

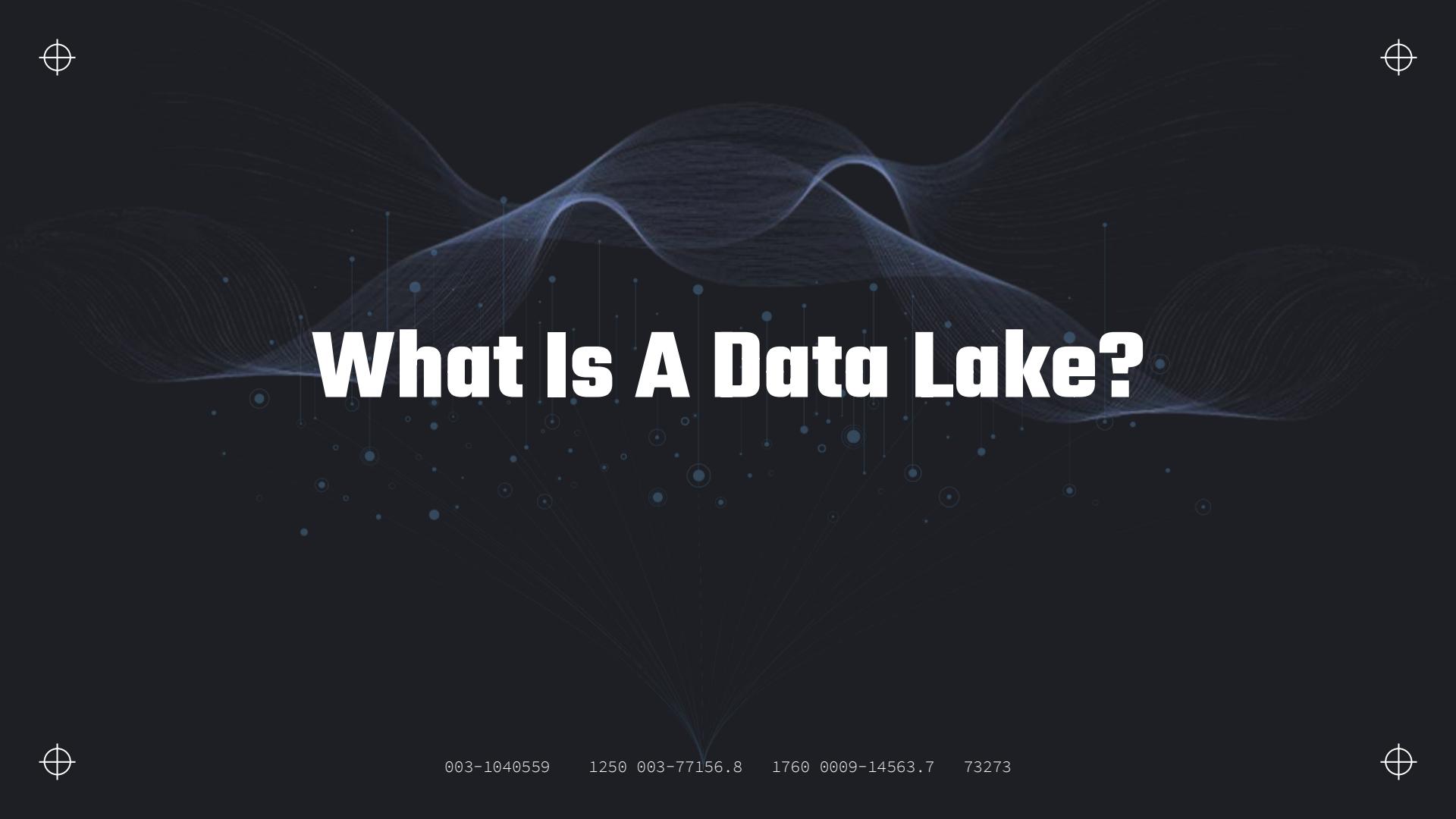
Section 1:

Introduction

003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



What Is A Data Lake?

