An Internship Report

on

AWS Data Engineering Virtual Internship

Submitted in partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science and Engineering (Data Science)

by

BHAVYA D **214G1A3208**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE)

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)

(Affiliated to JNTUA, accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE))
Rotarypuram village, B K Samudram Mandal, Ananthapuramu-515701.

2024 - 2025

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

(Affiliated to JNTUA, accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE))
Rotarypuram village, B K Samudram Mandal, Ananthapuramu-515701.

Department of Computer Science & Engineering (Data Science)

Certificate

This is to certify that the internship report entitled AWS Data Engineering Virtual Internship is the bonafide work carried out by **BHAVYA D** bearing Roll Number 214G1A3208 in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering (Data Science)** for ten weeks from April 2024 to June 2024.

Internship Coordinator

Dr. G. Hemanth Kumar Yadav, M. Tech., Ph.D.,
Associate Professor

Head of the Department

Dr. P. Chitralingappa, M.Tech., Ph.D.,
Associate Professor, HOD
CSE(AI & ML, Data Science)

Date:

Place: Ananthapuramu

EXTERNAL EXAMINER

PREFACE

All India Council for Technical Education (AICTE) has initiated various activities for promoting industrial internship at the graduate level in technical institutes and Eduskills is a Non-profit organization which enables Industry 4.0 ready digital workforce in India. The vision of the organization is to fill the gap between Academic and Industry by ensuring world class curriculum access to the faculties and students. Formation of the All-India Council for Technical Education (AICTE) in 1945 by the Government of India.

Purpose:

With a vision to create an industry-ready workforce who will eventually become leaders in emerging technologies, EduSkills & AICTE launches 'Virtual Internship' program on AWS cloud. This field is one of the most in-demand, and this internship will serve as a primer.

Company's Mission Statement:

The main mission of these initiatives is enhancement of the employability skills of the students passing out from Technical Institutions Business Activities.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that I would like to express my indebted gratitude to my internship coordinator **Dr. G. Hemanth Kumar Yadav, Associate Professor** who has supported me a lot and encouraged me in every step of the internship work. I thank him for the stimulating support, constant encouragement and constructive criticism which have made possible to bring out this internship work.

I am very much thankful to **Dr. P. Chitralingappa, Associate Professor & HOD, Computer Science and Engineering (AI & ML, Data Science),** for his kind support and for providing necessary facilities to carry out the work.

I wish to convey my special thanks to **Dr. G. Balakrishna, Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing my internship. Not to forget, I thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported me in completing my internship in time.

I also express our sincere thanks to the Management for providing excellent facilities and support**.**

Finally, I wish to convey my gratitude to my family who fostered all the requirements and facilities that I need.

BHAVYA D

(214G1A3208)

INDEX

LIST OF FIGURES

LIST OF ABBREVATIONS

AWS	Amazon Web Services
CDN	Content Delivery Network
IAM	Identity and Access Management
EC2	Elastic Cloud Compute
VPC	Virtual Private Cloud
KMS	Key Management System
KPI	Key Performance Indicators
DNS	Domain Name System
ETL	Extract. Transform, Load
JSON	JavaScript Object Notation
CRM	Customer Relationship Management
S3	Simple Storage Service
ELB	Elastic Load Balancer
RDS	Relational Database Service

CHAPTER -1

INTRODUCTION

Introduction to AWS:

Amazon Web Services (AWS) was officially launched in 2006, marking the beginning of a significant shift in how computing resources are delivered and consumed. The initial offerings, Amazon S3 (Simple Storage Service) and Amazon EC2 (Elastic Compute Cloud), were designed to address the need for scalable and cost-effective infrastructure solutions. Amazon S3 provided a robust and scalable storage solution, allowing users to store and retrieve any amount of data from anywhere on the web. Meanwhile, Amazon EC2 offered resizable compute capacity in the cloud, enabling usersto run virtual servers with flexibility and scalability.

What is AWS ?

Amazon Web Services (AWS) is a comprehensive and widely adopted cloud computing platform offered by Amazon, providing a vast array of cloud services that include computing power, storage, and networking. Launched in 2006, AWS has revolutionized the way businesses and individuals deploy and manage IT infrastructure. It offers scalable and flexible resources that enable users to run applications, store data, and process information without the need for physical hardware. AWS's suite of services encompasses various domains, including artificial intelligence, machine learning, analytics, databases, and IoT, among others. It operates on a pay-as-you-go pricing model, which allows users to pay only for the resources they consume, thereby optimizing costs and eliminating the need for significant upfront investments. AWS's global infrastructure, with multiple geographic regions and Availability Zones, ensures high availability, reliability, and low latency for applications worldwide. The platform's robust security features, compliance certifications, and extensive range of tools and services make it a preferred choice for businesses of all sizes seeking to innovate, scale, and optimize.

Fig 1.1 AWS Cloud

1.2 Introduction to Data Engineering:

Data Engineering involves designing, constructing, and maintaining data architectures and systems. It focuses on transforming raw data into valuable information through processes such as ETL (Extract, Transform, Load). Data engineers use various tools and technologies to build data pipelines and ensure data quality. This field is essential for enabling data-driven decision-making within organizations.

What is Data ?

Data is a collection of facts, measurements, or observations that can be used as a basis for reasoning, discussion, or calculation. Data is the raw material for generating meaningful information.

Types of Data

1. Structured Data Formats

Structured data refers to data that is highly organized and formatted in a way that makes it easy to enter, store, query, and analyze. Common formats include:

- **CSV (Comma-Separated Values):** A simple text format where data is stored in rows and columns, with values separated by commas. It's easy to read and write but can be inefficient for large datasets.

