**CHAPTER 1**

**INTRODUCTION TO DATA ENGINEERING**

* 1. **Overview of AWS as a leading cloud service provider**

Amazon Web Services (AWS) is the leading cloud service provider globally, renowned for its extensive and innovative offerings that cater to businesses of all sizes. Since its launch in 2006, AWS has grown to provide over 200 services across computing, storage, databases, machine learning, and more, making it the go-to platform for enterprises seeking to leverage cloud technology.

Fig No 1.1 Features of AWS

**Compute Storage:** AWS offers scalable compute and storage solutions like Amazon EC2 and S3, enabling flexible and reliable virtual servers and data storage.

**Database Service:** Database services such as RDS and DynamoDB, supporting both SQL and NoSQL databases with high availability and scalability.

**Monitoring Services:** Tools like Amazon CloudWatch allow users to monitor and manage application performance, resource utilization, and operational health.

**Web Hosting:** AWS enables scalable and secure web hosting solutions through services like S3, EC2, and Lightsail, supporting both static and dynamic websites.

**Security and Compliance:** AWS offers robust security features, including IAM and AWS Shield, ensuring data protection and compliance with industry standards.

**Data Streaming and Analytics on AWS:** AWS facilitates real-time data processing and analytics with services like Amazon Kinesis, enabling immediate insights from streaming data.

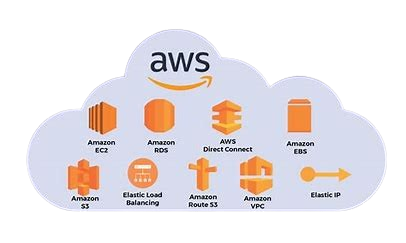
* 1. **How AWS services facilitate Data Engineering tasks**

Fig No 1.2 AWS Services for Data Engineering

**Amazon EC2:** Elastic Compute Cloud is an AWS computing web service that allows users to create and manage virtual servers, known as instances, in the cloud.

**Amazon RDS**: A managed relational database service that simplifies database setup, operation, and scaling, essential for structured data management.

**AWS Direct Connect:** It is a high-speed, low-latency connection that allows you to access public and private AWS Cloud services from your local infrastructure.

**Amazon EBS:** Elastic Block Store (EBS) provides scalable, high-performance block storage resources that can be attached to Amazon EC2 instances. It's ideal for mission-critical applications, databases, big data analytics, and more

**Amazon S3**: Provides scalable and secure object storage for large datasets, crucial for data lakes and archiving.

**Elastic Load Balancing:** It automatically distributes incoming application traffic across multiple Amazon EC2 instances, enhancing fault tolerance and ensuring seamless load balancing capacity for your applications

**AWS Route 53:** It is a scalable Domain Name System web service designed to route end-user requests to endpoints in a globally distributed and reliable manner. It offers features like domain registration, DNS routing, and health checking for applications.

**AWS VPC:** AWS Virtual Private Cloud allows you to create a private network within the AWS cloud, giving you control over your network configuration, including IP address ranges, subnets, and route tables.

**AWS Elastic IP:** AWS Elastic IP is a static, public IP address that can be associated with your AWS resources, ensuring a consistent IP address even if you need to stop and restart your instance or switch instances.

**CHAPTER 2**

**TECHNOLOGIES AND SERVICES**

### 2.1 Overview of core AWS services used in Data Engineering

Fig No 2.1 Core AWS Services used in Data Engineering

**Amazon Redshift:** Data warehousing service designed for large-scale data analysis and complex queries. It is Used for querying and analyzing large datasets. Supports fast SQL queries and integrates with various BI tools.

**AWS Glue:** Managed ETL (Extract, Transform, Load) service for data preparation and transformation. It Automates data discovery, schema inference, and data cataloging. Facilitates data cleaning, transformation, and loading into data stores.

**Amazon Athena:** Interactive query service that allows querying data directly in Amazon S3 using SQL. It is Ideal for running ad-hoc queries on data stored in S3 without the need for data movement or infrastructure management.

**Amazon Kinesis:** Real-time data streaming service. It Collects, processes, and analyzes real-time data streams. Includes Kinesis Data Streams (for streaming data), Kinesis Data Firehose (for loading data into other AWS services), and Kinesis Data Analytics (for real-time analytics).

**Amazon EMR (Elastic MapReduce):** Managed Hadoop and Spark framework for big data processing. It Processes large datasets using distributed computing. Supports frameworks like Hadoop, Spark, HBase, and Presto.

