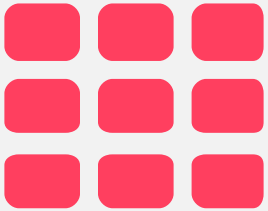

aws
certified
Solutions
Architect
Professional
DOLFINed



ANALYTICS, DATA STREAMING, AND INTERNET OF THINGS

AMAZON REDSHIFT

Amazon REDSHIFT

Introduction

AWS Services : Redshift

- Redshift, is an AWS, fully managed, **Petabyte scale** data warehouse service in the cloud
- Amazon Redshift gives you fast querying capabilities **over structured data** using familiar SQL-based clients and business intelligence (BI) tools using standard ODBC and JDBC connections.
- Queries are distributed and parallelized across multiple physical resources.
- Amazon Redshift uses **replication and continuous backups** to enhance availability and improve data durability and can automatically recover from component and node failures.
- Redshift is a SQL based data warehouse used for analytics applications (Analytics DB)
  - Example use cases: Sales Reporting, Health Care analytics
  - It is suited for OLAP-based use cases (On Line Analytics Processing)
  - **Can store huge amount of data** (a database), **but can't ingest huge amounts of data in real time** (not like what Kinesis can do)
- Redshift can:
  - Fully recover from a node or component failure
  - It automatically patches and performs data backup
  - Backups can be stored for a user defined retention period
  - Is 10 times faster than traditional SQL RDBMS

source: aws.amazon.co

Redshift Performance

- Redshift uses a variety of features to achieve much faster performance compared to traditional SQL DBs
  - Columnar Data Storage:
    - Data is stored sequentially in columns instead of rows
  - Advanced Compression
    - Data is stored sequentially in columns which allows for much better compression
    - Redshift automatically selects compression scheme
  - Massive Parallel Processing (MPP) : Data and query loads are distributed across all nodes

Redshift – Data Security

- RedShift supports encryption of data “at rest” using hardware accelerated AES-256 bits
  - By default, AWS Redshift takes care of encryption key management
  - You can choose to manage your own keys through HSM (Hardware Security Modules), or AWS KMS (Key Management Service)
- It also supports SSL Encryption, in-transit, between client applications and Redshift data warehouse cluster
- You can’t have direct access to your AWS Redshift cluster nodes, however, you can through the applications themselves

Redshift

- No upfront commitment, you can start small and grow as required
  - You can start with a single, 160GB, Redshift data warehouse node

- For a multi-node deployment (Cluster), you need a leader node and compute node(s)
  - The leader node manages client connections and receives queries
  - The compute nodes store data and perform queries and computations
  - You can have up to 128 compute nodes in a cluster

Redshift – Backup Retention

- Amazon Redshift automatically patches and backs up (Snapshot) your data warehouse, storing the backups for a user-defined retention period in AWS S3.
  - It keeps the backup by default for one day (24 hours) but you can configure it from 0 to 35 days
  - You have access to these automated snapshots during the retention period

- If you delete the cluster,
  - You can choose to have a final snapshot to use later

- Manual backups are not deleted automatically, if you do not manually delete them, you will be charged standard S3 storage rates

- AWS Redshift currently supports only one AZ (no Multi-AZ option)

- You can restore from your backup to a new Redshift cluster in the same or a different AZ
  - This is helpful in case the AZ hosting your cluster fails

Redshift - Availability and Durability

- Redshift automatically **replicates all** your data within your data warehouse cluster
  - To other drives within the cluster to maintain copies of your original data

- Redshift always keeps three copies of your data:
  - The original one
  - A replica on compute nodes (within the cluster)
  - A backup copy on S3

- **Cross Region Replication**
  - Redshift can asynchronously replicate your snapshots to S3 in another region for DR

source:www.amazon.com

Redshift Spectrum

- Amazon Redshift Spectrum allows you to directly run SQL queries against exabytes of data in Amazon S3 (instead of loading the data to the Cluster).
- With Redshift Spectrum, you can run multiple Amazon Redshift clusters accessing the same data in Amazon S3.
- You can use one cluster for standard reporting and another for data science queries. Your marketing team can use their own clusters different from your operations team.
- Redshift Spectrum automatically distributes the execution of your query to several Redshift Spectrum workers out of a shared resource pool to read and process data from Amazon S3, and pulls results back into your Amazon Redshift cluster for any remaining processing.
- You are charged for the number of bytes scanned by Redshift Spectrum, rounded up to the next megabyte, with a 10MB minimum per query.
- Redshift Spectrum supports Amazon S3's Server-Side Encryption (SSE) using your account's default key managed used by the AWS Key Management Service (KMS).

Redshift Enhanced VPC Routing

- Use Redshift's Enhanced VPC Routing to force all the COPY and UNLOAD traffic to go through the VPC privately through endpoints.

- This way it becomes possible to tightly manage the flow of data between Amazon Redshift clusters and all of your data sources.

- You can also add a policy to your VPC endpoint to restrict unloading data only to a specific S3 bucket in your account and monitor all COPY and UNLOAD traffic using VPC flow logs.

Redshift – Billing

You pay for:

- **Compute node hours** – Compute node hours are the total number of hours you run across all your compute nodes for the billing period. You are billed for 1 unit per node per hour. You are not charged for the leader node

- **Backup Storage** – Backup storage is the S3 storage associated with your automated and manual snapshots for your data warehouse.
  - Be careful if you change your backup retention period (extra charges)

- **Data transfer** – There is no data transfer charge for data transferred to, or from, Amazon Redshift and Amazon S3 within the same AWS Region.
  - For all other data transfers into and out of Amazon Redshift, you will be billed at standard AWS data transfer rates.

source:www.amazon.com

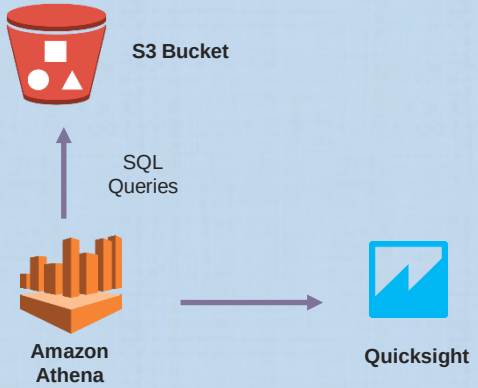
AMAZON ATHENA

Amazon Athena

What it is & how it works

Amazon Athena – What is it?

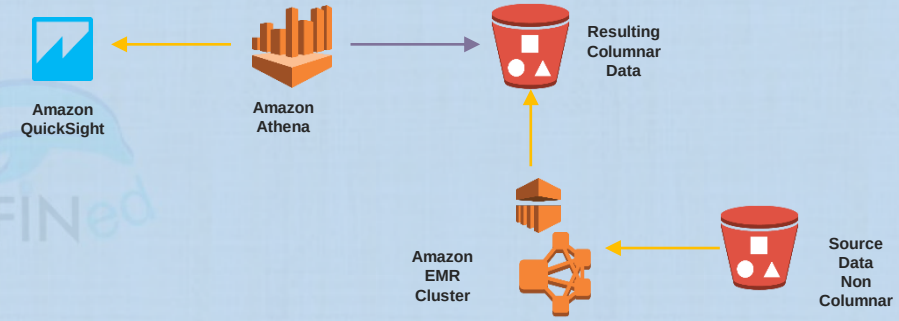
- Amazon Athena is an interactive, **serverless,** query service that makes it easy to analyze data directly in Amazon S3 using standard SQL.
  - It is simply pointing Athena at data stored in Amazon S3 and begin using standard SQL to run ad-hoc queries, without the need to aggregate or load the data into Athena and get results in seconds.
  - Athena integrates with Amazon QuickSight for data visualization.
  - Charge is based on the queries used.
  - You create Schema (tables and data sets) in Athena and apply it to data in S3 when querying the S3 data
- Athena scales automatically
  - Queries are executed in parallel, hence, Results are fast, even with large datasets and complex queries.
- Athena helps to analyze unstructured, semi-structured, and structured data stored in Amazon S3.
  - Examples include **CSV**, **JSON**, or columnar data formats such as Apache **Parquet** and Apache **ORC**.



