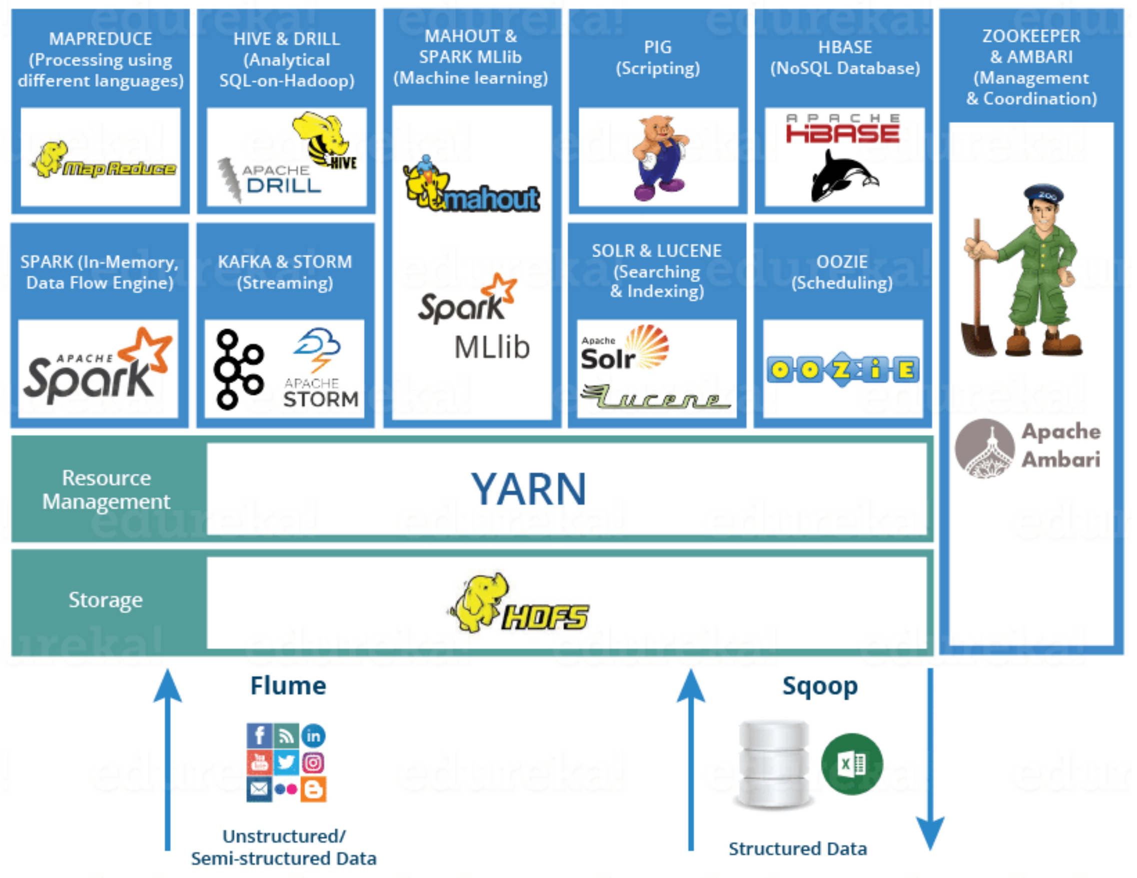
AWS Data Pipeline to schedule and start your clusters.

EMRFS (S3 based) and HDFS; EMR Serverless (Spark, Hive, Presto)

Removing EMR core node risks data loss. Task node don't have data. (master/core/task nodes)

Spark is not meant for OLTP.

Pig introduces Pig Latin, a scripting language that lets you use SQL like syntax to define your map and reduce steps. If you use MapReduce, you have to write code. Pig simplifies it.


MAPREDUCE (Processing using different languages)
HIVE & DRILL (Analytical SQL-on-Hadoop)
MAHOUT & SPARK MLlib (Machine learning)
PIG (Scripting)
HBASE (NoSQL Database)
ZOOKEEPER & AMBARI (Management & Coordination)
SPARK (In-Memory, Data Flow Engine)
KAFKA & STORM (Streaming)
SOLR & LUCENE (Searching & Indexing)
OOZIE (Scheduling)
Resource Management
YARN
Storage
HDFS
Apache Ambari
Flume
Unstructured/ Semi-structured Data
Sqoop
Structured Data

- HDFS -> Hadoop Distributed File System
- YARN -> Yet Another Resource Negotiator
- MapReduce -> Data processing using programming.
- Spark -> In-memory Data Processing
- PIG, HIVE-> Data Processing Services using Query (SQL-like)
- HBase -> NoSQL Database
- Mahout, Spark MLlib -> Machine Learning
- Apache Drill -> SQL on Hadoop
- Zookeeper -> Managing Cluster
- Oozie -> Job Scheduling
- Flume, Sqoop -> Data Ingesting Services

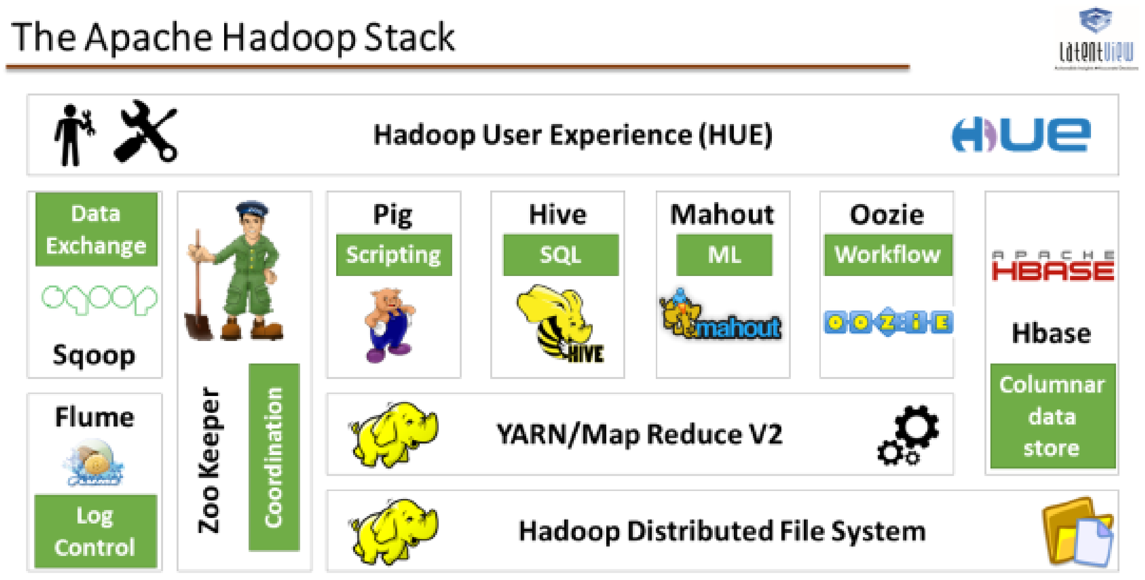
HBase created based on Google's BigTable. Non-relational. In Memory. Can store data in S3 through EMRFS. Better integration with Hadoop (than DynamoDB)

HUE - Hadoop User Experience - Graphical frontend for applications on your EMR cluster

MXnet – Library for neural networks – part of EMR (like TensorFlow?)

Flume - Another way to stream data into your cluster. Originally for log aggregation.

S3DistCP: Tool for copying large amounts of data - 1) From S3 into HDFS   2) From HDFS into S3



Kinesis Data Analytics use Flink under the hood.

RANDOM_CUT_FOREST: SQL function used for anomaly detection on numeric columns in a stream.

Athena cost model - Successful or cancelled queries are charged, failed queries do not. Save 30 to 90%, by using columnar format.

EMR/Spark can do ML? (but it is unmanaged unlike SageMaker)

Apache Tez is a framework that creates a complex directed acyclic graph (DAG) of tasks for processing data. You can use it as an alternative to Hadoop MapReduce for some use cases.

Ganglia open source project is a scalable, distributed system designed to monitor clusters and grids

Presto – Distributed SQL query

An algorithm that supports splitting (LZO, bzip2).

S3DistCp can compress output that is transferred from HDFS to Amazon S3 and vice versa.

Upload the JAR files to an S3 bucket for your mappers and reducers.

Amazon EMR 5.x + uses Apache Tez as its default execution engine. Tez is used by Hive and Pig to process data much more quickly than MapReduce.

Dynamic resizing allows you to add nodes to tune cluster to optimize your application.

Impala is an open source massively parallel processing SQL query engine for data stored in a computer cluster running Apache Hadoop.

Steps (units of work) can be sent to cluster in several ways. By default, steps run sequentially on the Master node using FIFO (“first in, first out”) scheduler. Steps can be added while the cluster is running.

With MapReduce, use OutputFormat interface to customize output. With Hive, use serializer/de-serializer (SerDe) to change format.

Amazon EMR can also save the output directly in Amazon Redshift using Spark-Redshift connector. Else EMR data lasts only for the life of the cluster.

Master node instance count is set to 1 by default and cannot be changed.

Apache Storm is a distributed stream processing computation framework. The project was open sourced after being acquired by Twitter.

MapReduce, Spark and Tez are the EMR computing engines.

Pig scripts are compiled into MapReduce jobs. For Pig, uses engines Tez or MapReduce. Not spark. Support for batch and interactive Pig jobs.

MapReduce high latency; Tez medium latency; Spark/Presto low latency

Mahout - Collection of machine learning algorithms built for Hadoop.

R - Open-source package for statistical computing and graphics.

Oozie for orchestrating big data workflows

Amazon EMR uses Amazon EC2 instance store for core nodes.

By default, data stored in HDFS is encrypted; HDFS - name node + data node

S3DistCP is the AWS version of DistCP.

A Lambda function can spin up an EMR cluster using the RunJobFlow API that will execute the Hive script for the batch process. RunJobFlow creates and starts running a new EMR cluster (Job flow). The cluster runs the steps specified.

HDFS is ephemeral storage. HDFS is useful for caching intermediate results during MapReduce processing or for workloads that have significant random I/O.

YARN does cluster resource management. EMR has an agent on each node that administers YARN components.

Each step is a unit of work that contains instructions to manipulate data, for later processing by software installed on the cluster.

Running EMR cluster can be resized. You can launch an EMR cluster with three master nodes. EMR supports Kerberos authentication.