**2.2 Explanation of ETL processes using AWS Glue and Lambda**

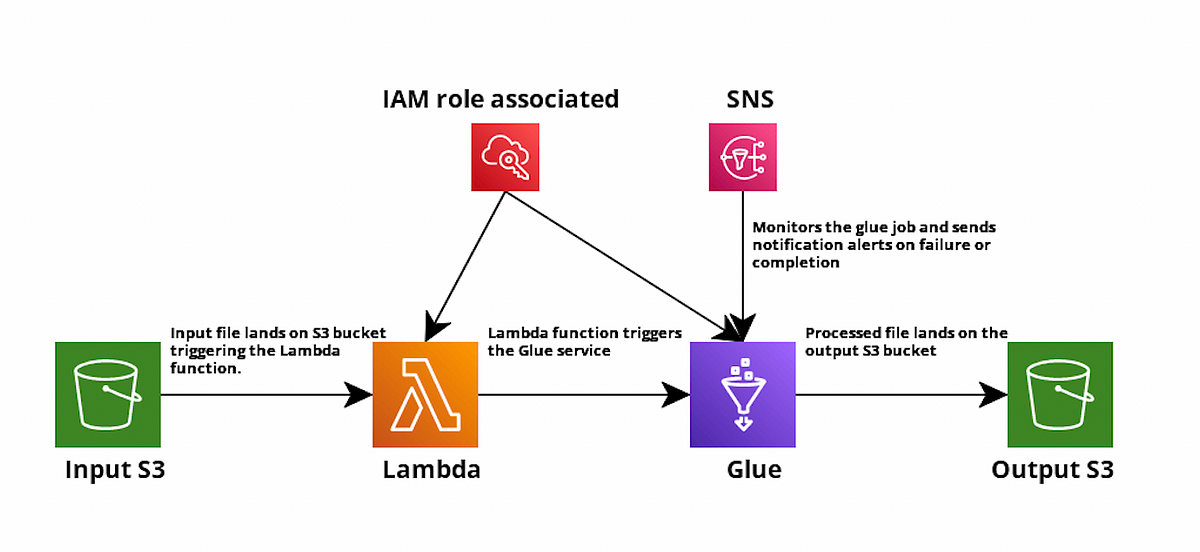
 ETL (Extract, Transform, Load) processes involve extracting data from various sources, transforming it into a usable format, and loading it into a target storage system for analysis or other purposes. Here’s how AWS Glue and AWS Lambda can be used in ETL processes:

Fig No 2.2 ETL process using AWS Glue and Lambda

**Input S3 Bucket:** The process begins when a file is uploaded to the Input S3 bucket. This bucket stores raw data that needs to be processed

**AWS Lambda:** Lambda Trigger is for uploading of a file to the Input S3 bucket triggers an AWS Lambda function. Lambda is a serverless compute service that automatically executes code in response to specific events.The Lambda function is responsible for triggering the AWS Glue job. This is done by invoking the Glue service, passing necessary parameters such as the location of the data in the Input S3 bucket.

**AWS Glue:** AWS Glue is a fully managed ETL service that handles the extraction, transformation, and loading of data. Upon being triggered by the Lambda function, Glue processes the input data, applying the necessary transformations as defined in the Glue job script.

**Output S3**: Once the Glue job completes the transformation, the processed data is written to the Output S3 bucket, which serves as the storage location for the final, transformed data.

**IAM Role Association**: An IAM (Identity and Access Management) role is associated with the Lambda function to grant it the necessary permissions to trigger Glue jobs, access S3 buckets, and interact with other AWS services.

**SNS Notifications**: SNS is integrated into the workflow to monitor the Glue job's execution. It sends notification alerts via email, SMS, or other messaging services upon the completion or failure of the Glue job, ensuring that the status of the ETL process is communicated to stakeholders or other systems in real-time.

**2.3** **Use of Amazon EMR for big data processing.**

Amazon EMR (Elastic MapReduce) is a managed big data processing service that simplifies running big data frameworks such as Hadoop, Apache Spark, HBase, Presto, and Flink on AWS. Here’s how Amazon EMR is used for big data processing:

**1. Data Processing Frameworks**

* **Apache Hadoop:** Distributed computing framework for processing large datasets across a cluster of computers. EMR allows you to run Hadoop MapReduce jobs for batch processing tasks.
* **Apache Spark:** In-memory data processing engine that supports batch and stream processing. Spark is often used for faster data processing and analytics compared to Hadoop MapReduce.

**2. Data Storage and Integration**

* **Data Sources:** EMR integrates seamlessly with various data storage solutions such as Amazon S3 (object storage for raw data), Amazon RDS (relational databases), and Amazon Redshift (data warehousing).
* **Data Ingestion:** EMR can ingest data from Amazon S3 and other AWS services, making it easy to process large volumes of data. It can also write processed data back to Amazon S3, RDS, or Redshift.

**3. Cluster Management**

* **Managed Clusters:** EMR manages the provisioning, configuration, and tuning of the cluster, which reduces the operational overhead. You can easily scale your cluster up or down based on your processing needs.
* **Automatic Scaling:** EMR supports automatic scaling, which adjusts the number of instances in the cluster based on workload requirements, optimizing cost and performance.

**4. Monitoring and Logging**

* + **Amazon CloudWatch:** Provides monitoring and alerting for cluster performance, resource utilization, and application logs.
  + **AWS CloudTrail:** Tracks API calls made to EMR, helping you audit and log activities for compliance and troubleshooting.

**5. Workflow Integration**

* + **AWS Glue:** You can use AWS Glue for ETL tasks and integrate it with EMR for more complex data processing workflows.
  + **AWS Step Functions:** Automate and orchestrate complex workflows involving EMR, Lambda, and other AWS services using Step Functions.