- **JSON (JavaScript Object Notation):** A lightweight format used to store and exchange data between systems, often in web applications. JSON is human-readable and can handle nested structures, making it ideal for semi-structured data.
- **XML (eXtensible Markup Language):** Similar to JSON but more verbose. It is often used in web services and for transferring data between different systems.
- **AVRO:** A row-based data serialization system that is compact and fast. Avro is commonly used in Hadoop and is language-agnostic, supporting schema evolution.

2. Semi-Structured Data Formats

Semi-structured data doesn't follow a strict tabular format but contains tags or markers to separate elements within the data. Common formats include:

- **Parquet:** A columnar storage format, highly efficient for querying and analytic workloads. Parquet is optimized for read-heavy tasks and supports complexnested data structures, making it suitable for big data processing platforms like Apache Spark and Hadoop.
- **ORC (Optimized Row Columnar):** Another columnar storage format optimized for big data frameworks like Hive and Hadoop. ORC reduces the size of the data significantly and improves read performance.
- **Thrift:** Originally developed by Facebook, Thrift is a serialization framework that is used to define and process complex data structures. It supports different languages, making it highly flexible.

3. Unstructured Data Formats

Unstructured data refers to data that lacks a predefined structure or format. This type of data is typically stored in its raw format, and common examples include:

- **Text Files (TXT):** Simple text documents with no predefined structure. Though inefficient for processing large datasets, text files are common for storing logs or streaming data.

- **Images, Videos, and Audio Files:** Formats such as JPEG, PNG, MP4, and MP3 are examples of unstructured data commonly handled in big data systems. These formats require specialized processing tools like deep learning frameworks
- **Binary Large Objects (BLOBs):** BLOBs can store a variety of unstructured data, including images, audio, and documents, and are often used in databases for handling large binary files.

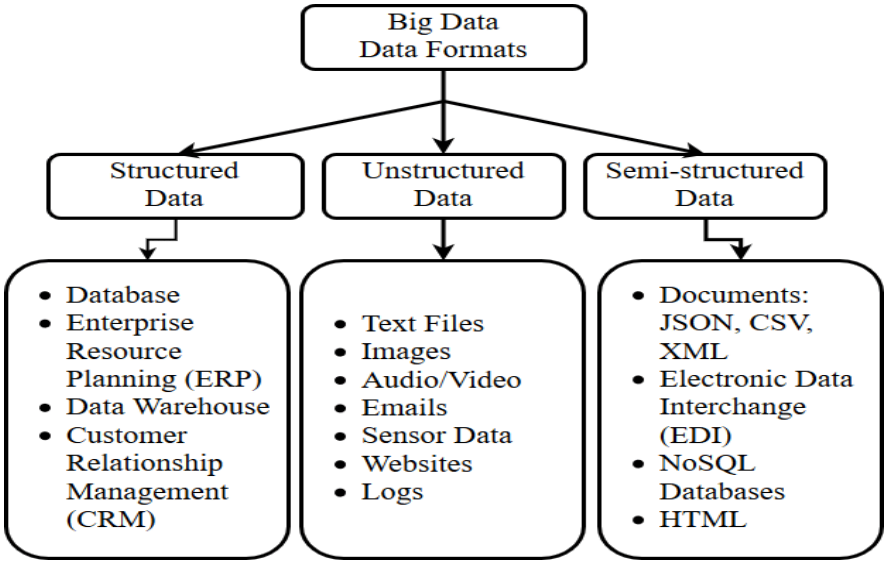


Fig 1.2 Types Of Data

Key Responsibilities of Data Engineering:

- **Designing Data Pipelines:** Creating systems that collect, process, and transport data from various sources to a centralized location.
- **Data Integration:** Combining data from multiple sources to provide a unified view.
- **Data Storage:** Setting up and managing databases, data warehouses, and data lakes.
- **ETL Processes:** Implementing Extract, Transform, Load workflows to clean and prepare data for analysis.
- **Performance Optimization:** Ensuring that data systems perform efficiently and can handle large volumes of data.

CHAPTER – 2

AWS CLOUD FOUNDATIONS

Amazon Web Services (AWS) provides a comprehensive suite of cloud computing services that enable businesses to build and deploy applications with flexibility, scalability, and cost efficiency. The AWS Cloud Foundations encompass a broad range of fundamental services and concepts essential for leveraging the full potential of the cloud.

2.1 AWS Global Infrastructure overview:

The AWS Global Infrastructure is designed to provide a robust, scalable, and reliable foundation for delivering services to customers across the globe. This infrastructure is the backbone of AWS's ability to deliver a wide array of cloud services, enabling customers to deploy applications closer to their users, meet regulatory requirements, and achieve high availability and reliability**.**

Fig 2.1 AWS Cloud Infrastructure

Regions: AWS Regions are separate geographic areas where AWS data centers are clustered. Each region is isolated from the others, providing a level of fault tolerance and stability.

Availability Zones: Each region consists of multiple, physically separated Availability Zones. AZs are designed to be isolated from failures in other AZs, offering customers the ability to run apps with high availability and fault tolerance.

Edge Locations: These are locations worldwide where AWS has deployed resources to cache copies of your data closer to end users. AWS uses edge locations for services like Amazon CloudFront and Route 53, reducing latency by serving content faster to users around the globe.

AWS Regional Edge Caches: These are a feature of Amazon CloudFront, a content delivery network (CDN) offered by AWS. They are used to cache content at edge locations, which are strategically situated around the world. This allows for faster delivery of content to users and reduces latency.

Wavelength Zones: There are specific edge locations that are designed to provide ultra-low latency and high-bandwidth access to AWS services and applications. They are typically deployed in conjunction with telecommunications providers and are connected to the AWS global network.