Launch a primary and secondary EMR HBase cluster. Configure multiple master nodes in the primary cluster and create a secondary read replica cluster in a different Availability Zone. Point the two clusters to the same S3 bucket and hbase.rootdir location.

EMR HBase cluster multiple master nodes can’t manage AZ failures. Using two primary clusters is not supported.

EMR: The instance fleets configuration offers the widest variety of provisioning options for EC2 instances. Each node type has a single instance fleet, and using a task instance fleet is optional. You can specify up to five EC2 instance types per fleet, or 30 EC2 instance types per fleet when you create a cluster using the AWS CLI or Amazon EMR API and an allocation strategy for On-Demand and Spot Instances.

Each Amazon EMR cluster can include up to 50 instance groups: one master instance group that contains one Amazon EC2 instance, a core instance group that contains one or more EC2 instances, and up to 48 optional task instance groups. Each core and task instance group can contain any number of Amazon EC2 instances.

The metric YARNMemoryAvailablePercentage represents the percentage of remaining memory available to YARN. This value is useful for scaling cluster resources based on YARN memory usage.

Spot fleet is applicable to EC2 instances and cannot be used directly with EMR. With EMR, you need to use the instance fleet option which does support automatic scaling.

Instance fleet and uniform instance group don't coexist in a cluster. The instance fleets configuration offers the widest variety of provisioning options for Amazon EC2 instances.

The metric CapacityRemainingGB represents the amount of remaining HDFS disk capacity.

Instead of replacing EC2 with EMR, you should replace EC2 with Lambda. EMR is costlier than EC2. Use Lambda instead.

Using an Amazon EMR cluster to convert the files into a cost-effective format is unnecessary. AWS Glue would suffice.

EMR managed scaling vs EMR Auto scaling: If you need to define custom rules involving custom metrics for applications running in the cluster, you should use Auto Scaling. Else use managed scaling.

Instance fleet does not help autoscaling. Instance group helps auto scaling.

When auto scaling is configured, EMR adds and removes instances based on Amazon CloudWatch metrics that you specify. The following are two of the most common metrics used for automatic scaling in Amazon EMR: YarnMemoryAvailablePercentage: The percentage of remaining memory available to YARN. - ContainerPendingRatio: The ratio of pending containers to containers allocated.

DS2-HDD-more storage and less I/O; DC2-SSD: less storage and more I/O

Amazon EMR archives the log files to Amazon S3 at 5-minute intervals. By default, each Amazon EMR cluster writes log files on the master node. These are written to the /mnt/var/log/ directory.

EMR log encryption with AWS KMS customer-managed key. Logging need to be enabled when you create cluster. You don't need to use lambda to copy the log files to S3.

It takes a lot of time to develop custom Apache Spark Scripts that will curate data to EMR Cluster.

It entails a lot of effort to launch an EMR cluster and configure Apache Hive to join the tables. It requires a lot of development overhead to build Glue ETL scripts. You can't move data to AWS Glue Data Catalog since it is not a data store.

Block Public Access configuration is an account-level configuration that helps you centrally manage public network access to EMR clusters in a region.

The command for S3DistCp is s3-dist-cp, which you add as a step in a cluster or at the command line. Used between S3 and HDFS (both directions). S3DistCp does not support concatenation for Parquet files. Amazon recommends using PySpark instead.

PySpark is the Python API for Apache Spark, an open source, distributed computing framework and set of libraries for real-time, large-scale data processing.

If you enable at-rest encryption for local disks, Amazon EC2 instance-store volumes and the attached EBS storage volumes are encrypted. To encrypt the root volumes, you must enable at-rest encryption for local disks. If you encrypt EMRFS, it won't encrypt root volume.

Local disk encryption can be enabled as part of a security configuration to encrypt root and storage volumes.

Tez - Tez is an extensible framework for building high performance batch and interactive data processing applications, coordinated by YARN in Apache Hadoop.

You can use a bootstrap action to install additional software or customize the configuration of the EMR cluster instances. Bootstrap actions are scripts that run on the cluster after EMR launches the instance using the AMI. You cannot replace custom bootstrap actions with AWS EMR CLI.

You can't configure Amazon EMR to use Amazon S3 instead of HDFS for the Hadoop storage layer. HDFS and the EMR File System (EMRFS), which uses Amazon S3, are both compatible with Amazon EMR, but they're not interchangeable. HDFS is an implementation of the Hadoop FileSystem API, which models POSIX file system behavior. EMRFS is an object store, not a file system.

An EMR cluster can only reside in a single Availability Zone. Hence you can have multiple master nodes in same AZ. Not in multiple AZ.

EMR can use instance fleet with provisioning timeout. It will try for spot instance. After prov timeout, it will switch to OnDemand. Instance group do not have a provisioning timeout.

In question, if it says you can complete work in 1 week, it means no SLA. You can use spot instance. Else instance fleet with prov timeout.

The generation of reports from HDFS would require additional tools.

If you are using spot instance, use EMRFS (not HDFS). You can use core node + HDFS.

IsIdle metric of EMR clusters in CloudWatch. Can terminate clusters/nodes based on that metric.

CUSTOM_JAR step on the EMR cluster. A CUSTOM JAR step can be configured to download a JAR file from an Amazon S3 bucket and execute it.
You can optionally add more steps to the EMR cluster via the AddJobFlowSteps API.

EMR studio is an IDE. You can use both EMR Studio and SageMaker Studio with Amazon EMR.

Amazon EMR supports many applications such as Hive, Pig, and the Spark Streaming library.

YARN for cluster management. Supports other cluster management tools too.

General

Data Ingest -> Store -> Process -> Visualize

Hive, Pig, Hue, Spark, Presto, Oozie, AWS Glue, Mahout, Ganglia

Pig - Simple, high-level, textual data flow language (Pig Latin). Pig is then compiled into MapReduce jobs that run on Hadoop.

Spark – In memory processing framework. Allows in-memory data mining. Spark is faster than MapReduce.

Mahout – clustering, data mining, classification

R - Open-source package for statistical computing and graphics.

Flink – real time stream processing.

Ganglia is a cluster monitoring tool; Apache Mesos: is an open-source project to manage computer clusters.

Apache HBase is a massively scalable, distributed big data store in the Hadoop ecosystem. It is an open-source, non-relational, versioned DB which runs on top of S3 (using EMRFS) or the Hadoop Distributed Filesystem (HDFS), and it is built for random, strictly consistent real time access for tables with billions of rows and millions of columns.

Yarn: Apache Hadoop YARN is the resource management and job scheduling technology in the Hadoop distributed processing framework.

Stream processing is actually batch-based processing of data streams.

SQS: SQS extended client message size >256kb

Flume, Sqoop - data ingestion. Flume - unstructured/semi-structured data ingestion. use this for data streaming;   Sqoop - structured data ingestion. You can use this when you have to take SQL data inside HDFS.

HBase - columnar and NoSQL. This is based on Google's BigTable project.

Pig - is used for scripting. This can reduce LOC of your code (Java 1000 lines = PigLatin 30 lines)

Mahout - Machine Learning tool (like SageMaker?)

Cloudera - for search; Oozie - workflow/schedule orchestration

YARN vs ZooKeeper. YARN handles a cluster of nodes; YARN has 2 components - ResourceManager and NodeManager

Zookeeper- large big data environment management. Manage all big data components.

Storage by HDFS, processing by MapReduce.

Hadoop User Experience (HUE) is an open-source interface which makes Hadoop's use easier.   It is a web-based application. It has a job designer for MapReduce, a file browser for HDFS, an Oozie application for making workflows and coordinators, an Impala, a shell, a Hive UI, and a group of Hadoop APIs.

Using Snowball Edge to ship data to S3 takes a lot of time.

Apache IceBerg is a high-performance format for huge analytic tables. Iceberg brings the reliability and simplicity of SQL tables to big data, while making it possible for engines like Spark, Trino, Flink, Presto, Hive and Impala to safely work with the same tables, at the same time.

AppFlow - Run flows on-demand or on a schedule to keep data in sync across SaaS applications and AWS services. Integrates with AWS PrivateLink to allow private data transfer over AWS rather than public data transfer over the internet. Publishes events on EventBridge.

Flows enable data transfer between the source and destination.

Apache Airflow is an open-source workflow management platform for data engineering pipelines. Started by AirBnB in 2014.

If the bulk operation task of AWS DMS fails, it forces the task to change to one-by-one mode.

Amazon Managed Workflows for Apache Airflow (MWAA)

Glue

Glue ETL – Trigger driven, on a schedule, or on-demand. Glue crawler scans data in S3, creates schema.

Can run periodically; Populates the Glue Data Catalog; Stores only table definition; Original data stays in S3; Once cataloged, you can treat your unstructured data like it's structured; These services can be used to query - Redshift Spectrum, Athena, EMR, QuickSight.

Hive lets you run SQL like queries from EMR. The Glue Data Catalog can serve as a Hive metastore. You can also import a Hive metastore into Glue.

DPU – Data Processing Unit. The CPU is for general-purpose computing, the GPU is for accelerated computing, and the DPU, which moves data around the data center, does data processing.

Glue ETL Jobs are run on a serverless Spark platform. Glue supports JSON, CSV, Avro, Parquet only. Glue can do format conversions.

A DynamicFrame is a collection of DynamicRecords. DynamicRecords are self describing, have a schema. Very much like a Spark DataFrame.

Glue ETL is based on Spark. If you want to use other engines (Hive, Pig, etc..) Data Pipeline EMR would be a better fit.

Streaming ETL – consumes from Kinesis/Kafka, then transforms, writes to S3. (KDF?)

Glue databrew – visual data preparation tool to normalize and cleanup data.

Glue elastic views Builds materialized views (query output as a table) from Aurora, RDS, DynamoDB.

LakeFormation uses Glue.

AWS Glue Studio is a graphical interface that makes it easy to create, run, and monitor ETL jobs in AWS Glue.

Glue is Apache Spark running on EMR running on containers?

Glue jobs Scala or Python based.

Data Pipeline moves data between on-premises data center and cloud.

DynamicFrames are like Spark's in-built data frames, but better suited for doing ETL on semi-structured datasets—in particular cleaning, structuring, or restructuring the data to prepare it for analysis.