**CHAPTER 3**

**AWS ACADEMY CLOUD FOUNDATIONS [81188]**

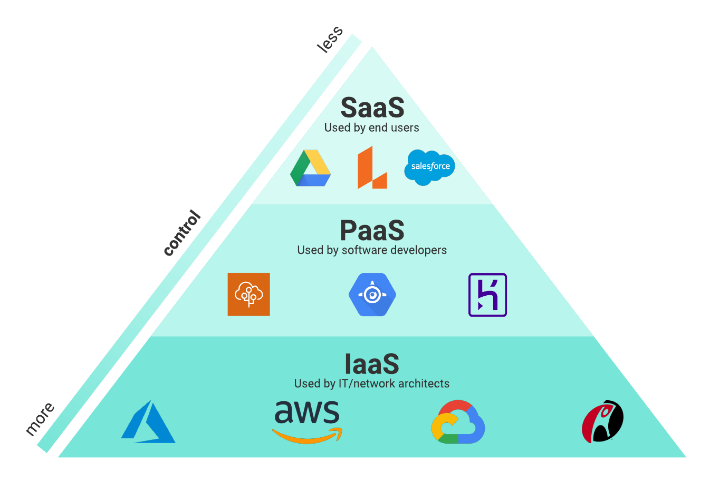
**3.1 Cloud Concepts overview:**

Fig No 3.1 Cloud Service Models

Cloud solutions come in three primary service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

* **IaaS:** Infrastructure as a service gives users access to storage, networking, servers, and other computing resources via the cloud. While the user is still responsible for managing their applications, data, middleware, etc., IaaS provides automated and scalable environments that provide a high degree of control and flexibility for the user.
* **PaaS:** Platform as a service (PaaS) is a middle ground between renting space in a datacenter (infrastructure as a service) and paying for a complete and deployed solution (software as a service). In a PaaS environment, the cloud provider maintains the physical infrastructure, physical security, and connection to the internet. They also maintain the operating systems, middleware, development tools, and business intelligence services that make up a cloud solution. In a PaaS scenario, you don't have to worry about the licensing or patching for operating systems and databases.
* **SaaS:** Software as a service (SaaS) is the most complete cloud service model from a product perspective. With SaaS, you’re essentially renting or using a fully developed application. Email, financial software, messaging applications, and connectivity software are all common examples of a SaaS implementation. While the SaaS model may be the least flexible, it’s also the easiest to get up and running. It requires the least amount of technical knowledge or expertise to fully employ.

**3.2 Cloud Economics and Billing:**

AWS costs are driven by compute, storage, and outbound data transfer. Inbound and in-region AWS transfers are usually free. Outbound data is aggregated and billed as “AWS Data Transfer Out.”

**Total Cost of Ownership (TCO):** Comparing on-premises to cloud solutions involves evaluating Total Cost of Ownership (TCO) — a financial estimate encompassing direct and indirect costs of a product or system. It considers service costs along with ownership-related expenses. TCO aids in budgeting and making informed business decisions by comparing infrastructure costs between on-premises, collocation, and cloud-based setups.

**AWS Pricing calculator:** The AWS Pricing Calculator aids in estimating monthly costs by exploring service options and creating use-case cost estimates on AWS. It facilitates informed decisions, pre-modelling solutions, and understanding price points. Users can identify cost-cutting opportunities, access instance types and contract terms, and categorize estimates into named groups for organization based on cost-centre, department, or product architecture. This tool empowers users to plan AWS costs, evaluate usage scenarios, and efficiently design new instances and services, fostering cost-effective decision-making and solution development.

**3.3 AWS Cloud Security:**

**IAM:** IAM is a feature of your AWS account offered at no additional charge. You will be charged only for use of other AWS services by your users. AWS IAM allows you to:

**Manage IAM users and their access:** You can create users in IAM, assign them individual security credentials (such as access keys, passwords, and multi-factor authentication devices), or request temporary security credentials to provide users access to AWS services and resources. You can manage permissions to control which operations a user can perform.

**Manage IAM roles and their permissions:** You can create roles in IAM and manage permissions to control which operations can be performed by the entity or AWS service that assumes the role. You can also define which entity is allowed to assume the role. In addition, you can use service-linked roles to delegate permissions to AWS services that create and manage AWS resources on your behalf.

**Manage federated users and their permissions:** You can enable identity federation to allow existing identities (users, groups, and roles) in your enterprise to access the AWS Management Console, call AWS APIs, and access resources without needing to create an IAM user for each identity. Use any identity management solution that supports SAML 2.0 or use one of our federation samples (AWS Console SSO or API federation).

**3.4 KMS:**

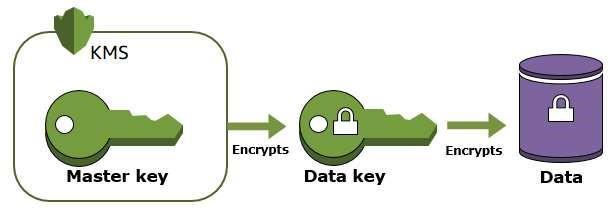
  **AWS Key Management Service (AWS KMS)** is an encryption and key management web service. Encrypts plaintext into ciphertext using a customer master key (CMK). The Encrypt operation has two primary use cases:

Fig No 3.4 KMS

* You can encrypt small amounts of arbitrary data, such as a personal identifier or database password, or other sensitive information.
* You can use the Encrypt operation to move encrypted data from one AWS Region to another.