Amazon Athena – Columnar Storage format

- Apache Parquet and ORC are columnar storage formats that are optimized for fast retrieval of data and used in AWS analytical applications.

- The Amazon Athena query performance improves if the data is converted into open source columnar formats, such as Apache Parquet or ORC.
  - This can be done to existing Amazon S3 data sources by creating a cluster in Amazon EMR and converting it using Hive.



Amazon Athena – Query Results

- Athena stores query results in Amazon S3. Each query that runs has:
  - A results file stored automatically in a CSV format (*.csv), and
  - An Athena metadata file (*.csv.metadata).

- Athena retains query history for 45 days.
  - If it is required to keep it beyond 45 days, a code can be written to read the data and store to S3.

Amazon Athena – Encryption Options

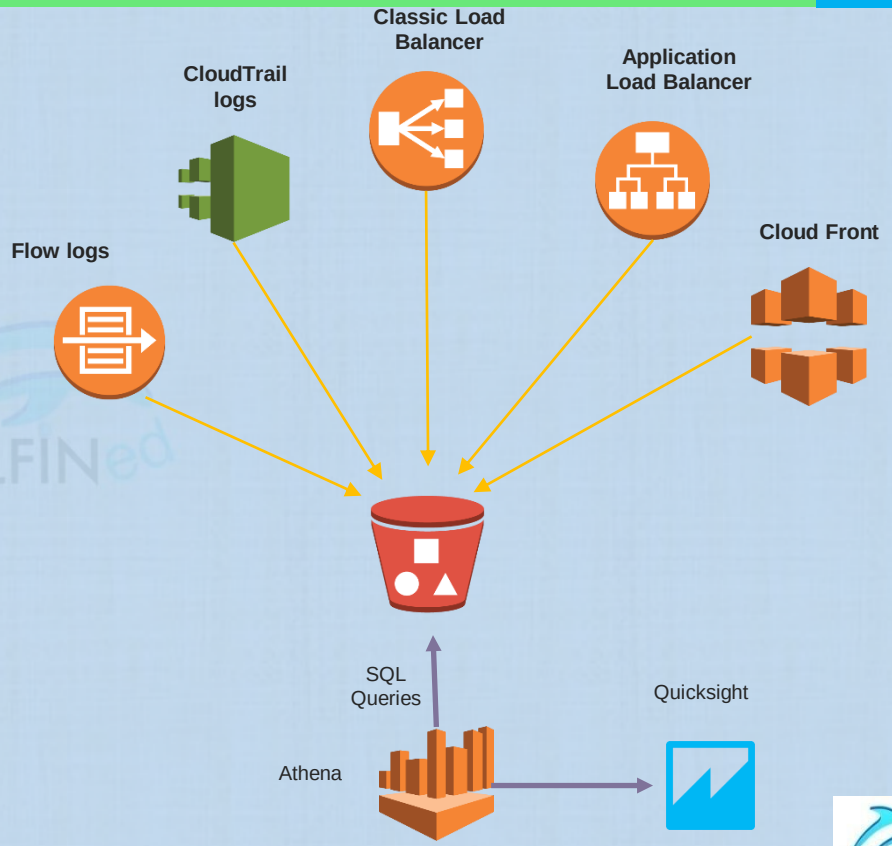
- Encryption options enable the encryption of query result files in Amazon S3 and query data encrypted in Amazon S3.
  - Users must have the appropriate permissions to access the Amazon S3 locations and decrypt files.

- Queries can be run in Athena to query encrypted data in Amazon S3 by indicating data encryption when creating a table.
  - Query results can be encrypted in Amazon S3, and the data in the AWS Glue Data Catalog.

Amazon Athena – Querying AWS Services Logs

Amazon Athena can be used to query popular datasets, such as:

- AWS CloudTrail logs,
- Amazon CloudFront logs,
- Classic Load Balancer logs,
- Application Load Balancer logs,
- Network Load Balancer logs,
- Amazon VPC flow logs.



AMAZON KINESIS

AWS Services : Kinesis

Streaming of Data

Streaming of data:

- Is the data that is generated and sent continuously from a large number (1000s or hundreds of 1000s) of data sources, where data is sent in small sizes (usually in Kbytes or MBytes)

- Kinesis, is a platform for streaming data on AWS (used for IoT and Bigdata Analytics)
  - It offers powerful services to make it easy to load and analyze streaming data
  - It facilitates the way to build custom streaming data Apps for specialized needs
  - Kinesis is an AWS managed streaming data service(s)

- Kinesis can continuously capture and store Terabytes of data per hour from 100s of thousands of sources

- IT Offers three **managed services**, there are:
  - Kinesis Streams, Kinesis Firehose, Kinesis Analytics

AWS Kinesis

Sources of data (examples)

- IoT sensors data
- Log files from customers of mobile and web apps
- eCommerce purchases
- In-game player activities
- Social media networks
- Financial trading floors and stock markets
- Telemetry

Amazon Kinesis

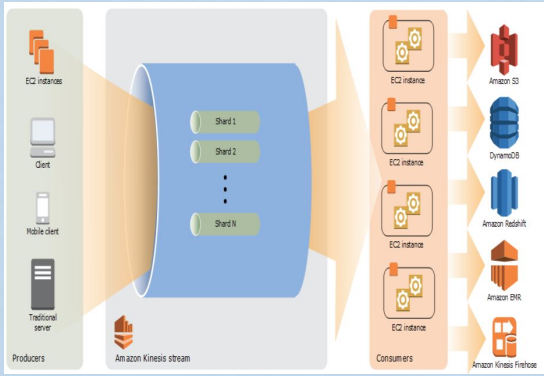
Kinesis Data Streams

AWS Kinesis Streams

Use Amazon Kinesis Streams to collect and process large stream of data records in real time.

Custom data-processing applications, known as ***Amazon Kinesis Streams applications*** read data from a *Kinesis stream* as data records.

- These applications use the Kinesis Client Library and run on Amazon EC2 instances.
- The processed records can be:
  - Sent to dashboards,
  - Used to generate alerts,
  - Used to Dynamically change pricing and advertising strategies,
  - Saved to a variety of other AWS services (EMR, DynamoDB, or Redshift)
- Kinesis can make this huge streams of data available for processing By AWS services or customer applications in less than a second



souce:aws.amazon.com

AWS Kinesis Streams

- Amazon Kinesis Streams manages the infrastructure, storage, networking, and configuration needed to stream your data at the level of your data throughput.
  - You do not need to worry about provisioning, deployment, ongoing-maintenance of hardware, software, or other services for your data streams.