Three components of AWS Glue - Data Catalog, ETL Engine, Job Orchestration

In AWS Glue, you can use workflows to create and visualize complex ETL activities involving multiple crawlers, jobs, and triggers. Before adding a workflow, create the jobs and crawlers that the workflow is to include.

You cannot use Glue triggers directly to start a job. Trigger has to be part of workflow.

When you use ETL, small no of large files is preferred?

AWS Glue job usually executes Apache Spark, Spark Streaming, or Python shell scripts only. AWS Glue doesn't directly run the Hive script to perform the batch operation at the specified time.

You can use 'Glue' resource policies to grant access to the corresponding table in the Data Catalog.

As soon as new data becomes available in Amazon S3, you can run an ETL job by invoking AWS Glue ETL jobs using an AWS Lambda function.

You can only use one data catalog per region.

To process your data, you can use AWS Glue built-in transforms. These transforms can be called from your ETL script. Enables you to manipulate your data into different formats.

Glue Dynamic Frame: A distributed table that supports nested data. For schema flexibility with semi-structured data. Each record consists of data and schema. You can use dynamic frames to provide a set of advanced transformations for data cleaning and ETL. DF is for working with semi structured data.

The crawler will write metadata to the AWS Glue Data Catalog. The metadata is stored in a table definition, and the table will be written to a database.

AWS Glue can generate an initial script (Scala or PySpark), but you can also edit the script if you need to add sources, targets, and transforms.

Glue Data Brew: A visual data preparation tool for cleaning and normalizing data to prepare it for analytics and machine learning.

AWS Glue only supports symmetric customer master keys (CMKs).

Set up a staging table in the AWS Glue job. Utilize the DynamicFrameWriter class in AWS Glue to replace the existing rows in the Redshift table before persisting the new data. Glue can do merge operation with staging table.

The upsert feature inserts or updates data if the row that is being inserted already exists in the table. With this Redshift won't have duplicate entries.

You can't use the COPY command to copy data directly from a MySQL database into Amazon Redshift. Go through S3.

Apache Spark's DataFrame can do dropDuplicates()

The AWS Glue Data Catalog maintains metadata about data lakes, data sources, transforms, and targets.

Blueprint is a data management template to ingest data into a data lake.

With AWS Glue directed acyclic graph (DAG), you can monitor the progress of the workflow.

To get the most recent data in Glue Data catalogue, you must implement an automated task that will allow the AWS Glue crawler to update the schema in real-time. Link with s3 event notifications?

Glue - The Standard worker type has a 50 GB disk and 2 executors. A data processing unit (DPU) is a relative measure of processing power that consists of vCPUs and memory. To improve the job execution time, you can enable job metrics in AWS Glue to estimate the number of data processing units (DPUs) that can be used to scale out an AWS Glue job.

job bookmark maintains state information and prevents AWS Glue from reprocessing old data. Job bookmarks are primarily used for incremental data processing.

Modify the job properties and enable job metrics to evaluate the required number of DPUs. Change the maximum capacity parameter value and set it to a higher number.

A crawler can crawl multiple data stores in a single run. Upon completion, the crawler creates or updates one or more tables in your Data Catalog. ETL jobs that you define in AWS Glue use these Data Catalog tables as sources and targets.

A classifier reads the data in a data store. If it recognizes the format of the data, it generates a schema. The classifier also returns a certainty number to indicate how certain the format recognition was.

AWS Glue provides a set of built-in classifiers, but you can also create custom classifiers. AWS Glue invokes custom classifiers first, in the order that you specify in your crawler definition. Depending on the results that are returned from custom classifiers, AWS Glue might also invoke built-in classifiers. If a classifier returns certainty=1.0 during processing, it indicates that it's 100 percent certain that it can create the correct schema.

If no classifier returns certainty=1.0, AWS Glue uses the output of the classifier that has the highest certainty. If no classifier returns a certainty greater than 0.0, AWS Glue returns the default classification string of UNKNOWN. Certainty level has no negative values.

AWS Glue crawlers automatically infer database and table schema from your dataset, storing the associated metadata in the AWS Glue Data Catalog. Glue Job (ETL) vs Glue Crawler

AWS Glue can crawl data in different AWS Regions. When you define an Amazon S3 data store to crawl, you can choose whether to crawl a path in your account or another account. The output of the crawler is one or more metadata tables defined in the AWS Glue Data Catalog.

Crawler can use custom classifiers or built-in classifiers.

Some common causes of long crawler run times are: 1) Frequently adding new data. 2) Crawling being done on compressed files. Crawler needs to decompress them first.

Unless you need to create a table in the AWS Glue Data Catalog and use the table in an ETL job or a downstream service such as Amazon Athena, you don't need to run a crawler.

For ETL jobs, you can use from_options to read the data directly from the data store and use the transformations on the DynamicFrame. When you do this, you don't need a crawler in your ETL pipeline.

To reduce crawler run times: 1) Use an exclude pattern so that only a subset of data is crawled. 2) Run multiple crawlers. 3) Combine smaller files to create larger ones. 4) data should be non-compressed.

Incremental crawl is good if the data catalog already exists.

The AWS Glue Data Catalog is an index to the location, schema, and runtime metrics of your data. You use the information in the Data Catalog to create and monitor your ETL jobs. Information in the Data Catalog is stored as metadata tables, where each table specifies a single datastore.

AWS Glue supports an extension of the PySpark Scala dialect for scripting ETL jobs.

Capturing and loading continuously changing updates from operational data stores into a data lake can be time-consuming and difficult to manage.

We can use the AWS DMS to migrate the on-premises data to an S3 bucket. Then, we run an AWS Glue job to do the data curation.

Create a catalog table using the CreateTable API operation. Catalog table will support custom naming conventions.

Updating the AWS Glue Data Catalog may take some time. Using AWS Glue, one can configure and run a job to transform CSV data to Parquet.

AWS Glue uses a single connection to read the entire dataset. If you're migrating a large JDBC table, read the JDBC table in parallel. Else it may take a long time and fail. If the table doesn't have numeric columns (INT or BIGINT), then use the hashfield option to partition the data.

Adding more DPUs (Data Processing Unit) helps only if the job workload is already parallelized.

For schemas to be considered similar, the following conditions must be true: 1. The partition threshold is higher than 0.7 (70%). 2. The maximum number of different schemas (also referred to as "clusters" in this context) doesn't exceed 5.

In a folder, suppose you have 2 schemas. SCH_A – 8 files; SCH_B – 2 files. Here, the crawler creates only one table, which comprises columns of both schema SCH_A and SCH_B. If the ratio was 70:30%, then 2 separate tables would have been created. 70% is the cut off.

Glue workers are measured in DPUs. G1.x (1 DPU) G2.x (2 DPU) etc..

An exclude pattern tells the crawler to skip certain files or paths. Exclude patterns reduce the number of files that the crawler must list, making the crawler run faster.

The AWS Glue Schema Registry allows you to centrally discover, control, and evolve data stream schemas. A schema defines the structure and format of a data record.

With Glue Schema Registry, you can manage and enforce schemas on your data streaming applications using convenient integrations with Kafka, MSK, Kinesis Data Streams, Managed Service for Apache Flink, and AWS Lambda.

With AWS Glue, you can categorize your data, clean it, enrich it, and move it reliably between various data stores cost-effectively.

You can schedule AWS Glue workflows to run daily. (Scheduled Trigger)

AWS Glue can transform the data that Forecast will ingest. Forecast is a service that provides ML algorithms to predict future on-time series data.

AWS Glue resource policies can be used to control access to Data Catalog resources.

AWS Glue automatically generates Scala or Python code for your ETL jobs

Run the AWS Glue crawler from the us-west-1 region to catalog datasets from all three regions. Once all the data is crawled, run the needed Athena queries from us-west-1.

Hadoop

https://www.datacamp.com/tutorial/tutorial-cloudera-hadoop-tutorial

Started in 2001; 2008 - Yahoo open-sourced Hadoop.; 2010 - Apache took it up

Hadoop is an implementation of the MapReduce computing model!

Hive

Apache Hive is an open-source, distributed system that provides data warehouse-like query capabilities. It enables users to read, write, and manage petabytes of data using a SQL-like interface.

Apache Hive is natively supported in Amazon EMR; Originally developed by Facebook. Developed since developers were more familiar with SQL & RDBMS.

Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data.

Configure Hive to use the AWS Glue Data Catalog as its metastore, if the metastore is shared by different clusters. (in this case, metastore is outside cluster)

Apache Hive is a data warehouse software project built on top of Apache Hadoop for providing data query and analysis. Hive gives an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop.

Hive metastore is like Glue data catalogue.

You can keep your Hive metastore in RDS (external table, persistent) which can be accessed from multiple EMR clusters.

Hive leverages Tez as its query engine by default, and can also run on Apache MapReduce as well.

Hive - SQL-like interface for querying and managing large datasets in distributed storage (S3, HDFS, HBase, etc.)

By default, Hive records metastore information in a MySQL database on the master node's file system. If you need the metastore to persist, you must create an external metastore that exists outside the cluster. You have two options for an external metastore: - AWS Glue Data Catalog, - Amazon RDS or Amazon Aurora.

Hive is generally slower than traditional relational databases due to its batch processing nature. Not suitable for real time processing. There is a latency in converting hive scripts into MapReduce scripts.

IoT

IoT Device Gateway, Device Shadow

MQTT (Message Queuing Telemetry Transport) is a lightweight, publish-subscribe, machine to machine network protocol for message queuing service.

IoT Greengrass brings the compute layer to the device directly.

Use AWS IoT core to collect MQTT messages from devices.

IoT device Gateway – GateWay between IoT sensors and AWS.

IoT Thing Registry = IAM of IoT

Device shadow

Kinesis

ThroughputExceededException – backoff and retry, change partition key, add shards.

The KPL can incur an additional processing delay of up to RecordMaxBufferedTime within the library (user configurable). Applications that cannot tolerate this additional delay may need to use the AWS SDK directly.

Kinesis agent monitors Log files and sends them to Kinesis Data Streams. Java based agent, built on top of KPL.

Maximum of 5 GetRecords API calls per shard per second => 200ms latency

Checkpointing feature: If your application goes down and comes back, it can restart processing from the point where it stopped earlier.

Kinesis Connector Library is old – used to write data to S3, DynamoDB, OpenSearch, Redshift etc. Already replaced by Firehose and Lambda.