**CHAPTER 4**

# AWS ACADEMY DATA ENGINEERING [70557]

**4.1 Elements of Data**

Fig No 4.1 Elements of Data

The five Vs are volume, velocity, variety, veracity and value.

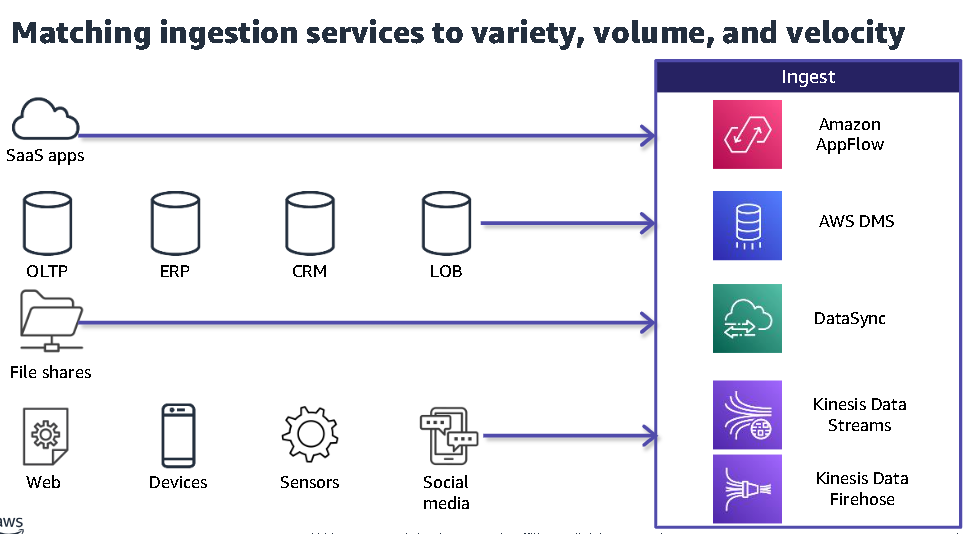
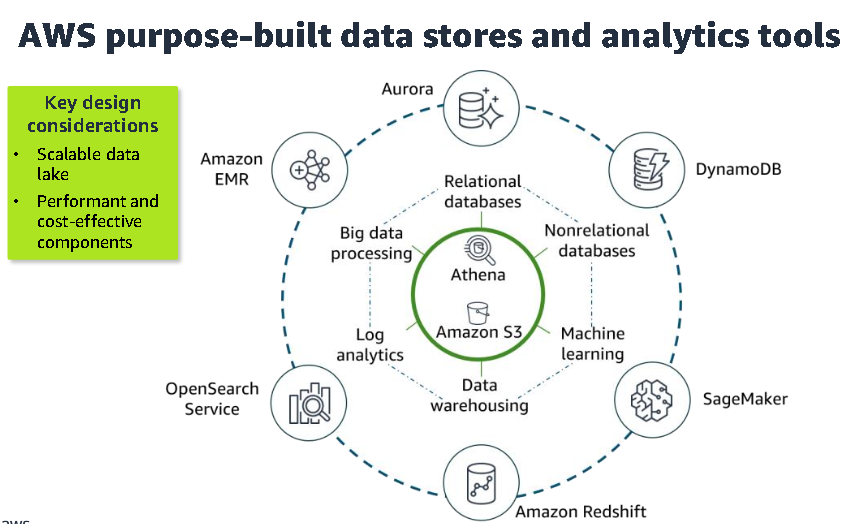
* Consider volume and velocity together because you will make infrastructure decisions about how to collect, store, and process data based on the combination of how much data you need to ingest and how quickly you will ingest it.
* Variety and veracity both relate to the data itself. What type of data is it and what's the quality of it. Data engineers and data scientists will transform and organize the data based on its variety and veracity to make it useful for analysis.
* Value is about ensuring that you are getting the most out of the data that you have collected. Value is also about ensuring that there is business value in the outputs from all that collecting, storing, and processing.
  1.  **Matching ingestion services to variety, volume, and velocity:**

Fig No 4.2 Ingestion services to variety-volume-velocity

The ingestion layer uses individual purpose-built AWS services to match the unique connectivity, data format, data structure, and data velocity requirements of source types and to deliver them to the storage layer components. The services include the following:

* Amazon AppFlow can ingest from Software as a service (SaaS) application, such as Salesforce or Zendesk
* AWS Database Migration Service (AWS DMS) can ingest from operational databases like online transaction processing (OLTP), enterprise resource planning (ERP), customer relationship management (CRM) and line of business (LOB) databases.
* AWS DataSync can ingest from file shares.
* Amazon Kinesis Data Streams and Amazon Kinesis Data Firehose can ingest from streaming data sources.
  1.  **Built-in Data stores and analytics tools**

The architecture illustrates the following other AWS purpose-built services that integrate with Amazon S3

Fig No 4.3 Built-in Data stores and analytics tools

* Amazon Redshift is a fully managed data warehouse service.
* Amazon OpenSearch Service is a purpose-built data store and search engine that is optimized for real-time analytics, including log analytics.
* Amazon EMR provides big data processing and simplifies some of the most complex elements of setting up big data processing.
* Amazon Aurora provides a relational database engine that was built for the cloud.
* Amazon DynamoDB is a fully managed nonrelational database that is designed to run high-performance applications.
* Amazon SageMaker is an AI/ML service that democratizes access to ML processing.