2.2 AWS Core Services Overview

Fig 2.2 AWS Core Services

2.2.1 Compute Services:

i) EC2 (Elastic Compute Cloud): EC2 provides virtual machines, known as instances, with varying CPU, memory, and storage capacities. You can choose from a range of instance types, including general-purpose, compute-optimized, memory-optimized, and more. EC2 supports various operating systems, such as Windows, Linux, and AWS Linux. EC2 instances run in a Virtual Private Cloud (VPC), providing isolation and security.

ii) **Lambda:** Lambda allows you to run small code snippets, called functions, written in supported languages like Node.js, Python, and Java. You can trigger these functions through API calls, data changes, schedules, Alexa skills, and more. Lambda provides a managed runtime environment for executing functions and automatically scales to handle demand. You can use Lambda for real-time data processing, API backends, IoT data processing, and serverless architectures.

iii) **Elastic Beanstalk:** Elastic Beanstalk is a managed platform for deploying web apps and services written in supported languages like Node.js, Python, and Ruby. You can upload your code and configure settings like environment variables and instance types. Elastic Beanstalk then deploys your application to AWS resources like EC2,RDS.

2.2.2 Storage Services:

i) **S3 (Simple Storage Service):** S3 is an object storage service that allows you to store and retrieve large amounts of data. You can store files, images, videos, and other types of data in S3. S3 provides durable, secure, and highly scalable storage that can be accessed from anywhere on the internet.

ii) **BS (Elastic Block Store):** EBS is a block-level storage service that provides persistent storage for EC2 instances. You can create EBS volumes and attach them to EC2 instances. EBS provides high-performance storage that can be used for databases, file systems, and other applications.

iii) **EFS (Elastic File System):** EFS is a file-level storage service that provides a shared file system for EC2 instances. You can create EFS file systems and mount them to EC2 instances. EFS provides a highly available and durable file system that can be used for big data analytics, machine learning, and other applications.

2.2.3 Data Base Services:

RDS (Relational Database Service): RDS is a managed relational database service that supports various database engines, including MySQL, PostgreSQL, Oracle, and SQL Server. You can create and manage databases, perform backups and restores, and monitor performance. RDS provides high availability, durability, and security for your relational databases.

DynamoDB: DynamoDB is a fast, fully managed NoSQL database service that provide single-digit millisecond performance. You can store and retrieve large amounts of data and use secondary indexes to improve query performance. DynamoDB provides high availability, durability, and security for your NoSQL databases.

DocumentDB: DocumentDB is a document-oriented database service that supports MongoDB workloads. You can store and retrieve JSON documents, and use secondary indexes to improve query performance. DocumentDB provides high availability, durability, and security for your document-oriented databases.

2.3 Networking and content delivery:

VPC (Virtual Private Cloud): VPC is a networking service that allows you to create a virtual network in the cloud. You can create subnets, route tables, and network ACLs to manage traffic flow. VPC provides a secure and isolated environment for your resources.

Route 53: Route 53 is a domain name system (DNS) service that allows you to route traffic to your resources. You can create hosted zones, record sets, and health checks to manage traffic flow. Route 53 provides high availability and durability for your DNS needs.

CloudFront: CloudFront is a content delivery network (CDN) service that allows you to distribute content to your users. You can create distributions, origins, and cache behaviors to manage content delivery. CloudFront provides high performance and low latency for your content delivery needs.

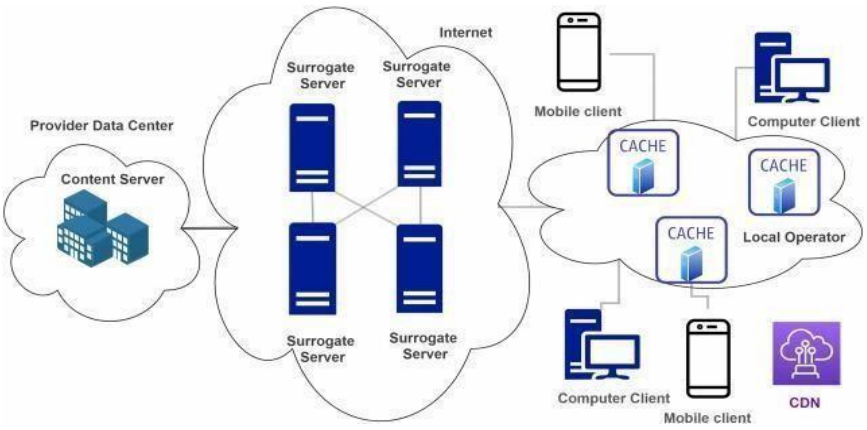


Fig 2.3 Content Delivery

2.4 Cloud Economics and Billing:

Cloud economics refers to the financial management and cost optimization of cloud computing resources. It involves understanding the costs associated with cloud computing, such as usage-based pricing, tiered pricing, and discounts for committed usage. By optimizing cloud economics, organizations can maximize the value of their cloud investments and achieve cost savings. Cloud economics involves analyzing usage patterns, rightsizing resources, and selecting the most cost-effective pricing models. It also involves taking advantage of discounts, such as reserved instances and spot instances, and using cost allocation tags to track costs.

Billing refers to the process of generating and managing invoices for cloud computing services. In the context of cloud economics, billing is a critical component as it directly impacts an organization's cost management and optimization strategies.

Cloud providers offer various billing models, including:

- Usage-based billing: Charging customers based on their actual resource usage.
- Tiered pricing: Offering discounts for higher usage levels.
- Reserved Instances: Discounted pricing for committed usage.
- Spot Instances: Bid on unused capacity for discounted pricing.

Fig 2.4 Billing and Pricing

2.5 AWS Cloud Security and compliance:

AWS Cloud Security and Compliance refer to the practices and controls used to protect and secure data and applications in the AWS cloud. AWS provides a secure infrastructure and services, but it's the customer's responsibility to ensure their data and applications are secure and compliant with relevant regulations.

2.5.1 Security:

Security in the AWS cloud refers to the practices and controls used to protect and secure data, applications, and infrastructure from unauthorized access, use, disclosure, disruption, modification, or destruction. AWS provides a secure infrastructure and services, but it's the customer's responsibility to ensure their data and applications are secure.