Fanout consumers limit – 20 consumers; 70 ms latency; std

Auto Scaling is not a native feature of Kinesis. We can implement Auto Scaling with AWS Lambda – call UpdateShardCount. For1000 shards, it takes 30K seconds (8.3 hours) to double the shards to 2000.

Kinesis Data Streams deduplication with unique id. Consumer retries can make your application read the same data twice. Make your application idempotent.

SQS can be used to buffer writes to a Database. And save WCU.

MSK cluster uses Apache zoo keeper

Snowmobile; use for more than 10 PB to same location; else snowball.

Prometheus (Open Source Monitoring) used for Kafka monitoring

Sliding window analysis: In Kinesis Data Analytics, instead of grouping records using GROUP BY, you can define a time-based or row-based window.

For new projects, we recommend that you use the new Managed Service for Apache Flink Studio over Kinesis Data Analytics for SQL Applications.

Kinesis Data Streams is open-ended for both producer and consumer. It supports KCL and works with Spark. But Kinesis Firehose is open-ended for producers only. Data is stored in S3, Redshift, and OpenSearch.

Amazon Kinesis synchronously replicates data across three facilities (3 AZs?) in a Region.

You can register up to 20 consumers per data stream. A given consumer can only be registered with one data stream at a time.

KDF -> S3 -> Redshift – this introduces latency of minutes

Single Kinesis data stream is capable of collecting data from multiple sources.

Amazon Kinesis Data Analytics is the easiest way to analyze streaming data, gain actionable insights, and respond to your business and customer needs in real-time.

Stagger windows allow for multiple overlapping windows. It is more suited for analyzing data that comes in at irregular intervals.

ProvisionedThroughputExceededException – means lack of shards

Random cut forest (RCF) can be used with KDA for anomaly detection.

KDA is the easiest way to transform and analyze streaming data in real time with Apache Flink. Apache Flink is an open source framework and engine for processing data streams.

How scaling works in KDS: Create a CloudWatch alarm that monitors Kinesis Data Stream shard metrics. When a custom threshold of the alarm is reached, It sends a notification to an Application Auto Scaling policy that calls the UpdateShardCount API operation. The call passes the new number of Kinesis Data Stream shards for the desired capacity.

Lambda transformation function of Kinesis Firehose is primarily used for data transformation and not for analyzing, detecting and sending notifications.

Data inserted in Kinesis cannot be deleted. (immutable).   Kinesis connector library (KCL), Kinesis Client Library

Amazon Kinesis Data Firehose can be considered as an ETL service!

Hot shard, Kinesis agent, Enhanced Fanout; Kinesis Producer Library will incur additional delays.

Amazon S3 Kafka Connector

'ProvisionedThroughputExceededException' comes after burst capacity is also consumed.

The Kinesis Producer Library is an easy-to-use, highly configurable library that helps you write to a Kinesis data stream.

Detailed monitoring does not send data in real time. It sends data once every minute.

Kinesis Data Streams has two capacity modes: on-demand and provisioned. In the on-demand mode, pricing is based on the volume of data ingested and retrieved along with a per-hour charge for each data stream in your account. With provisioned capacity mode, you specify the number of shards necessary for your application based on its write and read request rate. You pay by the shard hour. A PUT Payload Unit is counted in 25 KB payload "chunks".

When you reshard, data records that were flowing to the parent shards are re-routed to flow to the child shards based on the hash key values that the data-record partition keys map to. However, any data records that were in the parent shards before the reshard remain in those shards. In other words, the parent shards do not disappear when the reshard occurs. They persist along with the data they contained before the reshard, until the configured retention time.

If you read data from the child shards before reading all data from the parent shards, you could read data for a particular hash key out of order given by the data records' sequence numbers.

PutRecords API does not have a *SequenceNumberforOrdering* parameter.

Shards provide a predefined write and read capacity. As workloads grow, an application may read or write to a shard at a rate that exceeds its capacity, creating a hot shard and requiring you to add capacity quickly. It can lead to throttling errors in the stream.

Distribute the hash key space evenly across shards to avoid hot shards. UpdateShardCount, SplitShard, MergeShards

Use an error retry and exponential backoff mechanism if you get ReadProvisionedThroughputExceeded errors

Amazon Kinesis Data Firehose captures, transforms, and loads streaming data from sources such as a Kinesis data stream, the Kinesis Agent, or Amazon CloudWatch Logs into downstream services such as Kinesis Data Analytics or Amazon S3. Then AWS Lambda can do reformatting/enrichment before sending it to next stage.

Data sent from CloudWatch Logs to Amazon Kinesis Data Firehose is already compressed with gzip.

Create a CloudWatch Logs subscription that sends log events to your delivery stream.

If you want something near real time, running a task every hour won't do.

If you send your logs from EC2 to CloudWatch and then to Kinesis, then you don't need Kinesis agent in EC2.

Amazon Kinesis Data Streams has no direct integration with Amazon OpenSearch Service. You have to use Amazon Kinesis Firehose instead.

Amazon Kinesis Data Firehose can convert the format of your input data from JSON to Apache Parquet or Apache ORC before storing the data in Amazon S3. Parquet and ORC are columnar data formats that enable faster queries. Snappy compression happens automatically as part of the conversion process. The framing format for Snappy that Kinesis Data Firehose uses, is compatible with Hadoop. It means that you can run queries on this data in Athena.

KDF automatically scales to match your data's throughput. It can also batch, compress, and encrypt the data before loading it, minimizing the storage used at the destination and increasing security.

The PutRecords operation sends multiple records to your stream per HTTP request, and the singular PutRecord operation sends records to your stream one at a time. You should prefer using PutRecords for most applications because it will achieve a higher throughput per data producer.

The enhanced fan-out feature is only available with Amazon Kinesis Data Streams.

Windowed SQL - bound queries using a window defined in terms of time or rows.

Tumbling window (distinct time-based windows that open and close at regular intervals.), sliding window (fixed time or rowcount interval.), or stagger window (suited for analysing groups of data that arrive at inconsistent times.)

Tumbling window for regular arriving data. No overlap between windows;   Sliding Window - record can be part of multiple windows and be processed with each window.; Continuous query does not use a window.

The HTTP/2 data retrieval API allows data to be delivered from producers to consumers in 70 milliseconds or better. You have to configure stream HTTP/2 consumers with an enhanced fan-out feature to benefit from the performance advantages.

enhanced fan-out feature - it takes time to implement and could get very expensive.

An Amazon Kinesis Data Analytics application can receive input from a single streaming source and, optionally, use one reference data source to enrich the data coming in from streaming sources. You must store reference data as an object in your Amazon S3 bucket. When the application starts, Amazon Kinesis Data Analytics reads the Amazon S3 object and creates an in-application reference table. Your application code can then join it with an in-application stream.

You can explicitly call the UpdateApplication API to refresh the reference source. DynamoDB and AWS Glue Data Catalog are not a valid reference data source for a Kinesis data analytics application.

1 KPU = 1 vCPU + 4GB RAM (Kinesis Processing Unit)

CNN and RNN (recurrent neural networks). Flink is a framework for processing data streams.

When a host needs to send many records per second (RPS) to Amazon Kinesis, simply calling the basic PutRecord API action in a loop is inadequate. To reduce overhead and increase throughput, the application must batch records and implement parallel HTTP requests. This is better than increasing Shards. Use Batch Messages is correct answer.

You can configure Amazon Kinesis Data Firehose to aggregate and collate CloudWatch Logs from different AWS accounts and receive their log events in a centralized logging AWS Account (cross-account data sharing) by using a CloudWatch Logs destination and then creating a Subscription Filter.

CloudWatch Logs agent cannot publish data to a Kinesis Data Firehose stream. The Kinesis Data Firehose stream cannot subscribe to CloudWatch Events. But, you can use a (CloudWatch) subscription filter with Kinesis Streams, Lambda, or Kinesis Data Firehose.

If Kinesis Data Firehose scales up to four times, the buffer size reduces to one-quarter of the overall buffer size. Hence, there will be four different channels creating four files in S3 during the same time interval. So, you will see more no of small files in S3 now.

When data delivery to destination is falling behind data writing to a delivery stream, Firehose raises buffer size dynamically to catch up and make sure that all data is delivered to the destination.

There is one KDS with 4 shards. One EC2 in ASG processing the 4 shards. If an EC2 is added in ASG, when the KCL worker starts up on the second instance, it load-balances with the first instance, and each instance will now process two shards.

GetRecords.IteratorAgeMilliseconds measures the age in milliseconds of the last record in the stream for all GetRecords requests. A value of zero for this metric indicates that the records are current within the stream. A lower value is preferred.

GetRecords.Latency - GetRecords.Latency measures the time taken for each GetRecords operation on the stream over a specified time period. ReadProvisionedThroughputExceeded measures the count of GetRecords calls that throttled during a given time period.

Duplicate data in Kinesis data stream due to networking issues (no ack received so producer sent msg again). Include primary key within the record to remove duplicates later when processing.

The KPL and the KCL cannot integrate with a Hive metastore for schema validation and evolution. They can't integrate with Data Catalogue as well. Use Glue schema registry instead?

Lambda

Lambda with event source mapping to read data from Kinesis Data streams. If you want to use Lambda for stateful app, then use DynamoDB or S3 to keep track of state.

AWS Lambda supports Parallelization Factor, a feature that allows you to process one shard of a Kinesis or DynamoDB data stream with more than one Lambda invocation simultaneously. Parallelization Factor can be set to increase concurrent Lambda invocations for each shard, which by default is 1. It allows for faster stream processing without the need to over-scale the number of shards while still guaranteeing the order of records processed.

Use of a Lambda function to update the table partitions entails a lot of development work.

Lambda is not the right choice to build a high volume and high-velocity streaming solution. Use Kinesis.

MSK, Kafka

MSK serverless is available.

MSK connect – source connectors (inward) and sink connectors (outward). To allow Amazon MSK Connect to access the internet, you can use VPC and set up a NAT gateway.

MSK connect units(MCU), Worker, Provisioned & auto scaled capacity modes.

You can migrate your clusters using Apache Kafka's MirrorMaker. A) Apache Kafka cluster to Amazon MSK. b) From one MSK cluster to another

For MSK Connect, you are charged for the number and size (MCUs) of each Kafka Connect worker.

A worker is a JVM process that runs the connector logic. Each worker creates a set of tasks that can operate in parallel threads and copy the data. The total capacity of a connector is determined by the number of workers and the number of MCUs per worker.