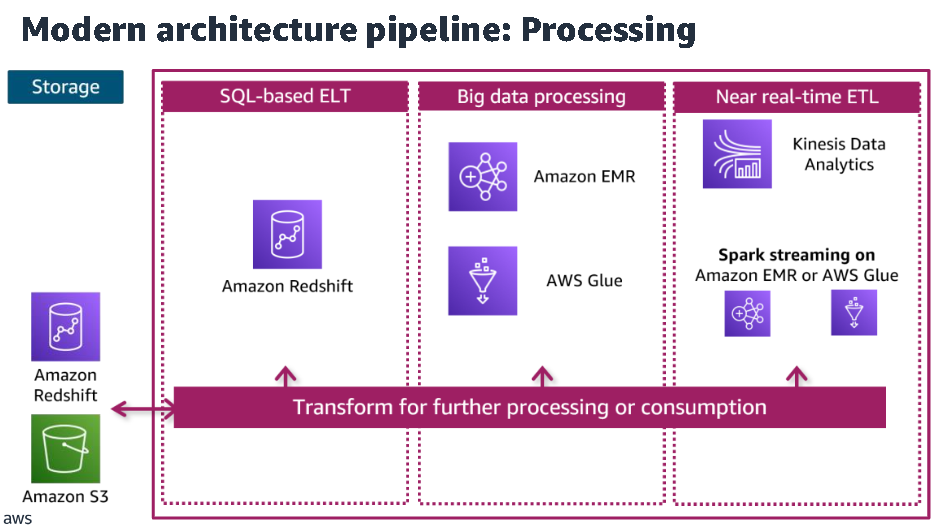
**4.4 Pipeline Architecture**

Fig No 4.4 Pipeline Architecture

Each pipeline reads data from the storage layer, processes it using temporary storage as needed, and then writes it to the appropriate location in the storage layer. The transformations are grouped into three main types, which are aligned to the use case:

* SQL-based processing using a data warehouse (in this case, Amazon RedShift)
* Big data processing using big data tools (in this case, Amazon EMR and AWS Glue)
* Near real-time processing using streaming (in this case, Amazon Kinesis Data Analytics or Spark streaming on Amazon EMR or AWS Glue)

**4.5 Stream processing pipeline**

Streaming pipelines follow the same general layers as other pipelines, but there are some unique considerations. Data sources include clickstream logs, mobile apps, existing application databases, or Internet of Things (IoT) sensors. You might want to respond to this data in real time or use it for analysis later Producers ingest records onto the stream. Producers are integrations that collect data from a source and load it onto the stream. Consumers process records. Consumers read data from the stream and perform their own processing on it. The stream itself provides a temporary but durable storage layer for the streaming solution.

In the pipeline that is depicted in this slide, Amazon CloudWatch Events is the producer that puts CloudWatch Events event data onto the stream. Kinesis Data Streams provides the

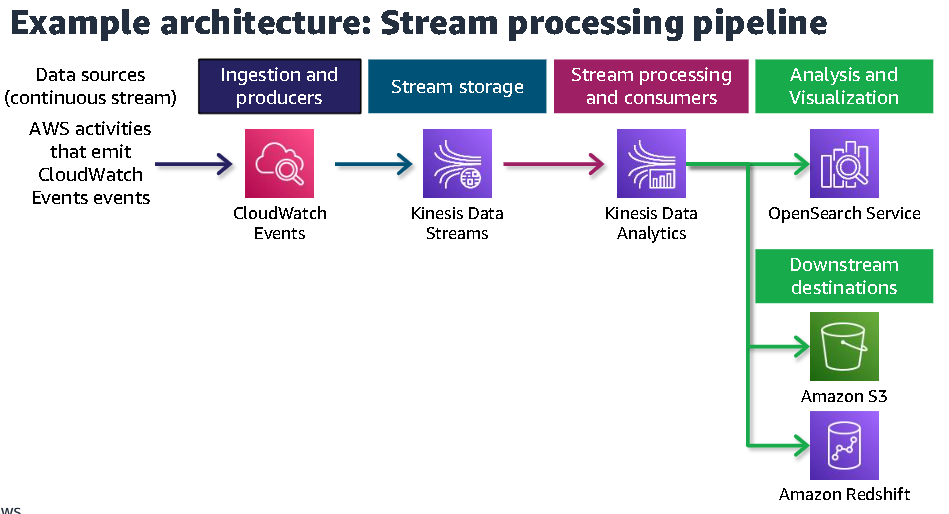
storage. The data is then available to multiple consumers.

Fig No 4.5 Stream processing pipeline

With real-time streaming analytics, records on the stream are typically processed sequentially and incrementally by recording over sliding time windows. In the pipeline that is depicted on the slide, Kinesis Data Analytics is a consumer of the stream and processes streaming data by using custom applications or standard SQL. In this example, results are sent to OpenSearch Service, where they can be used to visualize real-time insights with OpenSearch Dashboards immediately.

In this scenario, Amazon S3 and Amazon Redshift also consume the data that Kinesis Data Analytics processes. These downstream destinations aren't being used for real-time analytics but could be used for serving applications such as one-time analytics and ML. This is an example of how the modern data architecture supports the goal of making ingested data available to let different consumers perform different types of analytics and run AI/ML applications.