003-1040559

1250 003-77156.8

1760 0009-14563.7 73273





01

02

03

04

05

06

Introduction to Data Lake

- Explore fundamental concept
- Foundation for modern data storage solutions
- Crucial for mastering data engineering

01

02

03

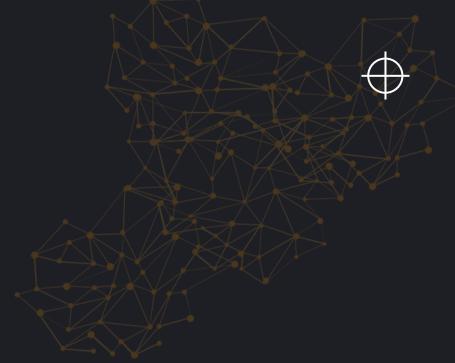
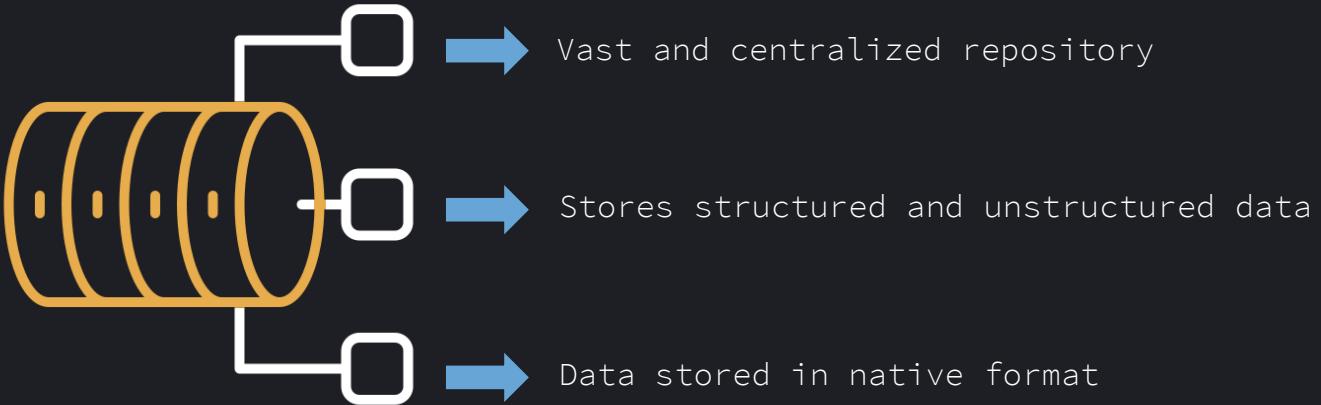
04

05

06



What is a Data Lake?





Evolution of Data Lakes

BIG DATA

Roger Magoulas
(2005)