From Kafka, you can write to KDS, not to KDF/KDA?

Amazon Managed Streaming for Apache Kafka (Amazon MSK) is a fully managed service that makes it easy for you to build and run applications that use Kafka to process streaming data. Kafka is an open-source platform for building real-time streaming data pipelines and applications.

With Amazon MSK, you can use Apache Kafka APIs to populate data lakes, stream changes to and from databases, and power machine learning and analytics applications.
PyTorch, mxNet,

It is not easy to set up Kafka, and it takes more development time.

Kafka - size limit 1MB, but with custom config you can change it to 10MB.

Apache Kafka is a distributed event store and stream-processing platform. It is an open-source system; written in Java and Scala. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds.

An Amazon Managed Streaming for Kafka cluster can be used to deliver the messages with very low latency.

OpenSearch

There is no ElasticSearch Import feature to load data from S3. You can use the Database Migration Service to load the entire data from S3 to ElasticSearch. But it's costly to read entire 50TB data.

You can use CloudSearch to index and search both structured data and plain text. It can integrate with API Gateway.

Search domain – search partition - A facet is an index field that represents a category that you want to use to refine and filter search results. To make your data searchable, you need to format your data in JSON or XML. CloudSearch returns results in JSON. Sends relevance-score.

Scale for traffic, scale for data. When you turn on the Multi-AZ option, CloudSearch provisions and maintains extra instances for your search domain in a second AZ to ensure high availability. The maximum number of AZs a domain can be deployed in is two.

OpenSearch service offers open-source Elasticsearch APIs, managed Kibana, and integrations with Logstash and other AWS Services. This combination is often coined as the ELK Stack.

You can create multiple Elasticsearch indices within the same domain. Elasticsearch automatically distributes the indices and any associated replicas between the instances allocated to the domain. Alerting feature. Can run SQL queries.

OpenSearch use instance store (up to 3PB) or EBS for storage. manual snapshots and automated snapshots. For migration, you use manual snapshots.

Before you can search data, you must index it. Indexing is the method by which search engines organize data for fast retrieval. In Elasticsearch, the basic unit of data is a JSON document. Within an index, Elasticsearch organizes documents into types.

Easily ingest structured and unstructured data into your Amazon Elasticsearch domain with Logstash, an open-source data pipeline that helps you process logs and other event data. You can also ingest data into your Amazon Elasticsearch domain using Amazon Kinesis Firehose, AWS IoT, or Amazon CloudWatch Logs.

You can deploy your Amazon OpenSearch instances across multiple AZs (up to three). If you enable replicas for your indexes, the shards will automatically be distributed such that you have cross-zone replication.

You can either launch your domain within a VPC or use a public endpoint, but not in both. You cannot switch your domain from a VPC to a public endpoint and vice versa. You can't launch your domain within a VPC that uses dedicated tenancy. You cannot move your domain to a different VPC than where it was initially launched in.

OpenSearch - You pay for EC2 used and EBS used. You can use Reserved Instances.

OpenSearch Use cases: clickstream analytics, log analytics, real time application monitoring.

OpenSearch: Try to keep a shard size between 10–50 GiB. Large shards can make it difficult for OpenSearch to recover from failure, but because each shard uses some amount of CPU and memory, having too many small shards can cause performance issues and out of memory errors.

Increasing master nodes in OpenSearch will only increase the cluster stability.

The JVMMemoryPressure error signifies that there is an unbalanced shard allocations across nodes. This means that there are too many shards in the Amazon OpenSearch cluster and not the other way around.

Using the OpenSearch dashboard is free. (Kibana)

OpenSearch service can do log analysis and visualization.

Amazon OpenSearch Service can already do the job of Amazon Kinesis Data Analytics and Amazon QuickSight at a much lower cost with lower operational overhead.

Opensearch is not serverless, but it is fully managed. Index state management (ISM), cross cluster replication (Opensearch related)

It will take long time to load data to OpenSearch and index the data.

Neptune, DocumentDB

DocumentDB - Fully managed document database service. Data is stored in JSON-like documents. Compatible with MongoDB. Flexible schema. Commonly used for content management, user profiles, and real-time big data.

Amazon Neptune is a fully managed graph database service used for building applications that work with highly connected datasets. Optimized for storing billions of relationships between pieces of information. Supports graph query languages like Apache TinkerPop Gremlin and W3C's SPARQL.

Neptune use cases – social networking, recommendation engine, knowledge graph, identity graph

Apache Hudi brings core warehouse and database functionality directly to a data lake.

Pig

The language offered by Pig, the Pig Latin, is roughly similar to scripting languages such as Perl, Python, or Ruby. However, it is more specific than the latter and is best described as a "data flow language".

To analyze data using Apache Pig, programmers need to write scripts using Pig Latin language. All these scripts are internally converted to Map and Reduce tasks. Apache Pig has a component known as Pig Engine that accepts the Pig Latin scripts as input and converts those scripts into MapReduce jobs. We can perform all the data manipulation operations in Hadoop using Pig.

Pig use case: ETL on unstructured data, perform simple SQL operations.

Presto

Presto is a distributed, interactive query engine for big data using the SQL query language. Its architecture allows users to query data sources such as Hadoop, Cassandra, Kafka, AWS S3, Alluxio, MySQL, MongoDB and Teradata, and allows use of multiple data sources within a query.

Started by Facebook. Presto can query from relational and non-relational data

Presto - can connect to many different “big data” databases and data stores at once, and query across them. Athena uses Presto under the hood.

Trino is the new name for PrestoSQL?

When a Presto task fails, the entire job must be restarted from the beginning.

Presto - Open-source, distributed, in-memory SQL query engine. (Athena uses this!)

Presto is complimentary to Hadoop; it does not have its own storage system.

Quicksight

QuickSight is for ad hoc queries, analysis, and visualization. QS can do some transformation – but should not be used as ETL.

Quicksight can only access data stored in the same region. QS has NLP capability?

Quicksight author, reader. QuickSight Reader sessions are of 30-minute duration each.

QuickSight supports all possible data sources. ‘Augment with SageMaker’ option.

QuickSight does inference on the full data every time it refreshes.

You have several options to get your data into Amazon QuickSight: file upload, connect to AWS data sources, connect to external data stores over JDBC/ODBC, or through other API-based data store connectors.

If your VPC has been set up with public connectivity, you can add Amazon QuickSight's IP address range to your database instances' security group rules to enable traffic flow into your VPC and database instances.

Row-level security (RLS) enables QuickSight dataset owners to control access to data at row granularity based on permissions associated with the user interacting with the data.

SPICE (Super-fast, Parallel, In-memory Calculation Engine) is the robust in-memory engine that QuickSight uses. SPICE is engineered to rapidly perform advanced calculations and serve data.

If it takes more than 30 minutes to import your data into SPICE it will time out

QuickSight Q – NLP; 10GB SPICE

You can refresh your SPICE datasets at any time. Refreshing imports the data into SPICE again.

To allow Amazon QuickSight to access the S3 bucket, add the Amazon QuickSight service role (aws-quicksight-service-role-v0) as an exception in your Deny policy.

Funnel Charts are more useful for trends.

Row-level security only works for fields containing textual data (string, char, varchar, and so on). It doesn't currently work for dates or numeric fields. Anomaly detection is not supported for datasets that use row-level security (RLS).

IAM Policies is not used directly for authorizing QuickSight users.

Row-level security is configured in the dataset, not in a manifest file.

AD Connector is the ideal choice when you want to leverage an existing on-premises AD infrastructure without the need to duplicate user information in the cloud.

For Amazon QuickSight to connect to an Amazon Redshift instance, you must create a new security group for that instance. You can configure such a security group irrespective of the Redshift cluster having been created in a VPC or not.

Redshift cluster, as well as a QuickSight dashboard, can be created outside of a VPC.

If the QuickSight dashboard and Redshift cluster are in separate Regions, you cannot use VPC endpoint for the given scenario.

Firehose cannot directly write into a DynamoDB table. You can use Lambda. QuickSight can have redshift as the data source.

Amazon QuickSight is built with "SPICE" – a Super-fast, Parallel, In-memory Calculation Engine. Built from the ground up for the cloud, SPICE uses a combination of columnar storage, in-memory technologies enabled through the latest hardware innovations and machine code generation to run interactive queries on large datasets and get rapid responses.

Quicksight Enterprise edition additionally offers encryption at rest and Microsoft Active Directory integration. Only Enterprise edition can support 200Gb data. You cannot create Readers using the Standard edition of QuickSight. Only QuickSight Enterprise edition can be integrated with the Amazon VPC service.

Quicksight can provide ML insights. ML powered anomaly detections and forecasting. Allows you to schedule automatic email-based reports.

If your data file is encrypted with an AWS KMS key, grant permissions to QuickSight IAM role to decrypt the key. You can run the create-grant command in AWS CLI to do this.

SPICE uses columnar storage, in-memory technologies. Data in SPICE is persisted until it is explicitly deleted by the user. SPICE engine supports up to 500 GB.

QuickSight Author, Reader; Standard vs Enterprise edition

You can also connect Amazon QuickSight to an Amazon EC2 or on-premises database.

QS AutoGraph allows to select the most appropriate visualizations based on the data properties.

Combo Charts and Line Charts are ideal for illustrating trends and movements over a specific timeline. Donut Charts are useful to show a percentage of the total amount. Heat Maps help to easily track outliers, magnitudes, and trends in two dimensions.

Tree maps can be used to represent hierarchical data in the form of nested rectangles. Waterfall Charts - Use a waterfall chart to visualize a sequential summation as values are added or subtracted. Funnel Charts - Use a funnel chart to visualize data that moves across multiple stages in a linear process.

VPC endpoints can only be used to access resources in the same Region as the endpoint.

For QuickSight to connect to a Redshift instance, you must create a new security group for that instance. This security group contains an inbound rule authorizing access from the appropriate CIDR address block for the Amazon QuickSight servers in that AWS Region.

A Redshift cluster, or a QuickSight dashboard, can be created outside of a VPC. Hence NACL won't work. SG is needed.

A QuickSight Author is a user who can connect to data sources (within AWS or outside), create visuals and analyze data. A QuickSight Reader is a user who consumes interactive dashboards.

Data stored in SPICE can be reused multiple times without incurring additional costs. If you use Athena as data source, it charges per query.