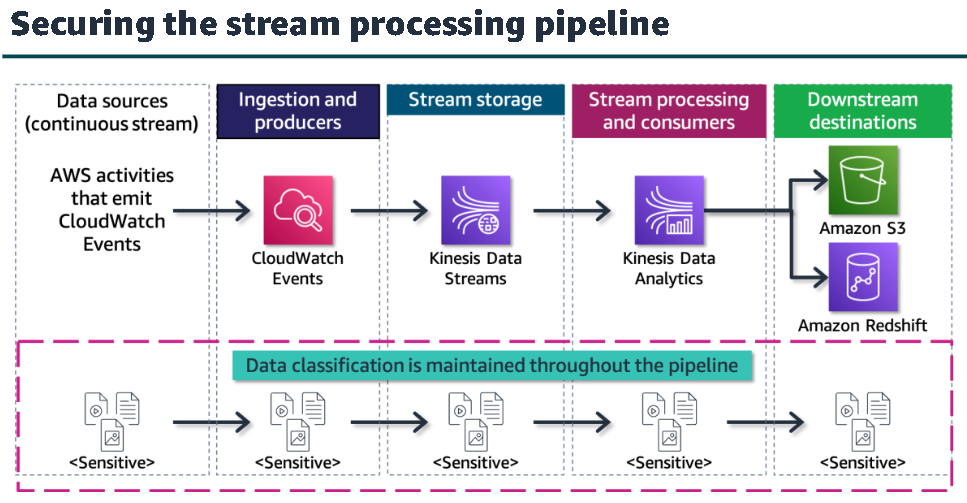
**4.6 Securing the stream processing pipeline**

Fig No 4.6 Securing the stream processing pipeline

How to apply some of the security best practices from the AWS Well-Architected Data Analytics Lens to a typical stream processing pipeline.

* **Understand data classifications and their protection policies:** This best practice will govern how you handle all the data that flows through your data pipeline. Reviewing organization's policies for classifying sensitive data and knowing what steps you will need to take to ensure adherence to those policies.
* Identify the source data owners and have them set the data classifications: The source data types that are listed in the operational data section of your pipeline will likely have various teams or individuals as owners. Identify who the dataset owners are, and request that they properly classify their datasets if they haven't already done so.
* Record data classifications into the Data Catalog so that analytics workloads can understand: Keep your Data Catalog up to date so that you have accurate and reliable records of data locations and classifications.
* Implement encryption policies for each class of data in the analytics workload: After you identify the source data that you will work with and have established the classification level of each dataset, you will need to implement the applicable encryption policies. For data at rest, use one of the multiple encryption options available in Amazon S3. Secure your Amazon Kinesis data streams with server-side encryption by using AWS KMS.
* Implement data retention policies for each class of data in the analytics workload: Use classification-based retention policies for your datasets. Back up and retain analytics datasets based on your organization's policies for classified data.
* Require downstream systems to honor the classifications: Ensure that downstream services (such as Amazon Redshift), storage services, and downstream workloads honor your data classifications. If confidential source data enters your pipeline, handle the output of your pipeline as confidential data.
* Allow data owners to determine which people or systems can access data in analytics and downstream workloads: This applies to the owners of the operational data shown in the diagram as well as any downstream workloads that could be using the data that your workload produces.
* Build user identity solutions that uniquely identify people and systems: Implement IAM or another centralized identity management solution to control which users, roles, or services can access your resources. Every service shown in the diagram integrates with IAM and should be correctly configured to do so.
* Implement the required data access authorization models: The operational data in this diagram is classified as sensitive, which means that appropriate control measures must be put in place to adequately secure the data.
* Establish an emergency access process to the source, analytics, and downstream systems: Each service and resource in the pipeline should have access controls implemented. Ensure that you implement emergency access capabilities that would enable expedited access to your workload in the event of an issue with your pipeline.
* Prevent unintended access to the infrastructure: Implement IAM to control user, role, and system access to the data and services within your workload. Isolate your network as much as possible while maintaining only the network connections that are necessary to support your analytics workload.
* Implement least privilege policies for source and downstream systems: After you have identified your source and downstream users and systems, identify the minimum privileges that each would require and implement only the necessary permissions. For example, if the downstream Amazon RedShift service needs access to your Amazon Kinesis Data Analytics service data, ensure that the Amazon Redshift users have the minimal amount of access that is required.
* Monitor the infrastructure changes and the user activities against the infrastructure: After you secure access to your operational analytics pipeline, it's vital that you monitor it for infrastructure changes and malicious activity. Use additional AWS services, such as AWS Config and GuardDuty, to monitor your infrastructure. Use CloudTrail in conjunction with IAM to log user activities against your resources.
* Secure the audit logs that record every data or resource access in analytics infrastructure: Finally, secure the audit logs for all the services in your operational analytics pipeline and the services that support it. Ensure that you maintain adequate log security over a fault-tolerant storage solution, such as Amazon S3.