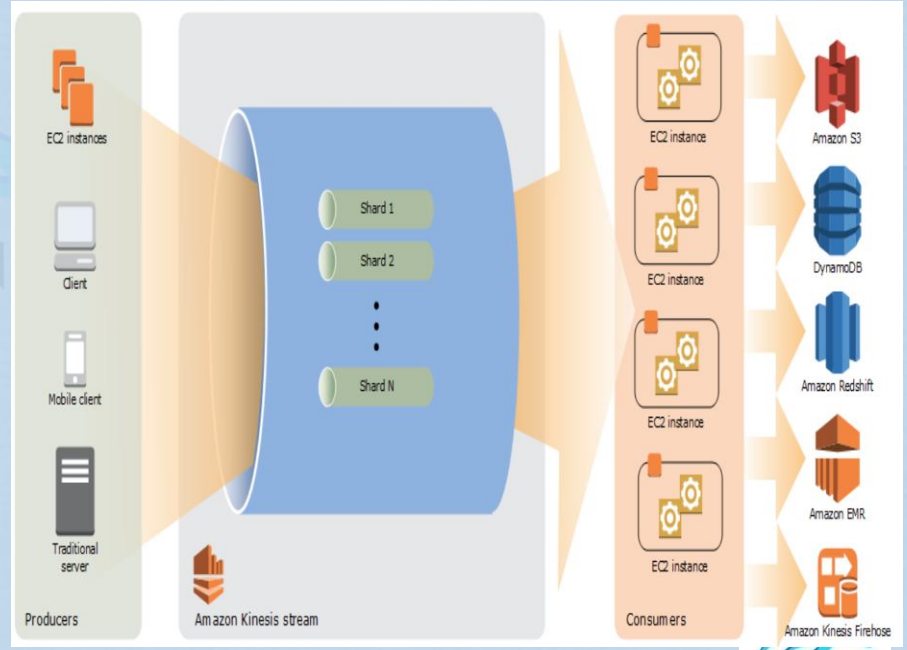
- Amazon Kinesis Streams synchronously replicates data across three availability zones, providing high availability and data durability.

- Kinesis Streams use cases:
  - Accelerate log & data feed intakes
  - Real time metric and reporting analytics
  - Complex stream processing (successive stages of stream processing)

souce:aws.amazon.com

AWS Kinesis Streams

- **Producers:** put records into Amazon Kinesis Streams
- **Consumers:** get records from Amazon Kinesis Streams and process them
- **Kinesis Streams Application:** is a consumer of a stream that commonly runs on a fleet of EC2 instances
- **Shard:** is a uniquely identified group of data records in a stream. A stream is composed of one or more shards.
  - A shard is the base throughput unit of a stream
    - it can take up to 1Mb/s input and 2 Mb/s output
- **Record:** the data unit stored in a Kinesis stream



souce:aws.amazon.com

AWS Kinesis Streams _ Retention Period

- **Retention Period:**
  - The time for which data records will be kept and made accessible to Kinesis streams applications
  - Default is 24 hours (1 day)
  - Can be extended to 7 days (168 hrs) for additional charges

souce:aws.amazon.com

AWS Kinesis Streams - Encryption

- **Server-side encryption**
  - Amazon Kinesis Streams can automatically encrypt sensitive data as a producer enters it into a stream.
  - Kinesis Streams uses KMS master keys for encryption
  - To read from or write to an encrypted stream, producer and consumer applications must have permission to access the master key

- **Note**
  - Using server-side encryption incurs KMS costs

souce:aws.amazon.com

Kinesis Producer & Client Libraries

Kinesis Producer Library:

- An Amazon Kinesis Streams producer is any application that puts user data records into a Kinesis stream (also called *data ingestion*).
- The Kinesis Producer Library (KPL) simplifies producer application development, allowing developers to achieve high write throughput to a Kinesis stream.
- It acts as an intermediary between your producer application code and the Kinesis Streams API actions.

Kinesis Client Library

- You can develop a consumer application for Amazon Kinesis Streams using the Kinesis Client Library (KCL).
  - AWS recommends using the design patterns and code for consumer applications provided by the KCL.
- The Kinesis Client Library (KCL) helps the consumer App process data from a Kinesis stream.
- The KCL takes care of distributed computing tasks, such as load-balancing across multiple instances, responding to instance failures, check pointing processed records, and reacting to resharding.

souce:aws.amazon.com

AWS Kinesis Data Streams – Workers

- At run time, a KCL application instantiates a worker with configuration information, and then uses a record processor to process the data received from a Kinesis stream.

- You can run a KCL application on any number of instances. Multiple instances of the same application coordinate on failures and load-balance dynamically.

- You can also have multiple KCL applications working on the same stream, subject to throughput limits.

souce:aws.amazon.com

Amazon Kinesis

Kinesis Firehose

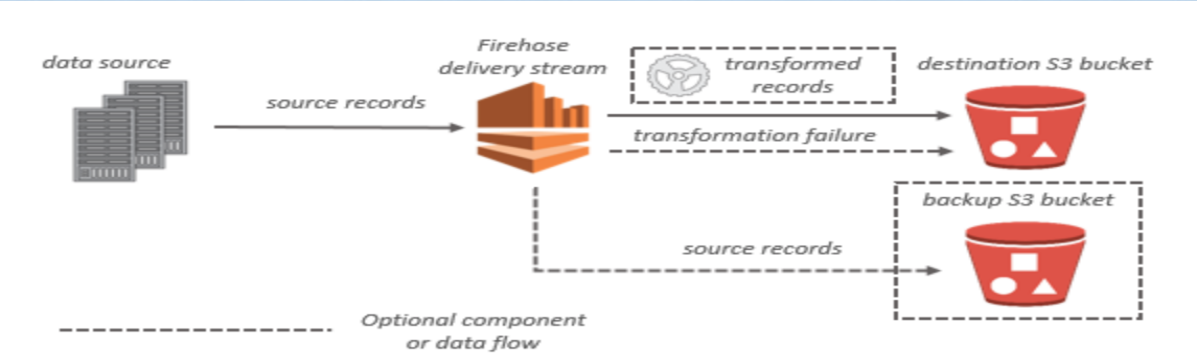
AWS Kinesis Firehose

- Use **Amazon Kinesis Firehose** to deliver real-time streaming data to destinations such as Amazon S3 and Amazon Redshift.
- Is the easiest way to load streaming data into AWS
  - Kinesis streams can be used as the source(s) to Kinesis Firehose
  - You can configure Kinesis Firehose to transform your data before delivering it.
- Amazon Kinesis Firehose is a fully managed service for automatically capturing real-time data stream from producers (sources) and delivering them to destinations such S3 , RedShift, ElasticSearch and Splunk.
- With Kinesis Firehose, you don't need to write applications or manage resources.
- It can batch, compress, and encrypt the data before loading it, minimizing the amount of storage used at the destination and increasing security.
- Amazon Kinesis Data Firehose manages all underlying infrastructure, storage, networking, You do not have to worry about provisioning, deployment, ongoing maintenance.
  - Firehose also scales elastically without requiring any intervention.
- It can batch, compress, and encrypt the data before loading it, minimizing the amount of storage used at the destination and increasing security.
- Amazon Kinesis Data Firehose manages all underlying infrastructure, storage, networking, You do not have to worry about provisioning, deployment, ongoing maintenance.