Parquet is always in compressed format?

Quicksight provides a) ML powered forecasting b) ML powered anomaly detection c) autonarratives - An 'autonarrative' is just a natural-language summary widget that displays descriptive text instead of charts. Autograph vs autonarrative.

QuickSight uses machine learning to help you uncover hidden insights and trends in your data, and forecast business metrics.

Quicksight in one region can access redshift in another region if security group allows.

If the analyst's user credentials do not have the necessary permissions to access the table containing the confidential data, it will not be visible in the QuickSight console.

Connecting to Athena from QuickSight is a 1-click process. There's no need to provide endpoints, username, and password. Simply select Athena as your data source, select the database and tables you want to analyze, and start visualizing in QuickSight.

To successfully connect QuickSight to the S3 buckets used by Athena, make sure that you authorized QuickSight to access the S3 account. It's not enough that you, the user, are authorized. Amazon QuickSight must be authorized separately.

To authorize Amazon QuickSight to access your Amazon S3 bucket, go to the QuickSight Console and do it.

QuickSight has options to refresh data on a Daily, Weekly, or Monthly basis. For enterprise edition only, you have an option to choose Hourly refresh. But if you have to refresh every 10 secs, then use OpenSearch dashboard (Kibana)

QuickSight supports identity federation in both Standard and Enterprise editions.

Configure Amazon QuickSight to use customer-provided keys imported into AWS KMS - This option is incorrect since all keys associated with Amazon QuickSight are managed by AWS.

QuickSight Enterprise edition integrates with your existing directories, using either Microsoft Active Directory or single sign-on (SSO) using SAML.

Datasets created using Amazon S3 as the data source are automatically imported into SPICE. The Amazon QuickSight console allows for the refresh of SPICE data on a schedule.

You cannot use pivot tables to identify trends and outliers.

RDS, ElastiCache

Amazon ElastiCache for Redis is a blazing fast in-memory data store that provides sub-millisecond latency to power internet-scale real-time applications.

To power a leaderboard, you can use ElastiCache Redis or DAX.

Automated RDS snapshots are not viewable on any S3 bucket that you own. You have to export the snapshot first in order to access it. Remember that the underlying S3 bucket that hosts the automated snapshots is owned by AWS. On the other hand, manual snapshots are different since you own the S3 bucket where they are stored.

You can't select a subset of data to be synced on an RDS read replica.

Redshift

Leader Node, compute Node; Dense Storage, Dense Compute

Redshift has much better performance than Athena for complex analytical queries.

Limited to single AZ. Replication within cluster

Auto, Even, Key, All – Redshift distribution styles. Even – round robin distribution to all nodes. Key – based on a specific column, All – complete table to all nodes/slices.

Rows are stored on disk in sorted order based on the column you designate as a sort key.

COPY command from S3, EMR, DynamoDB, remote hosts; UNLOAD command - Unload from a Redshift table into files in S3.

Snapshot copy grant; Redshift Workload Management;

DBLink PostgreSQL to Redshift

Automatic Concurrency scaling; Large queries – concurrency lowered.

Define up to 8 queues, up to concurrency level 50. (One default queue with concurrency level of 5 (5 queries at)

Short query acceleration – separate queue for short queries

Classic and elastic resize; RA3 and DC2 nodes can be only halved or doubled.

VACUUM command recovers space from deleted rows.

AQUA – Advanced Query Accelerator; RPU – Redshift Processing Unit.

both provisioned and serverless options.

Consider using RA3 if. A) You need the flexibility to scale and pay for compute separate from storage. B) You query a fraction of your total data. C) Your data volume is growing rapidly or is expected to grow rapidly. D) You want the flexibility to size the cluster based only on your performance needs.

RA3 instances with managed storage use high performance SSDs for your hot data and Amazon S3 for your cold data.

Amazon Redshift spatial provides location-based analytics for rich insights into your data.

Redshift auto-copy provides the ability to automate copy statements by tracking Amazon S3 folders and ingesting new files without customer intervention.

Amazon Aurora Zero-ETL to Amazon Redshift enables Amazon Aurora and Amazon Redshift customers to run near real-time analytics. Enables analytics & machine learning on petabytes of transactional data by offering a fully managed solution for making transactional data from Amazon Aurora available in Amazon Redshift within seconds of being written.

We provide a method to use Redshift Materialized Views (MVs) = view based on output of a query. You can query from that materialized view afterwards, as though the view is a table.

AWS datashare between Redshift environments, between accounts.

Cross-database queries give you flexibility to organize data as separate databases to support multi-tenant configurations.

When you use the Concurrency Scaling feature, the cluster is fully available for read and write during concurrency scaling. With Elastic resize, the cluster is unavailable for four to eight minutes of the resize period. With the Redshift RA3 storage elasticity in managed storage, the cluster is fully available and data is automatically moved between managed storage and compute nodes.

Elastic Resize adds or removes nodes from a single Redshift cluster within minutes to manage its query throughput. For example, an ETL workload for certain hours in a day or month-end reporting might need additional Amazon Redshift resources to complete on time.

Concurrency Scaling adds additional cluster resources to increase the overall query concurrency. Concurrency Scaling is a massively scalable pool of Amazon Redshift resources and customers do not have direct access.

With support for dynamic data masking, customers can easily protect their sensitive data and control granular access by managing Data Masking policies.

Amazon Redshift will automatically detect and replace a failed node in your data warehouse cluster. On Dense Compute (DC) and Dense Storage (DS2) clusters, the data is stored on the compute nodes to ensure high data durability. When a node is replaced, the data is refreshed from the mirror copy on the other node.

RA3 clusters and Redshift serverless are not impacted the same way since the data is stored in Amazon S3 and the local drive is just used as a data cache. The data warehouse cluster will be unavailable for queries and updates until a replacement node is provisioned and added to the DB.

If your Amazon Redshift data warehouse is a single-AZ deployment and the cluster's Availability Zone becomes unavailable, then Amazon Redshift will automatically move your cluster to another AZ without any data loss or application changes. To activate this, you must enable the relocation capability in your cluster configuration settings.

Redshift Multi-AZ supports RPO = 0 meaning data is guaranteed to be current and up to date in the event of a failure. RTO is under a minute.

Redshift Relocation is enabled by default on all new RA3 clusters and serverless endpoints, which allows a data warehouse to be re-started in another AZ in the event of a large-scale outage, without any data loss or additional cost. While using Relocate is free, it is a best-effort approach subject to resource availability in the AZ being recovered in. RTO can be between 10 and 60 minutes.

What happens if a table in my local storage has the same name as an external table? Just like with local tables, you can use the schema name to pick exactly which one you mean by using schema_name.table_name in your query.

Amazon Redshift ML allows you to leverage your data in Amazon Redshift with Amazon SageMaker, a fully managed ML service. Amazon Redshift supports both unsupervised learning (K-Means) and supervised learning (Autopilot, XGBoost, MLP algorithms).

Data warehouse data is stored in a separate storage tier Redshift Managed Storage (RMS). RMS provides the ability to scale your storage to petabytes using Amazon S3 storage.

A compute node is partitioned into slices. Each slice is allocated a portion of the node's memory and disk space, where it processes a portion of the workload assigned to the node.

The leader node manages distributing data to the slices and apportions the workload for any queries or other database operations to the slices. The slices then work in parallel to complete the operation.

Amazon Redshift is based on PostgreSQL.

DynamoDB is directly integrated with Amazon Redshift via a single COPY command.

EMR-Redshift integration

Kinesis Data Firehose to Redshift via S3 limitless concurrency; horizontal scaling

Specify sort key – rows will be sorted and stored. Compound sort key also possible.

COPY from S3 requires a manifest file which lists all data files to load; COPY can parallelize execution.

Redshift streaming ingestion (from MSK or Kinesis)

COPY command for external data; if data is already in redshift, you can use INSERT into or CREATE TABLE AS

SQL JOIN is a clause that is used for combining specific fields from two or more tables based on the common columns available. Joins are used to combine rows from multiple tables.

You only use the INSERT command when you need to move data or a subset of data from one table into another. With S3, you have to use COPY, and not INSERT.

The VACUUM command simply re-sorts rows and reclaims space.

Amazon Redshift workload management (WLM) enables users to flexibly manage priorities within workloads so that short, fast-running queries won't get stuck in queues behind long-running queries.

Automatic WLM or manual WLM.

Elastic resize takes a cluster snapshot and the cluster is temporarily unavailable while elastic resize migrates cluster metadata.

For a Redshift destination, streaming data is delivered to an S3 bucket first. Kinesis Data Firehose then issues an Amazon Redshift COPY command to load data from your S3 bucket to your Amazon Redshift cluster.

Storing the data in S3 and then querying the data via Athena would not facilitate the execution of complex queries in the fastest possible time since Redshift has much better performance than Athena for complex analytical queries.

Data Warehouse includes historical data, aggregated data and summarized data. Unrequired data are removed. DB contains detailed data; DB query is slow and affects all transactions. DWH is designed for faster queries.

Dense storage (DS) node type – for large data workloads; Use hard disk drive (HDD) storage. Dense Compute (DC) node types – optimized for performance-intensive (I/O) workloads. Uses SSD storage.

Enhanced VPC routing: When you use Amazon Redshift enhanced VPC routing, Amazon Redshift forces all COPY and UNLOAD traffic between your cluster and your data repositories through your virtual private cloud (VPC) based on the Amazon VPC service. In this case, you can also use VPC flow logs to monitor COPY and UNLOAD traffic.

Redshift streaming ingestion from KDS and MSK; Redshift ML uses Amazon SageMaker.

Redshift data sharing: With Amazon Redshift data sharing, you can securely share read access to live data across Amazon Redshift clusters, workgroups, AWS accounts, and AWS Regions without manually moving or copying the data.

Automated snapshots are enabled by default when you create a cluster. Manual snapshots are retained indefinitely.

You can configure your cluster to copy snapshots to only one destination AWS Region at a time.

CloudWatch alarm - average percentage of disk space that is used across all of the nodes in your cluster, referred to as the *default disk space alarm*.

Snapshots created from encrypted cluster are encrypted. Security group is mandatory for Redshift.

You can reserve instances by committing to using Redshift for a 1 or 3-year term and save costs.

Enabling the concurrency scaling option in Redshift doesn't guarantee that you will fetch the most recent data. This feature is only used to support virtually unlimited concurrent users and concurrent queries, with consistently fast query performance.