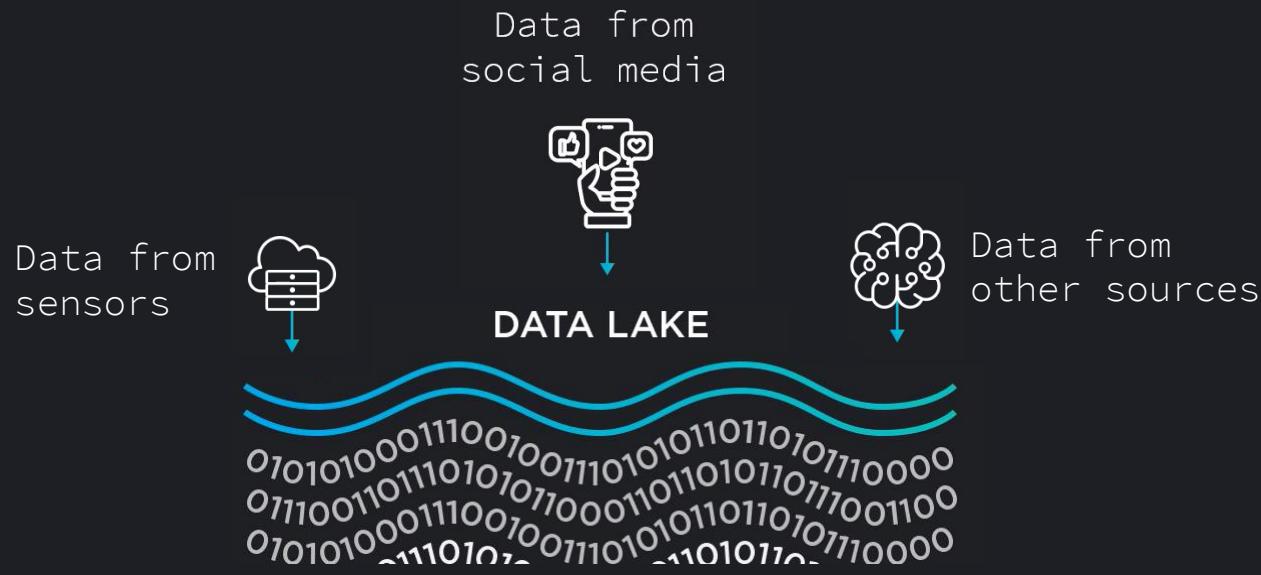
- Manage Large Data Volumes
- SQL tools struggle to manage and analyze data.

DATA LAKE

James Dixon
CTO of Pentaho
(2010)



Data Sources for Data Lakes





Data Explosion Demands Flexibility

James Dixon "Large body of water in a more natural state"





Comparison with Data Warehouse



- Data lakes represent a paradigm shift.
- Departure from the rigid structure of a data warehouse
- Designed for the age of massive and diverse data.

01

02

03

04

05

06

Upcoming Lecture: Dives into the advantages of leveraging data lakes.





Why Use A Data Lake?



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



01

02

03

04

05

06

Evolutionary Response to Big Data

- Data lakes evolved in response to the big data explosion.
- Stored semi-structured, unstructured, and structured data.
- Served as a flexible repository



01

02

03

04

05

06



Benefits of a Data Lake

- They provide:
 - ✓ Centralized
 - ✓ Flexible
 - ✓ Scalable
 - ✓ Cost Effective

Solution to store and analyze huge amounts of data





Centralization & Accessibility





Centralization & Accessibility

Data lake centralizes data



01010100011100100100111010101101010110101011000
0111001101110101010110001101101011010111001100
01010100011100100100111010101101010110101100000



Democratizes data access
within organizations





Scalability



Product Catalog



Growing Customer





Scalability



Data lake designed to handle petabytes of data



Flexibility



Diverse Data
Formats



IoT devices





Flexibility



Diverse Data Formats



IoT devices



Data stored in its native format.



Data Engineers And Data Scientists



They can retrieve data make transformations on the fly



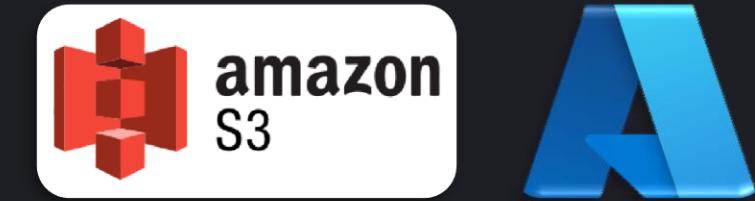


Cost-Effectiveness





Cost-Effectiveness



Cloud platforms offer cost-effective storage solutions





Data Swamp





Data Swamp



WITHOUT

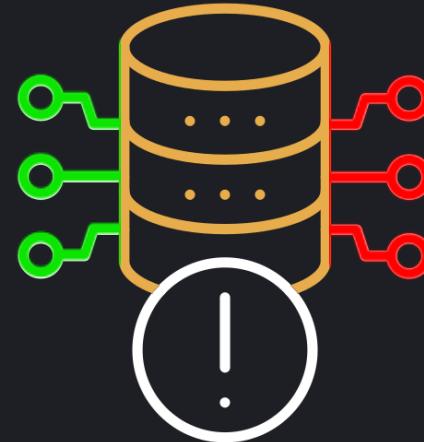
- Adequate Metadata
- Quality Checks
- Organization



Data Swamp

ENSURE

- Strict Data Governance
- Quality Control
- Use Metadata Management
- Use Data Catalog



WITHOUT

- Adequate Metadata
- Quality Checks
- Organization

Diving deeper into important concepts in the next lecture.