souce:aws.amazon.com

AWS Kinesis Data Firehose

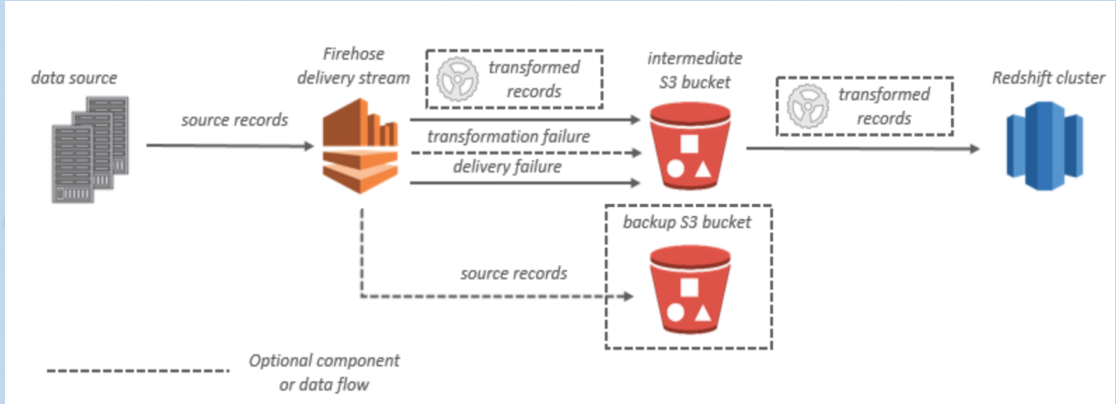
- Firehose also scales elastically without requiring any intervention.
- Amazon Kinesis Data Firehose synchronously replicates data across three facilities in an AWS Region, providing high availability and durability for the data as it is transported to the destinations.
- Each delivery stream stores data records **for up to 24 hours in case** the delivery destination is unavailable.
- Firehose can invoke an AWS Lambda function to transform (ETL) incoming data before delivering it to destinations.
- For AWS S3 destinations, streaming data is delivered to your S3 bucket.
- If data transformation is enabled, you can optionally back up source data to another Amazon S3 bucket.
- ElasticSearch case is similar



souce:aws.amazon.com

AWS Kinesis Firehose

- For AWS Redshift destinations, streaming data is delivered to your S3 bucket first.
  - Kinesis Firehose then issues an Amazon Redshift **COPY** command to load data from your S3 bucket to your Amazon Redshift cluster
  - If data transformation is enabled, you can optionally back up source data to another Amazon S3 bucket



souce:aws.amazon.com

AWS Kinesis Firehose – Data Encryption

- **Server-side encryption**
  - If you have sensitive data, you can enable server-side data encryption when you use Amazon Kinesis Firehose.
    - However, this is only possible if you DO NOT use a Kinesis stream as your data source.
    - When you configure a Kinesis stream as the data source of a Kinesis Firehose delivery stream, the data is stored in the Kinesis stream.
      - Enable Kinesis Streams to encrypt the data in this case.

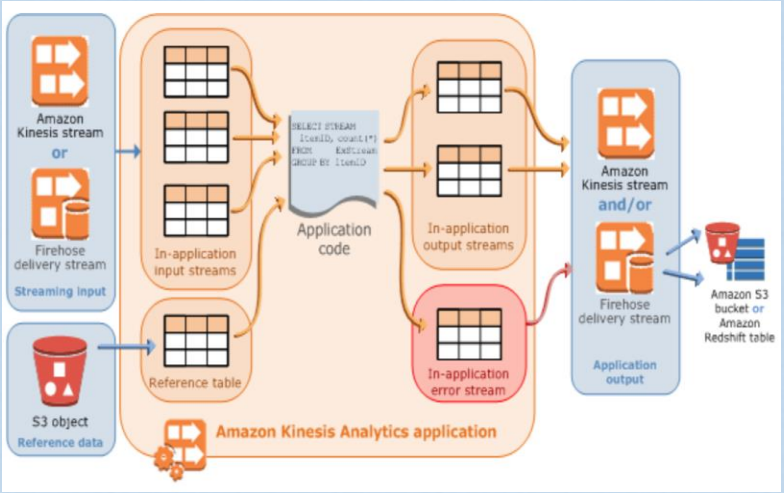
souce:aws.amazon.com

Amazon Kinesis

Kinesis Analytics

AWS Kinesis Analytics

- Use **Amazon Kinesis Analytics** to process and analyze streaming data with standard SQL
- The service enables you to quickly author and run powerful SQL code against streaming sources
- The service supports ingesting data from Amazon Kinesis Streams and Amazon Kinesis Firehose streaming sources.
- You can also configure destinations where you want Amazon Kinesis Analytics to persist the results.
- Amazon Kinesis Analytics supports Amazon Kinesis Firehose (Amazon S3, Amazon Redshift, and Amazon Elasticsearch Service), and Amazon Kinesis Streams as destinations.
- Amazon Kinesis Analytics needs permissions to read records from a streaming source and write application output to the external destinations.
- You use IAM roles to grant these permissions.
- Note here the sources and the fact that the application is SQL code based



souce:aws.amazon.com

AWS Kinesis Analytics

- Amazon Kinesis Analytics enables you to quickly author SQL code that continuously reads, processes, and stores data in near real time.
- Using standard SQL queries on the streaming data, you can construct applications that transform and gain insights into your data.

- Use cases:
  - **Generate time-series analytics** – You can calculate metrics over time windows, and then stream values to Amazon S3 or Amazon Redshift through an Amazon Kinesis Firehose delivery stream.
  - **Feed real-time dashboards** – You can send aggregated and processed streaming data results downstream to feed real-time dashboards.
  - **Create real-time metrics** – You can create custom metrics and triggers for use in real-time monitoring, notifications, and alarms.

souce:aws.amazon.com

AMAZON ELASTIC MAP REDUCE

Amazon Elastic Map Reduce

Introduction

AWS Elastic Map Reduce (EMR)

EMR and Apache Hadoop

- Amazon EMR is a managed cluster platform that simplifies running big data frameworks, **such as Apache Hadoop and Apache Spark, on AWS** to process and analyze vast amounts of data.
- Hadoop uses a distributed processing model called **MapReduce,** which is also designed to process large data sets quickly.
- By using these frameworks and related open-source projects, such as Apache Hive and Apache Pig, you can process data for analytics purposes and business intelligence workloads
- Additionally, you can use Amazon EMR to transform and move large amounts of data into and out of other AWS data stores and databases, such as S3 and DybamoDB
- Amazon EMR is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data.
- Amazon EMR lets you focus on crunching or analyzing your data without having to worry about time-consuming set-up, management or tuning **of Hadoop clusters** or the compute capacity upon which they sit.
- It utilizes a hosted Hadoop framework running on the web-scale infrastructure of Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3).
- Amazon EMR is ideal for problems that necessitate the fast and efficient processing of large amounts of data.

AWS Elastic Map Reduce (EMR)

Introduction

- EMR launches clusters on Amazon EC2 instances.
- EMR leverages **Apache Hadoop** as its distributed data processing engine.
- MapResuce is used primarily in Analytics and Business intelligence purposes.
- **Hadoop** is not a type of database, but rather a software ecosystem that allows for massively parallel computing. It is an enabler of certain types **NoSQL** distributed databases (such as HBase), which can allow for data to be spread across thousands of servers with little reduction in performance.
- Using Amazon EMR, you can instantly provision as much or as little capacity as you like to perform data-intensive tasks for applications such as:
  - Web indexing,
  - Data mining,
  - Log file analysis,
  - Machine learning,
  - Financial analysis,
  - Scientific simulation, and
  - Bioinformatics research.
- You can also use EMR to transform and move large amounts of data into and out of other AWS data stores and databases, such as S3 and DynamoDB.