If your files are too small (generally less than 128 MB), the execution engine might be spending additional time with the overhead of opening S3 files, listing directories, getting object metadata, setting up data transfer, reading file headers, and so on. Hence, you can optimize the file sizes by merging smaller files into larger objects in S3.

Query on redshift is faster than query in RDS read replica. Using a read replica on RDS is much costlier than S3. It is possible to query the entire data in S3 using Athena, however, it will not be able to match the high performance offered by Redshift to query the last six months of data.

You cannot set a direct cross-Region snapshot in Redshift. You need to configure an automated snapshot in the same Region as the Redshift cluster and then you can configure Amazon Redshift to automatically copy snapshots (automated or manual) for a cluster to another AWS Region.

Keeping 12 months of data in a Redshift cluster costs more than just unloading the infrequently used data to Amazon S3.

The COPY command leverages the Amazon Redshift massively parallel processing (MPP) architecture to read and load data in parallel from files in an Amazon S3 bucket. Single COPY command is sufficient for copying multiple files.

AWS recommends that you efficiently update and insert new data and load data into a staging table first. Use temporary staging tables to hold the data for transformation. These tables are automatically dropped after the ETL session is complete.

You can use a manifest to ensure that the COPY command loads all of the required files, and only the required files, for a data load. This can also be used to load files from different buckets or files that do not share the same prefix. Instead of supplying an object path for the COPY command, you supply the name of a JSON-formatted text file that explicitly lists the files to be loaded.

The COPY command loads the data in parallel from multiple files, dividing the workload among the nodes in your cluster. When you load all the data from a single large file, Amazon Redshift is forced to perform a serialized load, which is much slower. Split your load data files so that the files are about equal size, between 1 MB and 1 GB after compression. For optimum parallelism, the ideal size is between 1 MB and 125 MB after compression. The number of files should be a multiple of the number of slices in your cluster. Files roughly of the same size.

DISTSTYLE KEY is used for rows that are distributed according to the values in one column. The leader node places matching values on the same node slice. If you distribute a pair of tables on the joining keys, the leader node collocates the rows on the slices according to the values in the joining columns so that matching values from the common columns are physically stored together.

DISTKEY for the 2 tables (which are joining / merging) should be similar.

One can follow this general process to load data from Amazon S3 to Redshift: a) Split your data into multiple files. B) Upload your files to Amazon S3.   C) Run a COPY command to load the table. D) Verify that the data was loaded correctly.

Apache Parquet is an open-source columnar storage format that is 2x faster to unload and takes up 6x less storage in S3 as compared to other text formats. We can now COPY Apache Parquet and Apache ORC file formats from S3 to your Redshift cluster.

Using AWS Glue, one can configure and run a job to transform CSV data to Parquet. Parquet is a columnar format that is well suited for AWS analytics services like Amazon Athena and Amazon Redshift Spectrum.

You have to split your data into multiple files to allow parallelization and use MPP architecture of Redshift.

Amazon Redshift is the best service to use when you need to perform complex queries on massive collections of structured and semi-structured data and get fast performance.

To save on costs, you have to unload the historical data to Amazon S3 from Redshift. Then use Redshift Spectrum to access the data.

Redshift RA3 nodes – compute and storage can scale independently.

A (redshift) cluster connection string is just used to connect to your cluster with a SQL client tool.

If you do S3 CRR or run 2 Redshift clusters with same data, it is additional cost and most likely wrong answer.

Redshift - Use elastic resize to change the node type, number of nodes, or both. If you only change the number of nodes, then queries are temporarily paused and connections are held open if possible. During the resize operation, the cluster is read-only. Typically, elastic resize takes 10–15 minutes. AWS recommends using elastic resize when possible.

Classic resize takes 2 hours to 2 days or longer, depending on your data's size. Can change node type, no of nodes or both. To keep your cluster available during a classic resize, you can first make a copy of an existing cluster, then resize the new cluster.

Diststyle even – round robin. No joins planned. Diststyle key – distribute according to a column value. Joins maybe planned. Diststyle ALL – full copy on all slices. relatively slow-moving tables. Multiple copies. More storage needed.

DISTSTYLE ALL is more appropriate when it is not updated frequently. You should apply DISTSTYLE EVEN to the table if it is frequently changing data. Apply the DISTSTYLE KEY to the table where you want to run query. Use the field (based on which you will do query) as the DISTKEY.

Amazon Redshift automatically integrates with AWS KMS but not with an HSM. When you use an HSM, you must use client and server certificates to configure a trusted connection between Amazon Redshift and your HSM.

To go from unencrypted to encrypted using HSM, you have to create new cluster and migrate to it. But for KMS encryption, you can just modify the existing cluster.

Key rotation can be done in the HSM automatically.

Amazon Redshift database users are named user accounts that can connect to a database. Groups are collections of users that can be collectively assigned privileges for easier security maintenance.

In Amazon Redshift, the COPY command loads data into a table from data files or an Amazon DynamoDB table. The files can be located in an S3 bucket, an EMR cluster, or a remote host accessed using a Secure Shell (SSH) connection.

COPY fails to load data to Redshift if the CSV file uses carriage returns ("\\r", "^M", or "0x0D" in hexadecimal) as a line terminator. Because Amazon Redshift doesn't recognize carriage returns as line terminators, the file is parsed as one line. When the COPY command has the IGNOREHEADER parameter set to a non-zero number, Redshift skips the first line, and therefore, the entire file.

You can create Amazon Redshift external tables by defining the structure for files and registering them as tables in the AWS Glue Data Catalog. The external table statement defines the table columns, the format of your data files, and the location of your data in S3.

The VACUUM process re-sorts rows and reclaims space in either a specified table or all tables in the current database in Amazon Redshift. BOOST option allocates additional resources to VACUUM, such as available memory and disk space.

VCUUM is slow if there is a high percentage of unsorted data or Large table with too many columns.

You can efficiently update and insert new data by loading your data into a staging table first. Amazon Redshift doesn't support a single merge statement (update or insert, also known as an upsert) to insert and update data from a single data source. However, you can effectively perform a merge operation. To do so, load your data into a staging table and then join the staging table with your target table for an UPDATE statement and an INSERT statement.

CREATE TABLE LIKE statement, the staging table will inherit the distribution key from the parent table. If you use a CREATE TABLE AS statement, the new table does not inherit the distribution key.

If an automated snapshot is enabled, Amazon Redshift will take a snapshot every eight hours or following every 5 GB per node of data changes. A manual snapshot can be taken at any time. Manual snapshots are retained indefinitely, even after you delete your cluster.

When you partition your data, you can restrict the amount of data that Redshift Spectrum scans by filtering on the partition key. You can partition your data by any key. A common practice is to partition the data based on time.

COPY command vs UNLOAD command (redshift): UNLOAD command allows Redshift users to export data from a SQL query run in the data warehouse into an Amazon S3 bucket – essentially doing the reverse of the COPY command.

Redshift – columnar, compressed, massively parallel processing, use ML.

A prime goal of data distribution is to minimize data movement when queries run. Therefore, the rows that participate in joins or aggregates has to be collocated on the nodes with their joining rows in other tables.

If the above is not possible, then you can improve query performance significantly by distributing the entire table to all of the nodes. ie, use 'All' distribution style.

Your fact table can have only one distribution key, not multiple. Any tables that join on another key aren't collocated with the fact table.

You can now control access to columns without having to implement view-based access control or use another system. Table level access control is also possible.

Amazon Redshift supports column-level access control for data in Redshift. Customers can use column-level GRANT and REVOKE statements to help meet their security and compliance needs. GRANT SQL command.

Create external schema spectrum, create external table – These will enable Redshift Spectrum engine to read the data in S3.

Use of a WLM queue would not fix the performance problem.

When Redshift performance has deteriorated, you can do:

- A separation of the data into a set of smaller time-series tables is a standard best practice that will improve query performance because the queries will run against smaller datasets.
- Additionally, by using a time-series table, the company can avoid the added cost of row deletion because a DROP TABLE operation is more efficient than a mass DELETE process.

The functionality of the reverse compound sort key does not exist in Amazon Redshift. In addition, the columns in a compound sort key that appear after a high-cardinality column have minimal benefit. Compound sort keys should be organized in order of lowest cardinality to highest cardinality.

When you are doing bulk ETL for Redshift, AWS recommends that you use temporary staging tables to hold the data for transformation. These tables are automatically dropped after the ETL session is complete.

The automatic vacuum operations can pause if the cluster experiences a period of high load.

Redshift - column-level access control, row-level security

When you enable logging on your redshift cluster, you are enabling logging to Amazon S3 only. Currently, you can only use SSE-S3 encryption for audit logging.

Redshift Spectrum

Redshift Spectrum is a feature of Redshift that lets you run queries against your data lake in S3, with no data loading or ETL required.

You can use Amazon Redshift Spectrum to query the entire historical data in S3 which would be a cost-effective solution to facilitate analysis of the historical data.

AWS manages a fleet of thousands of Redshift Spectrum nodes spread across multiple Availability Zones. For historical data query, redshift is better than RDS. Using a read replica on RDS to store and analyze the entire historical data is much costlier than storing the data on S3 and enable analysis using Redshift spectrum.

Use Redshift Spectrum to create Redshift cluster tables pointing to the underlying data in S3.

Redshift Spectrum is also highly concurrent—you can access your Amazon S3 data from any number of Amazon Redshift clusters. To access this data on S3 via Redshift Spectrum, you need to create an external schema in Amazon Redshift.

Under the hood, AWS manages a fleet of thousands of Redshift Spectrum nodes spread across multiple Availability Zones.

Amazon Redshift Spectrum supports column level access control for data stored in Amazon S3 and managed by AWS Lake Formation. Permission to access the columns should be given by DataLake, not by S3.

S3

It is difficult to aggregate data from different sources by row. The columnar data format is faster at aggregating large volumes of data.

Your application can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per prefix in a bucket. There are no limits to the number of prefixes in a bucket.

S3 One Zone-IA and S3 Standard has the same read performance.

Using the Range HTTP header in a GET Object request, you can fetch a byte-range from an object, transferring only the specified portion. You can use concurrent connections to Amazon S3 to fetch different byte ranges from within the same object.

With Amazon S3 Select, you can scan a subset of an object by specifying a range of bytes to query using the ScanRange parameter.