Some key security features and services in AWS include:

- Identity and Access Management (IAM): manages access to AWS resources
- Virtual Private Cloud (VPC): provides a secure network environment
- Security Groups: controls access to instances
- Network ACLs: controls access to subnets
- Encryption: protects data at rest and in transit
- Key Management Service (KMS): manages encryption keys
- CloudWatch: monitors and logs security-related events

2.5.2 Compliance:

Compliance in the AWS cloud refers to the process of ensuring that your cloud resources and data meet specific regulatory, industry, or organizational requirements. AWS provides various compliance programs and certifications to help customers meet these requirements.

Some key compliance features and services in AWS include:

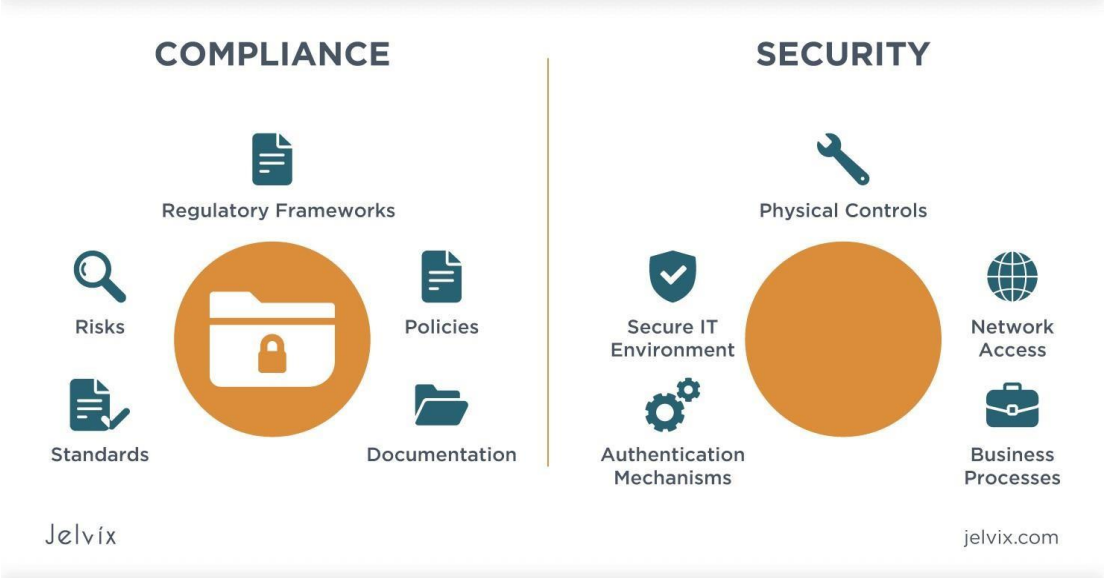
- Compliance frameworks: AWS supports various compliance frameworks such as PCI-DSS, HIPAA/HITECH, GDPR, and more

Certifications: AWS holds various certifications such as SOC, ISO, and PCI-DSS

Security and compliance controls: AWS provides security and compliance controls such as IAM, VPC, and encryption

Audit and logging: AWS provides audit and logging capabilities such as CloudWatch and CloudTrail

Compliance monitoring: AWS provides compliance monitoring capabilities such as Inspector and Config.



n

Fig 2.5 Compliance and Security

2.6 Auto Load Balancing:

Auto Load Balancing in AWS is a feature that automatically distributes incoming traffic across multiple targets, such as EC2 instances, containers, or IP addresses, to ensure high availability and scalability. AWS offers several types of load balancers, including Application Load Balancer (ALB), Network Load Balancer (NLB), and Classic Load Balancer (CLB), each suitable for different types of traffic and use cases. By using Auto Load Balancing, applications can ensure high availability, scalability, and improved responsiveness, as traffic is automatically redirected away from unhealthy targets and adjusted to changes in traffic volume.

To set up Auto Load Balancing, users create a load balancer, configure settings and targets, and enable health checks and Auto Scaling.

Load balancing options to support different use cases and requirements. By leveraging Auto Load Balancing, users can ensure their applications are always available and responsive, even in the face of changing traffic demands.

Auto Load Balancing in AWS provides several benefits, including:

1. High Availability: Ensures that applications are always available and accessible.

2. Scalability: Automatically adjusts to changes in traffic volume.

3. Fault Tolerance: Detects and redirects traffic away from unhealthy targets.

4. Improved Responsiveness: Reduces latencyand improves application responsiveness.

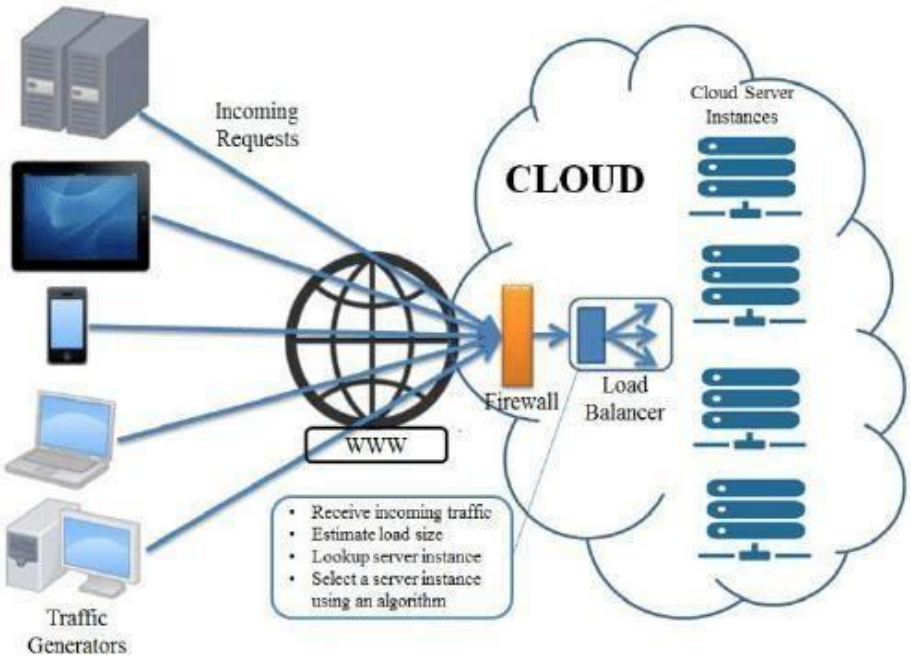


Fig 2.6 Working of Auto Load Balancing

To set up Auto Load Balancing in AWS, follow these steps:

1. Create a load balancer (ALB, NLB, or CLB).

2. Configure the load balancer with the desired settings (e.g., protocol, port, and targets)

3. Add targets (EC2 instances, containers, or IP addresses) to the load balancer.

4. Configure health checks to monitor target health.

5. Enable Auto Scaling to adjust the number of targets based on traffic demand.

CHAPTER 3

AWS Data Engineering

3.1 Introduction to Data Engineering:

When integrating **data analysis** and **AI/ML (Artificial Intelligence and Machine Learning)** approaches into **Customer Relationship Management (CRM)**, the focus is on optimizing customer interactions, improving customer satisfaction, and ultimately increasing business value. Here's a breakdown of how both approaches can be applied using real-world examples:

1. Data Analysis

Data analysis refers to examining data using various tools and techniques to extract actionable insights, trends, and patterns. In the context of CRM, data analysis helps businesses understand customer behaviour, identify market trends, and optimize marketing and sales strategies.

Key Steps in Data Analysis for CRM:

- **Data Collection**: Gather data from multiple customer touchpoints, including website interactions, social media, email communications, and purchase history.
- **Data Cleaning and Preparation**: Ensure that customer data is accurate, consistent, and organized. This step involves handling missing data, duplicates, and ensuring proper data formatting.
- **Exploratory Data Analysis (EDA)**: Perform statistical analysis to uncover trends and insights. This could involve:
    - **Customer Segmentation**: Divide customers into meaningful groups based on demographics, purchasing behaviour, and engagement levels.
    - **Churn Analysis**: Analyse patterns in customer data to identify factors that may lead to customer attrition.
    - **Lifetime Value Prediction**: Calculate the expected revenue a customer will generate over their relationship with the company.

- **Visualization**: Create dashboards and reports to visualize customer trends, behavior patterns, and KPIs such as sales performance, retention rates, and engagement metrics.

Example:

A retail company uses data analysis to understand customer purchase behaviour. By segmenting customers based on demographics, purchase history, and loyalty program engagement, they identify that a certain segment responds well to personalized discounts. As a result, they target that group with a specialized marketing campaign, increasing sales by 15%.

2. AI & ML Approach in CRM

AI and ML take CRM to a higher level by automating processes, predicting customer behaviour, and personalizing interactions at scale. Machine learning models can learn from historical customer data to make predictions and provide real-time insights.

Example:

A telecom company uses machine learning to predict customer churn by analysing historical data such as call logs, service usage, and complaint frequency. The model identifies at-risk customers, and the CRM system automatically sends them special retention offers (e.g., discounts or service upgrades). The company reduces churn by 20%, significantly increasing profitability.

Combining Data Analysis and AI/ML in CRM:

Both approaches can work together to enhance CRM capabilities. For example:

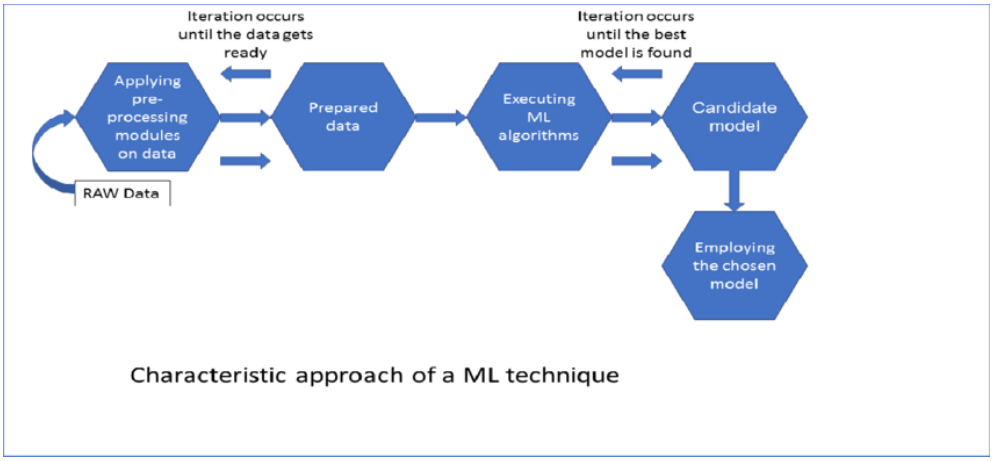
- **Data Analysis** can identify important metrics, trends, and customer segments, which can then be fed into **AI/ML models** for automation, personalization, and predictions.
- **AI/ML Models** can continuously improve based on feedback loops, incorporating insights from traditional data analysis.

Combined Example:

A financial services firm uses data analysis to segment its customer base by transaction history, demographics, and engagement levels. The AI/ML system then provides personalized investment advice to each customer segment and predicts their likelihood of upgrading to premium services. Meanwhile, the CRM system tracks customer satisfaction and adjusts service offerings in real-time. This results in a 25% increase in customer retention and a 10% increase in premium service sales.

Benefits of AI & ML in CRM:

- **Scalability**: AI systems can handle large volumes of customer interactions, unlike manual data analysis.
- **Real-time insights**: ML algorithms can process data in real-time, enabling businesses to react faster to customer needs.
- **Improved personalization**: AI allows hyper-personalization, which increases customer satisfaction and loyalty.
- **Automation**: Repetitive tasks such as data entry, customer follow-ups, and basic inquiries are automated, freeing human agents for more complex tasks.