AWS Elastic Map Reduce (EMR)

EMR quick Introduction

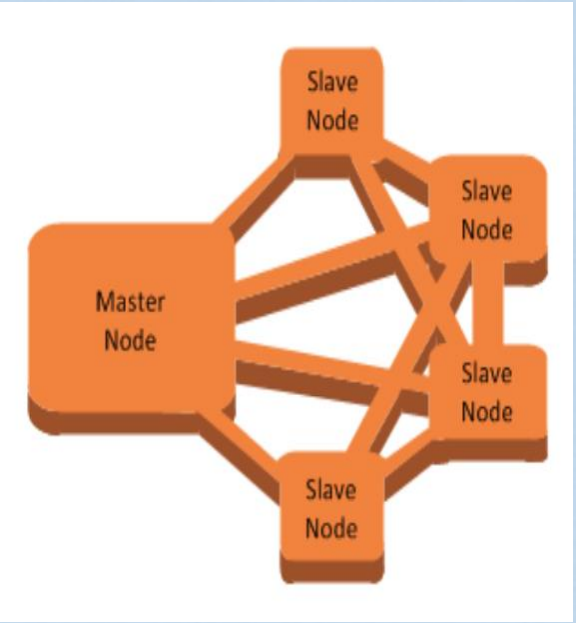
- **Apache Hadoop** is an open-source software for reliable, scalable, distributed computing.
- **Hbase** A scalable, distributed database that supports structured data storage for large tables
- **Hive** A data warehouse infrastructure that provides data summarization and ad hoc querying.
- **Hadoop Spark:** A fast and general compute engine for Hadoop data.

- **Hive** is a data warehouse infrastructure that provides data summarization and ad hoc querying
  - it is a SQL-like (uses Hive QL which is based on SQL) data warehouse infrastructure built on HDFS (Hadoop Distributed File System).
  - Hive doesn't have as robust a set of analytical functions as the MPP options.
  - It also does not fully comply with standard SQL.

- Hive + Hadoop are great for **querying unstructured data**

EMR Clusters

- **Master node**:
  - A node that manages the cluster by running software components to coordinate the distribution of data and tasks among other nodes for processing.
    - The master node tracks the status of tasks and monitors the health of the cluster. Every cluster has a master node, and it's possible to create a single-node cluster with only the master node.
- **Core node**:
  - A node with software components that run tasks and store data in the Hadoop Distributed File System (HDFS) on your cluster.
  - Multi-node clusters have at least one core node.
- **Task node**:
  - A node with software components that only runs tasks and does not store data in HDFS. Task nodes are optional.



AWS Elastic Map Reduce (EMR)

EMR Cluster Nodes

- You have root access to the EC2 instances in the EMR Cluster.
- EMR provides flexibility to scale your cluster up or down as your computing needs change.
- EMR also provides the option to run multiple instance groups so that you can:
  - Use On-Demand Instances in one group for guaranteed processing power
  - Use Spot Instances in another group to have your jobs completed faster and for lower costs.
  - You can also mix different instance types to take advantage of better pricing for one Spot Instance type over another.
- Two choices for the File system
  - **Hadoop Distributed File System (HDFS)** which runs on the master and core nodes of your cluster for processing data that you do not need to store beyond your cluster's lifecycle.
  - **EMR File System (EMRFS)** to use Amazon S3 as a data layer for applications running on your cluster so that you can separate your compute and storage and persist data outside of the lifecycle of your cluster.
- EMRFS provides the added benefit of allowing you to scale up or down for your compute and storage needs independently.
  - You can scale your compute needs by resizing your cluster and
  - You can scale your storage needs by using Amazon S3.

EMR and AWS Availability Zones

- Amazon EMR launches all nodes for a given cluster in **the same Amazon EC2 Availability Zone**.
- Running a cluster in the same zone improves performance of the jobs flows because it provides a higher data access rate.
- By default, Amazon EMR chooses the Availability Zone with the most available resources in which to run your cluster.
  - However, you can specify another Availability Zone if required.

AWS Elastic Map Reduce (EMR)

EMR Integration with other AWS Services

- Amazon EMR integrates with other AWS services to provide capabilities and functionality related to networking, storage, security, and so on, for your cluster.
- The following list provides several examples of this integration:
  - Amazon EC2
  - Amazon VPC to configure the virtual network in which you launch your instances
  - Amazon S3 to store input and output data
  - Amazon CloudWatch to monitor cluster performance and configure alarms
  - AWS Identity and Access Management (IAM) to configure permissions
  - AWS CloudTrail to audit requests made to the service
  - AWS Data Pipeline to schedule and start your clusters
  - EC2 Key pairs to allow for SSH access to EMR Cluster Nodes

Amazon EMR

Data to be processed

EMR and Data to be processed

- EMR is not about real time fast, large data ingestion
- Load data to be processed by EMR to S3
- Customers upload their input data into Amazon S3.
  - Amazon EMR then launches several Amazon EC2 instances as specified by the customer.
  - EMR pulls the data from Amazon S3 into the launched Amazon EC2 instances.
  - Once the cluster is finished, Amazon EMR transfers the output data to Amazon S3, where customers can then retrieve it or use as input in another cluster.

- The Hadoop application can also load the data from anywhere on the internet or from other AWS services.

EMR and Hive

- Hive is an open source data warehouse and analytics package that runs on top of Hadoop.
- Hive is operated by a SQL-based language called Hive QL that allows users to structure, summarize, and query data sources stored in Amazon S3.
- Hive QL supports map/reduce functions that allows processing of complex and even unstructured data sources such as text documents and log files.
- You can use Hive to process data in DynamoDB tables (Querying the data).
  - To do this, define an external Hive table based on your DynamoDB table.
  - You can then use Hive to analyze the data stored in DynamoDB (without having to transfer it to S3 first), and either load the results back into DynamoDB or archive them in Amazon S3.

EMR Kinesis Connector

- Without the Kinesis connector, for EMR to read and process data from a Kinesis stream,
  - This would require writing, deploying and maintaining independent stream processing applications.

- With the EMR Kinesis connector, you can start reading and analyzing a Kinesis stream by writing a simple **Hive or Pig script.**
  - This allows for analyzing Kinesis streams using SQL!
  - There is no need to develop or maintain a new set of stream processing applications.

- The connector enables EMR to directly read and query data from Kinesis streams.
  - This allows for performing batch processing and ad-hoc querying of Kinesis streams using existing Hadoop ecosystem tools such as Hive, Pig, MapReduce, Hadoop Streaming, and Cascading.

EMR Kinesis Connector – Use Cases

- **Streaming Log Analysis:** You can analyze streaming web logs to generate a list of top 10 error type every few minutes by region, browser, and access domains.

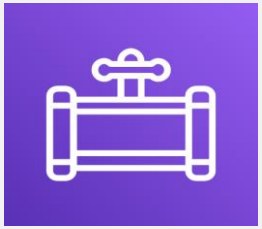
- **Complex Data Processing Workflows:**
  - You can join Kinesis stream with data stored in S3, Dynamo DB tables, and HDFS.
  - You can write queries that join clickstream data from Kinesis with advertising campaign information stored in a DynamoDB table to identify the most effective categories of ads that are displayed on particular websites.
  - Ad-hoc Queries:
    - EMR Hadoop input connector for Kinesis Does NOT support continuous stream processing

- You can NOT push data from EMR into Kinesis stream using the connector

EMR Encryption and Logs

- Amazon EMR supports optional Amazon S3 server-side and client-side encryption **with EMRFS** to help protect the data that you store in Amazon S3.
- EMR always uses HTTPS to move data between S3 and EMR cluster's EC2 instances
- Hadoop system logs as well as user logs will be placed in the Amazon S3 bucket which you specify when creating a cluster.