By default, an S3 object is owned by the AWS account that uploaded it. So, the S3 bucket owner will not implicitly have access to the objects written by the Redshift cluster. Because the Amazon Redshift data files from the UNLOAD command were put into your bucket by another account, you (the bucket owner) don't have default permission to access those files.

1 PUT payload unit – 25kb

No chaining of bucket replication allowed. A is replicated to B. B is replicated to C. But A won't be replicated to C.

S3 lifecylce rules can be created based on certain prefix or object tags. You can transition the "noncurrent versions" of the object to Standard IA.

If you use SSE-KMS, you may be impacted by the KMS limits. ie, no of API calls.

S3 Object Lambda - Use AWS Lambda Functions to change the object before it is retrieved by the caller application. Linked with Access Point. Each S3 access point has AP policy and DNS name.

Byte-range fetch: Using the Range HTTP header in a GET Object request, you can fetch a byte-range from an object, transferring only the specified portion. With Amazon S3 Select, you can scan a subset of an object by specifying a range of bytes to query using the ScanRange parameter.

Archive objects queried by S3 Glacier Select must be formatted as uncompressed CSV.

S3 Select costs less than Athena.

If you export assets to a signed URL, your AWS account is charged for data transfer costs from Amazon S3 to the Internet.

Amazon S3 is strongly consistent for all GET, PUT and LIST operations. S3 provides Eventual Consistency for overwrite PUTS and DELETES.

By default, S3 stores object replicas using the same storage class as the source object. You can also specify a different storage class for the replicas.

S3 standard IA has a minimum storage duration charge of 30 days, so S3 std may be better in some use cases.

Instead of loading an entire large Amazon S3 object, load only what is needed using S3 Select. Keeping the data in Amazon S3 avoids loading the large dataset into HDFS.

S3 Glacier vault and Vault Lock policy

Security

You can use key alias so that key rotation won't impact.

AWS IoT is attached to X.509 certificates or Cognito Identities.

EMR EC2 Security Groups - One for master node; Another one for cluster node (core node or task node); Kerberos authentication (provide authentication from Active Directory) for EMR.

Federated identity pools for public application. Web Identity Federation is an alternative to using Cognito but AWS recommends against it.

AWS Variables:
${aws:username } : to restrict users to tables / buckets
${aws:principaltype } : account, user, federated, or assumed role
${aws:PrincipalTag /department} : to restrict using Tags
${aws:FederatedProvider } : which IdP was used for the user (Cognito, Amazon..)
${www.amazon.com:user_id } , cognito identity.amazonaws.com:sub }
${saml:sub }, sts:ExternalId
IAM policies don't help with in-database security.

CloudTrail - The default UI only shows "Create", "Modify" or "Delete" events. Create CloudTrail Trail: to get a detailed list of all the events you choose.

VPC endpoint also known as VPC PrivateLink.

EMR components communicate using SSL

Kerberos authentication

For security, you can use SSE-S3 for S3 or EMRFS. EBS encryption is the recommended mechanism to encrypt local and root volumes, and the company can provide the in-transit encryption artifacts through S3 zipped files.

EMR - encrypting data in transit using a security configuration: You can manually create PEM certificates, include them in a .zip file, and then reference the .zip file in Amazon S3.

You cannot use DynamoDB to specify the artifacts for encryption of data in transit. There are two ways to specify the artifacts for encryption of data in transit: 1) Upload the artifacts to Amazon S3. 2) Implement a custom certificate provider as a Java class.

LUKS (Linux Unified Key Setup) encryption – If you choose LUKS for EBS volumes, the LUKS encryption applies only to attached storage volumes, not to the root device volume.

EMRFS encryption allowed – SSE-S3, SSE-KMS, CSE-KMS, CSE-Custom.

A company can use per bucket encryption overrides to specify different encryption methods for individual buckets. This happens if you use separate encryption keys for buckets.

AWS Glue S3 encryption – only SSE-S3 and SSE-KMS?

EMR – EC2 require IAM roles.

Apache Ranger: Centralized Authorization (RBAC Role Based Access) setup on external EC2.

Redshift: The COPY command supports the different types of S3 encryption.

Spark

Spark Dataframes

Resilient Distributed Datasets (RDD); Spark is 10x faster than MapReduce. DAG (Directed Acyclic Graph for each RDD)

Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams.

SparkML – Machine Learning Algorithm Library

GraphX: page ranking, library of graph algorithms

Hive on Spark, Hive on Tez, Hive on MapReduce

Apache Spark - Faster than MapReduce. Can be deployed through Apache Mesos

Spark Metrics - Monitor memory usage with JvmHeapUsed metric. Monitor load using Amazon EMR TotalLoad metric.

Spark is a distributed in-memory computing engine. Compared to MapReduce, which works in batch mode, Spark's computation model works in interactive mode, i.e., assembles the data in memory before processing it and is therefore very suitable for Machine Learning processing.

One of the speed advantages of Apache Spark comes from loading data into immutable dataframes, which can be accessed repeatedly in memory. Spark DataFrames organizes distributed data into columns. This makes summaries and aggregates much quicker to calculate.

Spark UI and YARN Timeline Service to simplify debugging.

When you run Spark on Amazon EMR, you can use EMRFS to directly access your data in Amazon S3. Spark supports multiple interactive query modules such as SparkSQL.

Anti Patterns

Athena does not support querying the data in GLACIER. Athena does not support user-defined functions(UDF), INSERT INTO statements, stored procedures.

Athena is only a query service. It is not used for loading data into the cluster.

You can’t integrate API gateway with Athena. OpenSearch can be integrated with API Gateway. WorkLoad Management (WLM) is available only in Redshift and not in Athena.

S3 Select is not suitable for analytic workloads. You should use Amazon Athena instead.

Using a row-based data format is not suitable for aggregating large amounts of data.

EMR is not suitable for small data and real time data processing.

Presto is not optimal for batch processing jobs.

The S3DistCp tool is used to copy data from S3 to EMR (HDFS) and not to Redshift.

Amazon Sagemaker Jupyter notebook cannot directly read data from Amazon Kinesis Data Firehose.

Amazon SQS does not support real-time streaming of data.

OpenSearch does not have a direct integration with Amazon QuickSight. You can’t use EMR as a data source for Amazon QuickSight.

Lambda is not a supported source for QuickSight. CloudWatch metrics data ingestion is also not supported by Quicksight.

AWS DMS doesn't support Amazon OpenSearch as a data source.

Row-level security doesn't currently work for dates or numeric fields. Works only for textual data. Anomaly detection is not supported for datasets that use row-level security (RLS).

You can’t create Readers using the Standard edition of QuickSight.

Quicksight in one region can access redshift in another region, if security group allows.

EMR resides in single AZ. and Redshift also resides in single AZ.

Amazon Redshift Spectrum external tables are read-only.

Athena can’t convert .csv to parquet.

If you are using spot instance, use EMRFS (not HDFS).

There is no TTL functionality in redshift.

The use of Hive to run one-time queries would not be the best solution because the company would need to launch an EMR cluster for launching Hive.

You cannot use DynamoDB as an external AWS Glue Data Catalog.

DataSync does not support S3 Transfer Acceleration (S3TA) as part of the data migration.

Logstash is a data ingestion tool and is not suitable to develop ML algorithms.

You can't directly copy data from Snowball Edge devices into AWS Glacier. For Snowmobile, you can import your data directly into Glacier.

Pig does not have SQL capability. Pig does not work with Spark; only MapReduce and Tez.

Hive is not suitable for real time processing.

Spark / KCL do not read from Kinesis Data Firehose! Firehose cannot directly write into a DynamoDB.

Kinesis Data Firehose can't convert data to Avro format. (Only Parquet and ORC?)

A single delivery stream can deliver data to only one S3 bucket.

Kinesis Agent cannot write to a Kinesis Firehose for which the delivery stream source is already set as Kinesis Data Streams.

Encryption at rest is available in the Enterprise edition only (Quicksight).

Redshift Serverless must be accessed within a VPC.

The Amazon S3 lifecycle policy is based on an object's age and not on the date it was last accessed.

S3 analytics won't work for OneZone-IA! SSE-C is not available for use with EMRFS.

LUKS applies only to storage volume, and not root volume. Similarly, if you encrypt EMRFS, it won't encrypt root volume; will encrypt only storage volume.

Users can only use DynamoDB as a checkpointing table for the KCL. Not RDS.

Compression formats

Hive Supports snappy and gzip compression.

With Hive, use serializer/de-serializer (SerDe) to change format.

Redshift/Redshift Spectrum - Gzip and snappy supported

ORC/Parquet is preferred over CSV when Redshift spectrum reads data.

Amazon S3 Select works on objects stored in CSV, JSON, or Parquet format. It also works with objects compressed with GZIP or BZIP2.

S3DistCp does not support concatenation for Parquet files.

Glue supports JSON, CSV, Avro, Parquet only.

Athena can't convert .csv to parquet. For Athena, Parquet is better than compressed CSV.

We can now COPY Parquet and ORC file formats from S3 to your Redshift cluster.

Athena supports CSV, JSON, Apache ORC, Apache Parquet, and Apache Avro. Athena supports most of the formats and compression techniques.

You can add partitions created by Data Firehose using ALTER TABLE DDL statements.

Athena supports UDF (???)

QuickSight does not support parquet format while reading the data from Amazon S3.

Comparisons

DDL vs DML

Amazon Redshift is more cost-effective than Amazon Athena for frequently accessed reports.

Building an Athena solution is faster than redshift solution.

AWS recommends using Amazon EMR if you use custom code to process and analyze extremely large datasets with big data processing frameworks.

Amazon Redshift is the best service to use when performing complex queries on massive collections of structured and semi-structured data and getting fast performance.

Use EMR for unstructured/semi structured data. Use Redshift for structured data.

Redshift - Using multiple COPY commands would result in a slower performance. Use single COPY command.

Taking advantage of Redshift's compute power to run join queries using Redshift Spectrum would be a better option than Athena.

Redshift beats Athena in terms of performance, but Athena is more flexible and portable, you can create a solution a lot faster by using Amazon Athena.

The maximum record size of an Amazon MSK is 100 MB. The maximum size of a record sent to Kinesis Data Firehose or Kinesis Data streams is 1MB. So if your file size is more than that, use MSK.

The End