**4.7 Processing of Big Data**

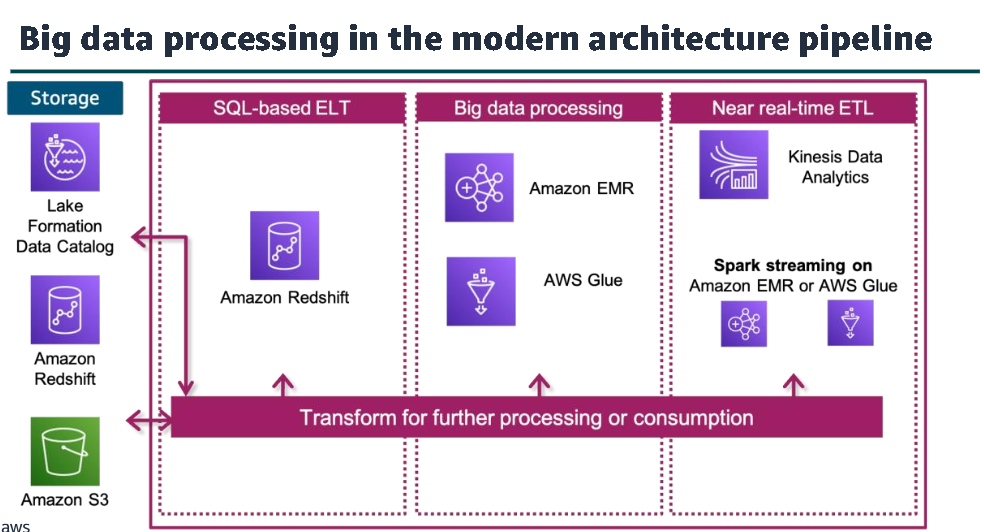
In the Design Principles and Patterns for Data Pipelines module, you learned that components in the data processing layer are responsible for transforming data into a consumable state. The processing layer provides purpose - built components that enable a variety of data types, velocities, and transformations. Each component can read and write data to both Amazon S3 and Amazon Redshift in the storage layer, and all can scale to high data volumes.

Fig No 4.7 Processing of Big Data

This architecture supports multiple ETL or ELT pipelines that perform iterative processing of data for different types of preparation and consumption. Each pipeline reads data from the

storage layer, processes it using temporary storage as needed, and then writes it to the appropriate location within the storage layer.

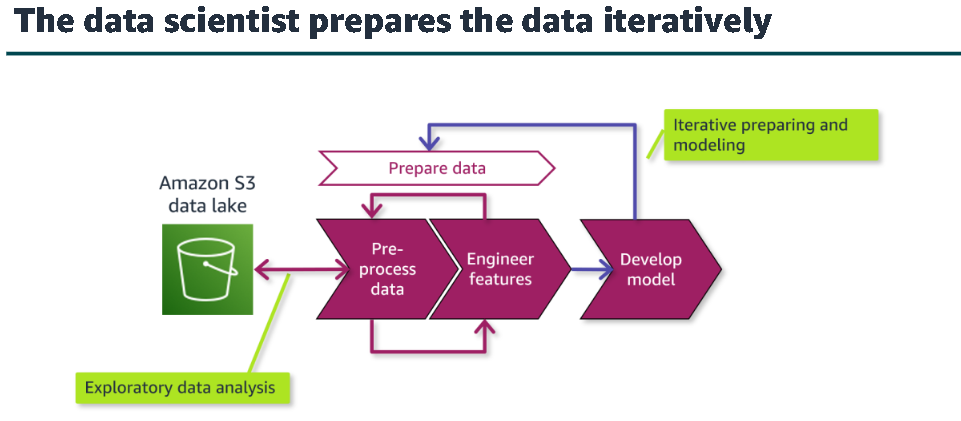
**4.8 Processing data for Machine Learning**

Fig No 4.8 Processing data for Machine Learning

Data preprocessing puts data into the correct shape and quality for training. With ML data processing, it’s mostly the data scientist's role to prepare and wrangle the data. Their activities are similar to the data wrangling steps that were presented in the Ingesting and Preparing Data module, but ML preprocessing includes activities that are specific to preparing data for ML model.

An important component of the data preparation step is the analysis and visualization of the data as it’s being explored and processed. Exploratory data analysis (EDA) with visualization tools can help a data scientist to quickly gain a deeper understanding of data.

This helps to identify patterns that are not evident by looking at data in tables. Data scientists might employ a variety of tools to serve this purpose, some of which let them combine data exploration, wrangling, and interactive analysis and visualization in an integrated way.

Many data preprocessing strategies exist, including those listed here. Data scientists would apply the appropriate strategies based on the dataset and the intended use case. This might include a variety of statistical modeling techniques or steps, such as creating rotated or resized versions of an image to augment an existing set of images. Many of the individual strategies that are listed contribute to the important strategy of balancing — or unbiasing — data. To produce accurate predictions, it’s important that the data scientist identify and mitigate bias in the data. For example, if the goal is to predict outcomes for a global customer base, but the dataset includes a large percentage of samples from one country, the results would be biased.

Data wrangling tools can simplify the application of these strategies. Preprocessing is closely tied to feature engineering, and you might find overlap in how different ML tools or references categorize preprocessing activities compared to feature engineering activities. The goal of preparing data is to provide a clean, balanced dataset that the desired algorithm can use and will produce accurate results. Preprocessing focuses on getting the dataset to where the data scientist can perform the more advanced feature engineering tasks.