- Also, EMR integrates with CloudTrail and can have all API calls logged to an S3 bucket of your choice



AMAZON DATA PIPELINE

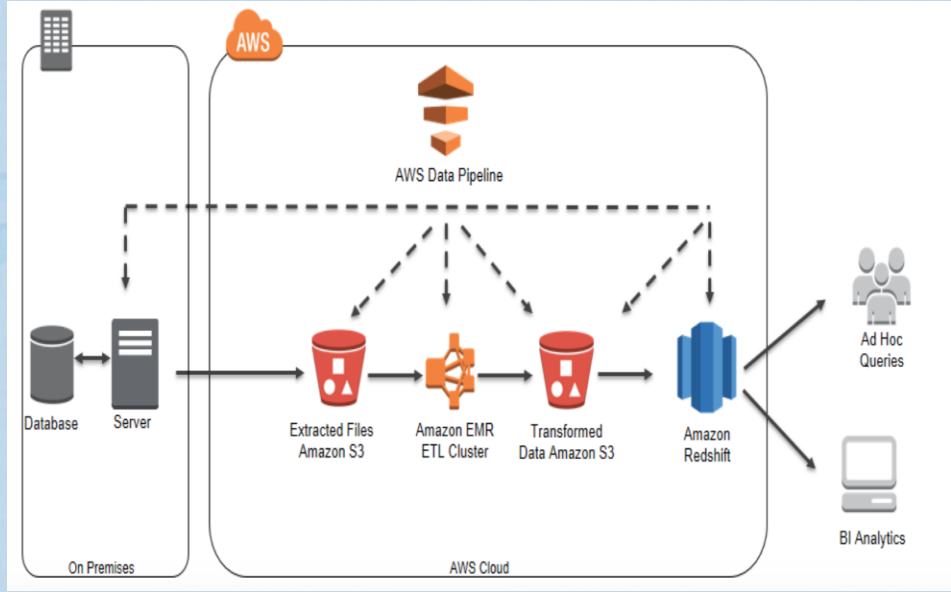
Amazon Data Pipeline

What is it?
& Components

AWS Data Pipeline

What is it?

- AWS Data Pipeline is a web service that makes it easy to schedule regular data movement and data Transformation/processing activities in the AWS cloud.
- AWS Data Pipeline integrates with **on-premise and cloud-based** storage systems to allow developers to use their data when they need it, where they want it, and in the required format.
- AWS Data Pipeline allows you to quickly define a dependent chain of data sources, destinations, and predefined or custom data processing activities called a pipeline.
- Data Pipeline provides a highly scalable and fully managed service,
  - Data Pipeline ensures that your pipelines are robust and highly available.

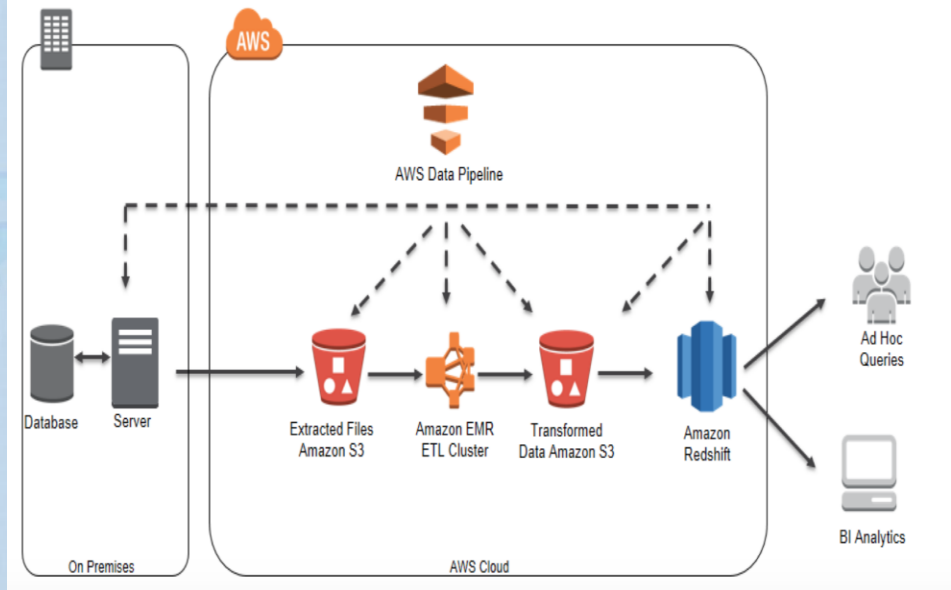


AWS Data Pipeline

What is it?

- 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 Amazon S3, Amazon RDS, or Amazon DynamoDB.

- DynamoDB to Redshift : You can load Data into Redshift from DynamoDB directly,
- RDS to Redshift : You need to copy first into S3 from RDS, then from S3 to Redshift



Data Pipeline Components

- **Task runner:**
  - Is an application that polls AWS Data Pipeline for tasks and then performs those tasks.
  - When a task is assigned to Task Runner, it performs that task and reports its status back to AWS Data Pipeline.
- **Data Nodes:**
  - A data node defines the location and type of data that a pipeline activity uses as input or output.(SQLDataNode, RedShiftDataNode, S3DataNode, DynamoDBDataNode)
- **Databases:**
  - RDS, Redshift, and JDBC are supported
- **Resources:**
  - Data Pipeline support EMR and EC2 compute Resources.
- **Activities:**
  - An activity is an action that AWS Data Pipeline initiates on your behalf as part of a pipeline.
    - EMR or Hive jobs, copies, SQL queries, or command-line scripts are example of activities.

Data Pipeline and On-Premise Resources

- You can have Task Runners on-premise as part of your Data Pipeline
- To enable running activities using on-premise resources, AWS Data Pipeline supplies a Task Runner package that can be installed on your on-premise hosts.
- This package continuously polls the AWS Data Pipeline service for work to perform.
- When it's time to run a activity on your on-premise resources, for example, executing a DB stored procedure or a database dump,
  - AWS Data Pipeline will issue the appropriate command to the Task Runner.
  - In order to ensure that your pipeline activities are highly available, you can optionally assign multiple Task Runners to poll for a given job.
    - This way, if one Task Runner becomes unavailable, the others will simply pick up its work.

Launching and Terminating Compute Resources

- AWS Data Pipeline supports Data Pipeline Managed compute resources (EMR or EC2 instances), or customer managed (on-premise or customer managed EC2 instances)

- AWS Data Pipeline will provision compute resources when the first activity for a scheduled time that uses those resources is ready to run and then,
  - Those instances will be terminated when the final activity that uses the resources has completed successfully or failed.

AWS Data Pipeline

Use Cases

Move to Cloud

- Easily copy data from your on-premises data store, like a MySQL database, and move it to an AWS data store, like S3 to make it available to a variety of AWS services such as Amazon EMR, Amazon Redshift, and Amazon RDS.

ETL Data to Amazon Redshift

- Copy RDS or DynamoDB tables to S3, transform data structure, run analytics using SQL queries and load it to Redshift.

ETL Unstructured Data

- Analyze unstructured data like clickstream logs using Hive or Pig on EMR, combine it with structured data from RDS and upload it to Redshift for easy querying.

Data Loads and Extracts

- Copy data from your RDS or Redshift table to S3 and vice-versa.

Load AWS Log Data to Amazon Redshift

- Load log files such as from the AWS billing logs, or AWS CloudTrail, Amazon CloudFront, and Amazon CloudWatch logs, from Amazon S3 to Redshift.

Amazon DynamoDB Backup and Recovery

- Periodically backup your Dynamo DB table to S3 for disaster recovery purposes.

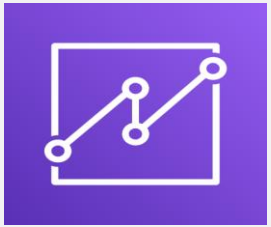
Amazon DynamoDB data copy to another Region

- You can use Data Pipeline to schedule a cross region copy of your DynamoDB table, and you can use time stamp attributes so you can only copy the data that has changed since last copy process

AWS Data Pipeline

- SNS Notification:
  - You can define Amazon SNS alarms to trigger on activity success, failure, or delay.
    - Create an alarm object and reference it in the onFail, onSuccess, or onLate slots of the activity object.