3.1 ML Approach in CRM

3.2 Data-Driven Organizations:

Data Pipeline:

A data pipeline is a method in which raw data is ingested from various data sources, transformed and then ported to a data store, such as a data lake or data warehouse, for analysis. Before data flows into a data repository, it usually undergoes some data processing.

A **data pipeline infrastructure** is a system that facilitates the movement, transformation, and processing of data from its source to its destination. It typically consists of multiple layers, each serving a specific role to ensure that data is ingested, processed, stored, and ultimately made available for analysis or further use. Here's a breakdown of the main layers in a data pipeline infrastructure:

A **data pipeline infrastructure** consists of several key layers, each with a specific function:

1. **Data Ingestion**: Collects data from various sources, either in batches or in real-time (streaming).
    - Tools: Apache Kafka, AWS Glue.
2. **Data Buffering**: Temporarily holds data for smooth processing, using queuing or buffering systems.
    - Tools: Apache Kafka, RabbitMQ.
3. **Data Processing**: Transforms and cleans data (ETL/ELT), in batch or real-time.
    - Tools: Apache Spark, Apache Flink.
4. **Data Storage**: Stores processed data in data lakes for raw data or data warehouses for structured data.
    - Tools: AWS S3, Snowflake, Redshift.
5. **Data Query/Analytics**: Allows querying and analysing processed data for insights.
    - Tools: Google Big Query, Tableau.

6. **Data Orchestration**: Automates and manages the flow and dependencies in the pipeline.
    - Tools: Apache Airflow, Prefect.
7. **Data Governance & Security**: Ensures data quality, compliance, and security across the pipeline.
    - Tools: Apache Atlas, encryption tools.
8. **Monitoring & Logging**: Tracks performance and logs key pipeline events for reliability.
    - Tools: Prometheus, ELK Stack.
9. **Data Consumption**: Provides data for end-users, applications, or systems for insights and actions.
    - Tools: Jupyter, business intelligence tools.

Each layer ensures efficient, secure, and reliable data processing from collection to consumption.

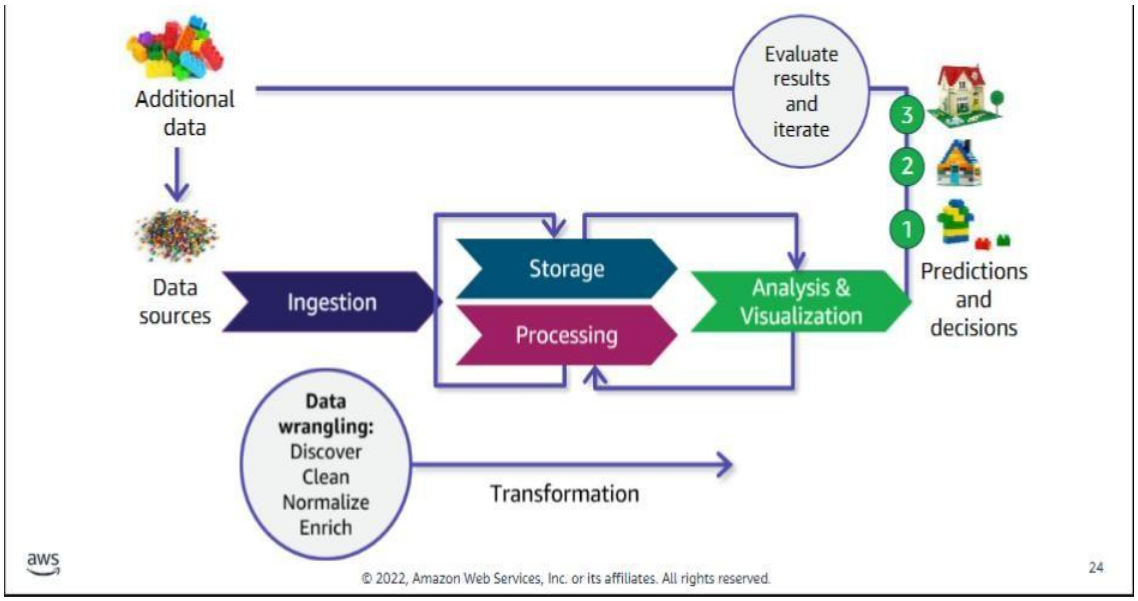


Fig 3.2 Data Pipeline

3.3 Design Principles and Patterns for Data Pipelines:

3.3.1 : Data Lake:

A data lake is a centralized repository that stores and processes large amounts of data in its native format. Data lakes can store any type or volume of data, including structured, semi-structured, and unstructured data. They can be used to power big data analytics, machine learning, and other forms of intelligent action.

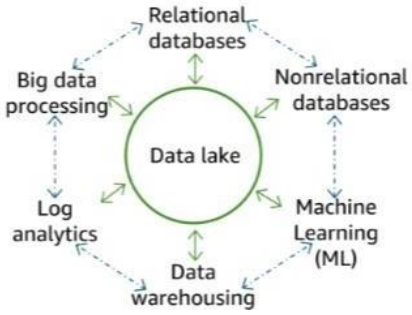


Fig 3.3 Data Lake

3.3.2 Processing and consumption layers in the reference architecture:

The processing and consumption layers in the modern data architecture prepare data and make it available to consumers. The consumption layer equates to the analysis and visualization layer of a data pipeline. This reflects that the data available in the pipeline can be used and consumed in a variety of ways either by end users or by other downstream systems that consume outputs of data processing.