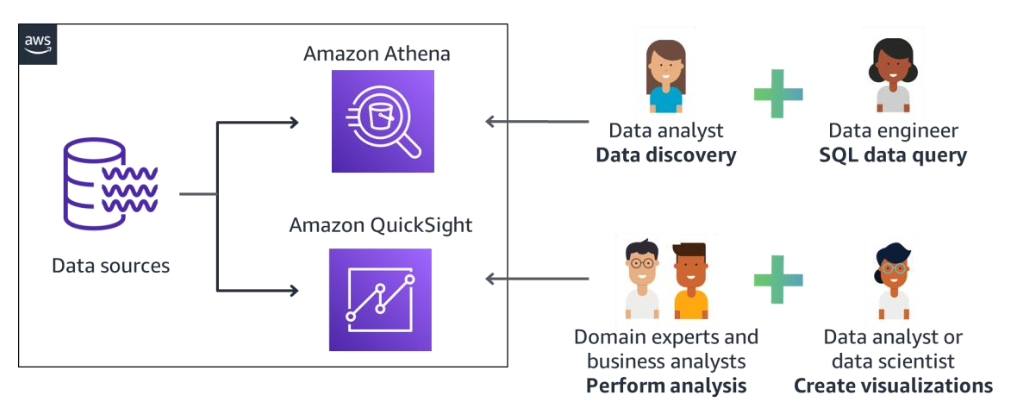
**4.9 Analyzing and Visualizing Data**

Fig No 4.9 Analyzing and Visualizing Data

**Amazon Athena:** Amazon Athena is an interactive query service that provides the ability to use SQL to analyze data in Amazon S3. Athena includes the following features:

* It is serverless
* Provides the ability to combine data from multiple data sources
* Can be used for one - time queries
* Can be used from your favorite business intelligence (BI) tools (such as QuickSight).
* Can update data stored in Amazon S3 with Apache Iceberg integration Amazon Athena is an interactive query service in which you can use standard SQL to query and analyze data.

**Amazon QuickSight:**

* It Is a cloud - scale BI service to deliver easy - to - understand insights
* Connects to data in the cloud and combines data from many different sources
* Gives decision - makers the opportunity to explore and interpret information in an interactive visual environment
* Provides forecasting visualization capabilities
* Provides ability to ask questions using natural language with QuickSight Q feature

QuickSight helps decision - makers interact with data in a visual environment. The service is designed to deliver easy - to - understand insights quickly.

**CHAPTER 5**

**LEARNING OUTCOMES**

* **Understanding AWS Services**: Gained comprehensive knowledge of various AWS services including Amazon EC2, S3, RDS, and DynamoDB, and their role in data engineering tasks such as compute, storage, database management, and real-time data analytics.
* **Data Engineering Concepts**: Developed a solid understanding of core data engineering concepts like ETL processes, big data processing, and real-time data streaming, using AWS services such as AWS Glue, Amazon Kinesis, and Amazon EMR.
* **Hands-on Experience with AWS Tools**: Acquired practical experience in deploying and managing cloud-based data pipelines, from data ingestion to transformation, using services like Amazon Redshift, Amazon Athena, and AWS Lambda for automation.
* **Cloud Architecture and Design**: Learned how to design scalable, secure, and cost-effective cloud architectures for handling large-scale data processing by leveraging AWS’s managed services.
* **Security and Compliance**: Gained awareness of AWS security best practices, including Identity and Access Management (IAM), encryption with AWS KMS, and implementing secure data retention and access policies.
* **Big Data and Machine Learning Integration**: Understood how AWS integrates with machine learning tools, facilitating data preparation, transformation, and training using Amazon SageMaker, and applied these concepts to real-world datasets.
* **Monitoring and Optimization**: Developed skills in monitoring cloud infrastructure and optimizing data workflows for cost-efficiency and performance using AWS CloudWatch, AWS Step Functions, and other monitoring tools.

**CHAPTER 6**

**CONCLUSION**

* The AWS Data Engineering internship provided an invaluable opportunity to explore and apply cutting-edge cloud technologies in real-world data engineering scenarios. Through hands-on experience with AWS services like EC2, S3, Redshift, Glue, and Kinesis, I gained a deep understanding of how cloud platforms support scalable, secure, and efficient data processing.
* The internship allowed me to build, manage, and optimize data pipelines that handle both batch and real-time data, while adhering to best practices in security and compliance. The exposure to tools such as Amazon Athena and QuickSight also helped me understand the importance of data visualization and business intelligence in driving data-driven decision-making.
* Additionally, the internship strengthened my knowledge of cloud architecture, enabling me to design robust, cost-effective solutions for processing large datasets. I gained proficiency in leveraging AWS tools for machine learning workflows, using SageMaker to integrate data science with cloud services. This experience not only enhanced my technical skills but also highlighted the importance of collaboration, problem-solving, and efficient communication in professional settings.
* In conclusion, this internship was instrumental in deepening my understanding of cloud data engineering and its applications. It equipped me with the skills and confidence to contribute meaningfully to future data-driven projects and provided a strong foundation for continued learning and growth in the field of cloud computing and data engineering.

**INTERNSHIP CERTIFACTE**

**REFERENCES**

* <https://awsacademy.instructure.com/courses/81188>
* <https://awsacademy.instructure.com/courses/70557>
* <https://github.com/pablosalme/aws-data-engineering>