- Cost Optimization
  - Use of Spot vs On-Demand Instances
    - In case of EC2
    - In case of EMR
      - EMR Core nodes and Task nodes do different tasks
      - Use Spot instances for Task Nodes



AMAZON QUICKSIGHT

Amazon QuickSight – What is it?

- Amazon QuickSight is a business analytics service that can be used to build visualizations, perform ad hoc analysis, and get business insights from the intended data.
- It can automatically discover AWS data sources and works with available (in or outside AWS) data sources.
- Amazon QuickSight enables organizations to scale to hundreds of thousands of users and delivers responsive performance by using a robust in-memory engine (SPICE).
- Amazon QuickSight offers Standard and Enterprise editions.
- Amazon QuickSight **data sets** can be created by using your own data sources or other data sources that are shared with you.
  - Then you can use Quicksight to create **Data Analyses**, **Visuals to visualize the data**, and **share it through data dashboards.**

Data Sources and Data Preparation

- Variety of data sources can be used for Quicksight data analysis, including files, AWS services, and on-premises DBs.
- Data preparation is the process of transforming raw data for use in an analysis. This includes making changes like:
  - Filtering out data so you can focus on what's important
  - Renaming fields to make them easier to read
  - Changing data types so they are more useful
  - Adding calculated fields to enhance analysis
  - Creating SQL queries to refine data

- To get ready to **create analyses**, first Data Sets need to be created based on the data sources.
  - A **Data Set** identifies the specific fields and rows that need to be used for the analysis (data to work with).
  - In addition to raw data, a data set stores any changes made to the data, so it's ready the next time when it is required to analyze the data.

- Multiple analyses can be created using the same data set.
  - Multiple data sets can be used in a single analysis.

Working with Analyses

- Analysis is used to create and interact with visuals and stories.
- An analysis can be thought of as a container for a set of related visuals and stories,
- Multiple data sets can be used in an analysis, although any given visual can only use one of those data sets.
- After an analysis and an initial visual are created, you can expand the analysis by adding data sets and visuals, and by creating stories to add narrative to the analysis data.

Data Sources QuickSight can work with

Amazon QuickSight supports a variety of data sources that you can use to provide data for analyses.

- The following data sources are supported:
  - Relational Data Sources
    - You can use any of the following relational data stores as data sources for Amazon QuickSight:
      - Amazon Athena
      - Amazon Aurora
      - Amazon Redshift
      - Amazon Redshift Spectrum
      - Amazon S3
      - Amazon S3 Analytics
      - Apache Spark 2.0 or later
      - MariaDB 10.0 or later
      - Microsoft SQL Server 2012 or later
      - MySQL 5.1 or later
      - PostgreSQL 9.3.1 or later
      - Presto 0.167 or later
      - Snowflake
      - Teradata 14.0 or later

Visuals

- A visual, also known as data visualization, is a graphical representation of your data set.
  - You can create a wide variety of visuals (diagram, chart, graph, table…) in an analysis, using different data sets and visual types.

- All visuals begin in AutoGraph mode, which automatically selects a visualization based on the fields you select.
  - You can also take control and choose your own visuals.

- Amazon QuickSight supports a variety of visuals including combo charts, heat and tree maps, pivot tables, and more.

- Amazon QuickSight supports up to 20 data sets in a single analysis, and up to 20 visuals in a single analysis.

SPICE

- SPICE is Amazon QuickSight's Super-fast, Parallel, In-memory calculation engine.
- SPICE is engineered to rapidly perform advanced calculations and serve data.
- The storage and processing capacity available in SPICE speeds up the analytical queries that you run against your imported data.
- By using SPICE, you save time because you don't need to retrieve the data every time you change an analysis or update a visual.

AWS GLUE

AWS Glue

What is it, and How it works?

Amazon Glue – What is it?

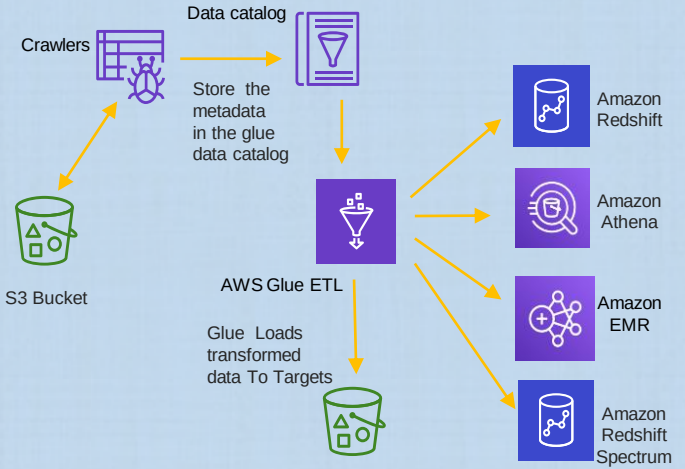
- AWS Glue is a fully-managed, **serverless, pay-as-you-go**, extract, transform, and load (ETL) service that automates the time-consuming steps of data preparation for analytics.
- It makes it simple and cost-effective to categorize the data, clean it, enrich it, and move it reliably between various data stores.
- It protects the data in transit and at rest.
- AWS Glue offers a persistent metadata store for your data in Amazon S3
- AWS Glue uses other AWS services to orchestrate your ETL (extract, transform, and load) jobs to build a data warehouse.

AWS Glue consists of:

- **Automatic crawling** to populate a central metadata repository known as the AWS Glue Data Catalog,
- **Job Authoring,** an ETL engine that automatically recommends and generates Python or Scala code, to transform the source data into target schemas.
  - It runs the ETL jobs on a fully managed, scale-out **Apache Spark** environment to load the data into its destination.
  - It also allows for the setup, orchestration, and monitoring of complex data flows.
- **Job execution,** a flexible scheduler that handles dependency resolution, job monitoring, and retries.

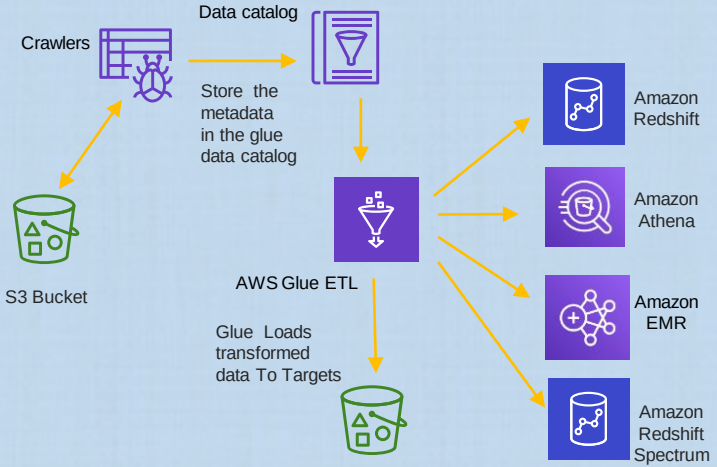
Amazon Glue – Data Catalog

- The AWS Glue Data Catalog is a central repository to store structural and operational metadata for all the relevant data assets.
  - AWS Glue automatically discovers and profiles the data via the Glue Data Catalog,
- The AWS Glue Data Catalog is Apache Hive Metastore compatible
  - It can be used as a drop-in *replacement for the Apache Hive Metastore for Big Data applications running on EMR.*
- The AWS Glue Data Catalog also provides out-of-box integration with **Amazon Athena, EMR, and Redshift Spectrum**.
  - With AWS Glue (Data Catalog), access and analysis of the data happens through one unified interface without having to load it into multiple data silos.
  - Once the table definitions (schemas) are added to the Glue Data Catalog, they are available for ETL.
    - Also the schemas will be readily available for querying in Amazon Athena, EMR, and Redshift Spectrum.
      - ➢ This is how data catalog allows for a common view of the data between these services.