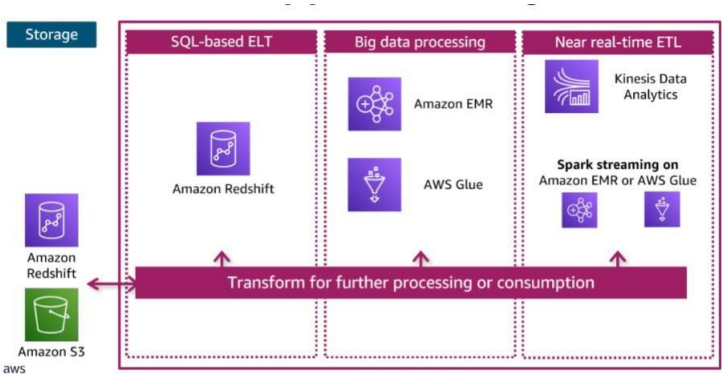


Fig 3.4 Modern Architecture Pipeline

3.4 Processing Data for ML:

Automating The dataset that is needed to build an ML model for supervised learning must provide enough quality labelled data to support both training and test datasets. The training dataset helps the model to learn the feature patterns in the data to make predictions. The test dataset helps the data scientist to validate the model and optimize the quality of the predictions. After the model has been trained and tested, it is deployed to production, where it runs against the production data source that you want to make decisions from. Feedback from production provides additional information about the accuracy of the model, and might identify issues with the model or the quality of the dataset that you used to train it.

The data engineer would not be responsible for deciding how to partition the dataset for training and test data. The data scientist would decide how to partition the data based on data characteristics and the model that is being developed. The source dataset must be large enough to account for the amount of data that needsto be labelled to train the model and the amount of test data that is needed to tune the model. The volume and quality the available data for training and tuning are importantfactors in the quality of the predictions that can be made, and will vary within the type of data and modelling.

Typically, 70 to 80 percent of the collected data is used to train the model. The remaining data is used to test and tune the model. After the model is deployed to production, the actual results can also be used to train and tune the model.

3.5 Analyzing and Visualizing Data:

A key aspect of getting insight from your data is finding patterns. Patterns are often much easier to see in a graph or chart rather than staring at data in a table. The right visualizationwill help you gain a deeper understanding in a much quicker timeframe. A visualization might beproduced as part of a report, or it might be used as part of an interactive dashboard where a user can drill down on something interesting. Visualizations highlight one of the following five types of insights:

•Key Performance Indicators (KPIs) which are usually a single variable that measures how wellyou are doing. For example how many sales leads become sales.

• Relationships between two variables, for example, whether or not sales revenue is tied tomarketing spending.

• Comparisons of how different variables change over time, for example showing month overmonth sales and web traffic.

•Distributions of data over certain clusters or groups, for example grouping customers by thenumber of purchases they've made.

•Compositions of elements that make up your data, for example, sales by region.

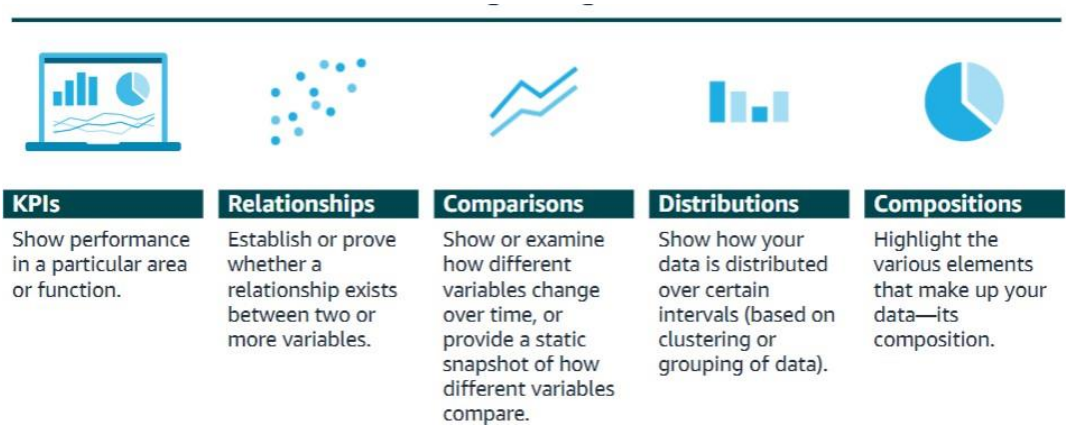


Fig 3.5 Visualizing insights

3.6 Automating the Pipeline:

A pipeline is a process that drives software development through a path ofbuilding, testing, and deploying code, also known as CI/CD. By automating the process, the objective is to minimize human error and maintain a consistent process for how software is released.

AWS Glue Crawler:

AWS Glue Crawler is a managed service in Amazon Web Services (AWS) that automatically discovers and catalogs data stored in various sources, such as Amazon S3, Amazon RDS, and other AWS databases. It is part of the AWS Glue ecosystem, which isdesigned for data integration, preparation, and ETL processes. Crawlers scan the data in your storage locations, infer its schema, and then populate the AWS Glue Data Catalog.

AWS Lambda:

AWS Lambda is a serverless computing service that lets you run code without provisioning or managing servers. You can execute functions in response to events like fileuploads to Amazon S3, updates to a DynamoDB table, or HTTP requests via API Gateway. Lambda automatically scales based on the incoming requests, and you only pay for the compute time you consume. It supports a variety of programming languages such as Python, Node.js, Java, and Go. AWS Lambda is ideal for building lightweight microservices, real-time file processing, and event-driven applications.

AWS Athena:

Amazon Athena is a serverless, interactive query service that allows you to analyze data stored in Amazon S3 using standard SQL. It is highly scalable, with no need for infrastructure setup, and you only pay for the queries you run. Athena is often used for querying structured, semi-structured, and unstructured data, such as logs, CSV, JSON, andParquet files. It integrates seamlessly with AWS Glue for managing metadata and performing ETL operations. This makes it an efficient tool for ad-hoc data analysis and querying large datasets without the overhead of managing a database.