AWS Glue Crawlers

- An **AWS Glue crawler** is a data crawling/cataloging program that can automatically scan your data sources, identify data formats, and derive the **schema.**
  - An AWS Glue crawler connects to a data store, progresses through a prioritized list of classifiers to extract the schema of your data and other statistics, and then populates the Glue Data Catalog with this metadata.
- Crawlers can run periodically to detect the availability of new data as well as changes to existing data, including table definition changes.
- Crawlers automatically add new tables, new partitions to existing table, and new versions of table definitions.
  - Glue crawlers can be customized to classify your own file types.
  - In addition, the crawler can detect and register partitions.



AWS Glue Crawlers (cont.)

- The AWS Glue Data Catalog provides a unified metadata repository across a variety of data sources and data formats,
  - It integrates with;
    - Amazon Athena,
    - Amazon S3,
    - Amazon RDS,
    - Amazon Redshift,
    - Amazon Redshift Spectrum,
    - Amazon EMR,
    - And any application compatible with the Apache Hive metastore.

AWS Glue

- When to use
- Patterns and Anti-Patterns

Amazon Glue – When to use it?

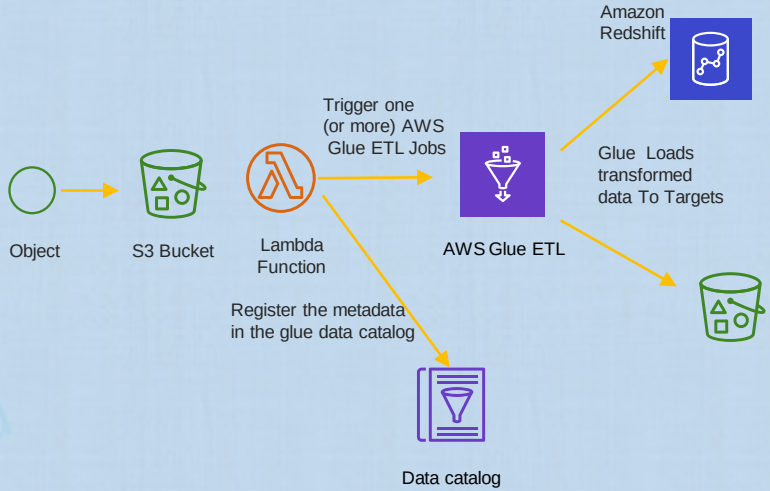
- **Use AWS Glue to run serverless queries against an Amazon S3 data lake.**
  - AWS Glue can catalog an Amazon S3 stored data, making it available for querying with **Amazon Athena** and **Amazon Redshift Spectrum.**
  - With crawlers, the metadata stays in sync with the underlying data.
    - **Athena and Redshift Spectrum** can directly query your Amazon S3 data lake using the AWS Glue Data Catalog.
  - With AWS Glue, you access and analyze data through one unified interface without loading it into multiple data silos.

- **Use AWS Glue to build a Data Warehouse to organize, cleanse, validate, and format data.**
  - You can transform and move AWS Cloud data into your data store.
  - You can also load data from disparate sources into your data warehouse for regular reporting and analysis.
    - By storing it in a data warehouse, you integrate information from different parts of your business and provide a common source of data for analytics and decision making.

Amazon Glue – When to use it?

- **Create event-driven ETL pipelines with AWS Glue.**
  - You can run your ETL jobs as soon as new data becomes available in Amazon S3 by **invoking your AWS Glue** ETL jobs from **an AWS Lambda function**.
  - You can also register this new dataset in the AWS Glue Data Catalog as part of your ETL jobs.

- **Use AWS Glue to understand your data assets.**
  - Use AWS Glue to discover properties of the data you own, transform it, and prepare it for analytics.
  - **Data Sources:** Glue can automatically discover both structured and semi-structured data stored in your data lake on Amazon S3, Data Warehouse in Amazon Redshift, and various DBs on AWS.
  - **Data Targets:** It provides a unified view of your data via the Glue Data Catalog that is available for ETL, querying and reporting using services like Amazon Athena, EMR, and Redshift Spectrum.
  - Glue automatically **generates Scala or Python code** for your ETL jobs that you can further customize using tools you are already familiar with.



Amazon Glue and building a Data Warehouse

When building a Data Warehouse, AWS Glue simplifies many tasks such as:

- It discovers and catalogs metadata about the involved data stores into a central catalog.
  - You can process semi-structured data, such as **clickstream or process logs**.
- It populates the AWS Glue Data Catalog with table definitions from **scheduled crawler** programs.
- Generates ETL scripts to transform, flatten, and enrich the data from source to target.
- Detects schema changes and adapts based on specified preferences.
- Triggers ETL jobs based on a schedule or event.
  - You can initiate jobs automatically to move data into the data warehouse.
  - Triggers can be used to create a dependency flow between jobs.
- Gathers runtime metrics to monitor the activities of the data warehouse.
- Handles errors and retries automatically.
- Scales resources, as needed, to run the scheduled jobs.

Amazon Glue & Data Pipeline

- AWS Glue provides a managed ETL service that runs on a serverless Apache Spark environment.
- It serverless hence, you do not have access to the underlying services or infrastructure used to do the Glue tasks.

- **Use Data Pipeline (with EMR) when:**
  - A managed orchestration service that gives you greater flexibility in terms of the execution environment, access and control over the compute resources that run your code, as well as the code itself that does data processing.
  - AWS Data Pipeline launches resources in your account; hence, you have low level access to EC2 instances or EMR clusters.
  - If your use case requires you to use an engine other than **Apache Spark (with Python or Scala)**
  - If you want to run a heterogeneous set of jobs that run on a variety of engines like Hive, Pig, etc., then AWS Data Pipeline would be a better choice.

Amazon Glue vs. other AWS Services

AWS Glue vs EMR

- AWS Glue works on top of the Apache Spark environment to provide a scale-out execution environment for your data transformation jobs.
- AWS Glue infers, evolves, and monitors ETL jobs to greatly simplify the process of creating and maintaining jobs.
- **Use EMR if** providing direct access to the Hadoop environment, affording a lower-level access and greater flexibility in using tools beyond Spark.

AWS Glue vs EMR

- Use DMS to move the data into AWS.

- Once your data is in AWS, you can use AWS Glue to move and transform data from your data source into another database or data warehouse.

Amazon Glue vs. other AWS Services

AWS Glue vs AWS Batch

- AWS Batch enables to easily and efficiently run any batch computing job on AWS regardless of the nature of the job.
- AWS Batch creates and manages the compute resources inside the AWS account, giving you full control and visibility into the resources being used.
- AWS Glue is a fully-managed ETL service that provides a serverless Apache Spark environment to run your ETL jobs. For your ETL use cases, we recommend you explore using AWS Glue.
- For other batch-oriented use cases, including some ETL use cases, AWS Batch might be a better fit.

AWS Glue vs Kinesis Data Analytics

- Amazon Kinesis Data Analytics, allows to run standard SQL queries on an incoming data stream.
- You can specify a destination like an S3 bucket to write the results.
- Once the data is available in the target data source, you can kick off an AWS Glue ETL job to do further transform your data and prepare it for additional analytics and reporting.