Amazon SNS:

Amazon SNS (Simple Notification Service) is a fully managed messaging service that enables the exchange of notifications between applications or directly to users via SMS, email, or mobile push. It supports the pub/sub (publish/subscribe) messaging model, allowing you to send messages to multiple subscribers simultaneously. SNS is often used for event-driven architectures, sending alerts, or triggering automated workflows. It integrates seamlessly with other AWS services like Lambda, SQS, and CloudWatch. Withits high scalability and reliability, Amazon SNS ensures timely message delivery across distributed systems.

Fig 3.6 Simplifying ETL

CHAPTER-4

REAL TIME APPLICATIONS OF DATA ENGINEERING

Data Engineering plays a critical role in real-time applications where large amounts of data must be ingested, processed, and analyzed quickly. Here are some real-time applications of data engineering.

Fig 4.1 Applications of data Engineering

1. Fraud Detection Systems:

Fraud detection systems in industries like finance and e-commerce. These systems analyze transactions in real-time to detect suspicious activity, allowing businesses to prevent fraudulent activities before they escalate. Data engineers build complex pipelines that continuously ingest transactional data, applying machine learning models that assess patterns indicative of fraud. By leveraging technologies such as Apache Flink and Kafka, these pipelines ensure rapid processing and instant action.

2. Real-Time Recommendation Engines:

It is widely used by companies like Amazon and Netflix. These systems provide personalized recommendations to users based on their browsing behavior and past interactions. Data engineers build real-time data pipelines that capture and analyze user activities, feeding this data into machine learning algorithms that generate recommendations within seconds. Tools like Apache Spark Streaming and Kafkaenable these engines to process data in real-time, enhancing customer engagement and increasing sales or viewership.

3. Predictive Maintenance:

Industries such as manufacturing and transportation use real-time data engineering to prevent equipment failures. By continuously monitoring sensor data from machinery, data pipelines can identify signs of wear or malfunction and trigger maintenance alerts. This proactive approach reduces downtime and saves costs, with technologies like Apache Flink and IoT platforms enabling the rapid processing of sensor data for predictive insights.

4. Stock Market Analytics and Trading:

In the **stock market**, real-time data engineering is essential for high-frequency trading and financial analytics. Stock prices and market data change rapidly, and trading algorithms need to process this data within milliseconds to capitalize on opportunities. Data engineers design pipelines that ingest market data streams and feed them into trading systems, using tools like Apache Storm and Kafka to ensure low-latency processing.

5. Real-Time Customer Support Chatbots:

The Application is **real-time customer support chatbots**, which have become ubiquitous in retail and customer service industries. These chatbots process user queries in real-time, retrieving information from backend systems and responding instantly. Data engineers ensure that the data pipelines integrate seamlessly with chat interfaces, leveraging real-time technologies like AWS Lambda and Google Cloud Functions.

6. Traffic Management and Optimization:

Smart cities and transportation systems use real-time data pipelines to optimize traffic flow. Data from sensors, GPS devices, and cameras is continuously processed to make immediate decisions that reduce congestion and improve travel times. By utilizing technologies like Apache NiFi and Kafka, these systems can react in real-time to changing traffic conditions, making cities more efficient and sustainable.

7. Health Monitoring Systems:

Health monitoring systems are another vital application of real-time data engineering, particularly in healthcare and wearables. Devices like smartwatches collect health data such as heart rates and activity levels, which are processed in real- time to detect anomalies. If irregularities are found, such as a sudden drop in heart rate, the system can immediately alert both the wearer and healthcare providers. Kafka and IoT platforms are critical in processing this data rapidly to ensure timely intervention.

CHAPTER 5
LEARNING OUTCOMES OF INTERNSHIP

1. Gain a deep understanding of aws and data types, including its benefits, deployment models (Structured, Unstructured, Semi-Structured).

2. Understand how to create, configure, and manage AWS resources using the AWS Management Console and AWS CLI.

3. Learn how to design and implement cloud architectures that are scalable, resilient, and cost-effective, following AWS best practices.

4. Understand AWS security features, such as encryption, security groups, and IAM policies, and how to implement them to secure cloud environments.

5. Understand strategies for data Pipelines, Data Lakes, Processing and visualizing the data.

6. Understand about the Automating the Pipeline and the types Amazon SNS.AWS Lambda, Amazon S3, AWS Glue Crawler

7. Explore emerging technologies in the cloud space, such as artificial intelligence (AI), machine learning (ML), Internet of Things (IoT), and edge computing, and how they integrate with AWS services.

CHAPTER 6
CONCLUSION

In conclusion, AWS Data Engineering internship has been an invaluable learning experience, providing me with both theoretical knowledge and practical skills in different processing, analyzing technologies. Throughout the internship, I gained a deep understanding of concepts and the extensive suite of services offered by AWS. By working on real-world projects, I learned to design and deploy scalable, secure,and cost-effective cloud solutions, applying best practices in cloud architecture and security.

I also developed a strong foundation in DevOps practices and automation, enabling me to manage and optimize cloud resources efficiently. The hands-on experience with AWS tools and services, combined with exposure to emerging technologies like AI, machine learning, and IoT, has broadened my technical expertise and prepared me for future roles in cloud computing.

This internship has significantly enhanced my problem-solving, collaboration, and communication skills, which are crucial for success in any technology-driven environment. As I move forward in my career, the knowledge and experience gained during this internship will serve as a solid foundation for pursuing advanced roles in cloud computing and contributing to innovative cloud-based solutions in the industry.

REFERENCES

- ✓ **https://docs.aws.amazon.com/prescriptive-guidance/latest/aws- caf-platform-perspective/data-eng.html**
- ✓ **https://awsacademy.instructure.com/courses/81188**
- ✓ **https://awsacademy.instructure.com/courses/70557/modules**
- ✓ **https://aws.amazon.com/training/awsacademy/**