Key Concepts & Terminology



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Key Concepts & Terminology

Data Lake

- Pool of raw data stored in its native format.
- Contrasts with data warehouses.



Key Concepts & Terminology

Metadata

- **Data** about data.
- Essential for navigating large volumes of data.
- Includes origin, generation time, collection method, and format.
- Crucial for effective data retrieval and usage.
- Effective metadata management prevents a data lake from becoming a **Data Swamp**.



Key Concepts & Terminology

Data Swamp

- Warning label
- Occurs when data is not categorized
- Need for proper data governance



Key Concepts & Terminology

Data Governance

- Prevents data lakes from becoming swamps.
- Includes:
 - Policies
 - Metadata management practices
 - Processes for data collection



Key Concepts & Terminology

Data Catalog

- Helps locate and manage assets
- Organized inventory



ech

Key Concepts & Terminology

ETL vs. ELT

- ETL **Extract ➔ Transform ➔ Load**
 - Traditional approach
 - Preparing data before loading into data warehouses
- ELT **Extract ➔ Load ➔ Transform**
 - Common in the context of data lakes
 - Loaded first in its raw form, then transformed



Key Concepts & Terminology

Structured, Semi-Structure and Unstructured Data

- Structured Data:
 - Fits easily into tables or spreadsheets with fixed schema.
- Unstructured Data:
 - No fixed schema (e.g., images, videos, PDFs).
- Semi-Structured Data:
 - Embraces diversity, includes data like JSON files.



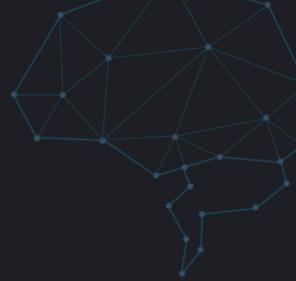
Key Concepts & Terminology

Schema-on-Read vs. Schema-on-Write

- Schema on Write (Traditional Approach):
 - Data structured when written into the database.
- Schema on Read (Data Lake Approach):
 - Data structure defined when the data is read.
 - Offers flexibility and agility for diverse data needs.



Conclusion



- Entire ecosystem involves ingesting, processing, and storing data.
- Data lake lies in its potential to serve as a flexible, scalable, and cost-effectiveness data storage and analytics solution.

**Next lecture will explore distinctions between
Data Lake, Data Warehouse, and Lakehouse**



Data Lake vs. Data Warehouse vs. Lakehouse



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273





Data Lakes

Data lakes are expansive repositories that store massive volumes of raw data in its native format.



PROS

- Scalability
- Flexibility
- Cost-effective

CONS

- Complexity in managing raw data
- Requires robust governance
- Security risks

USE CASE:

Storing diverse data for potential future use





Data Warehouse

Data warehouses are structured and highly organized. They store processed, filtered, and defined data, tailored for specific uses.



PROS

- Efficient
- Fast data retrieval
- Well-organized data
- Mature technology

CONS

- Less flexibility
- More expensive
- Rigid structure

USE CASE:

Ideal for structured data, reporting, and interactive dashboards.



Lakehouses

It's offer the structured querying capability of data warehouses with the raw data storage flexibility of data lakes.



PROS

- Versatility
- Real-Time Processing
- Scalability

CONS

- Complex Architecture
- Emerging Technology
- Integration Challenges

USE CASE:

A hybrid solution that provides structured querying with raw data storage flexibility.





Data Lake

Ideal for diverse, raw data storage and potential future use.



Data Warehouse

Efficient for structured data, reporting, and specific use cases.

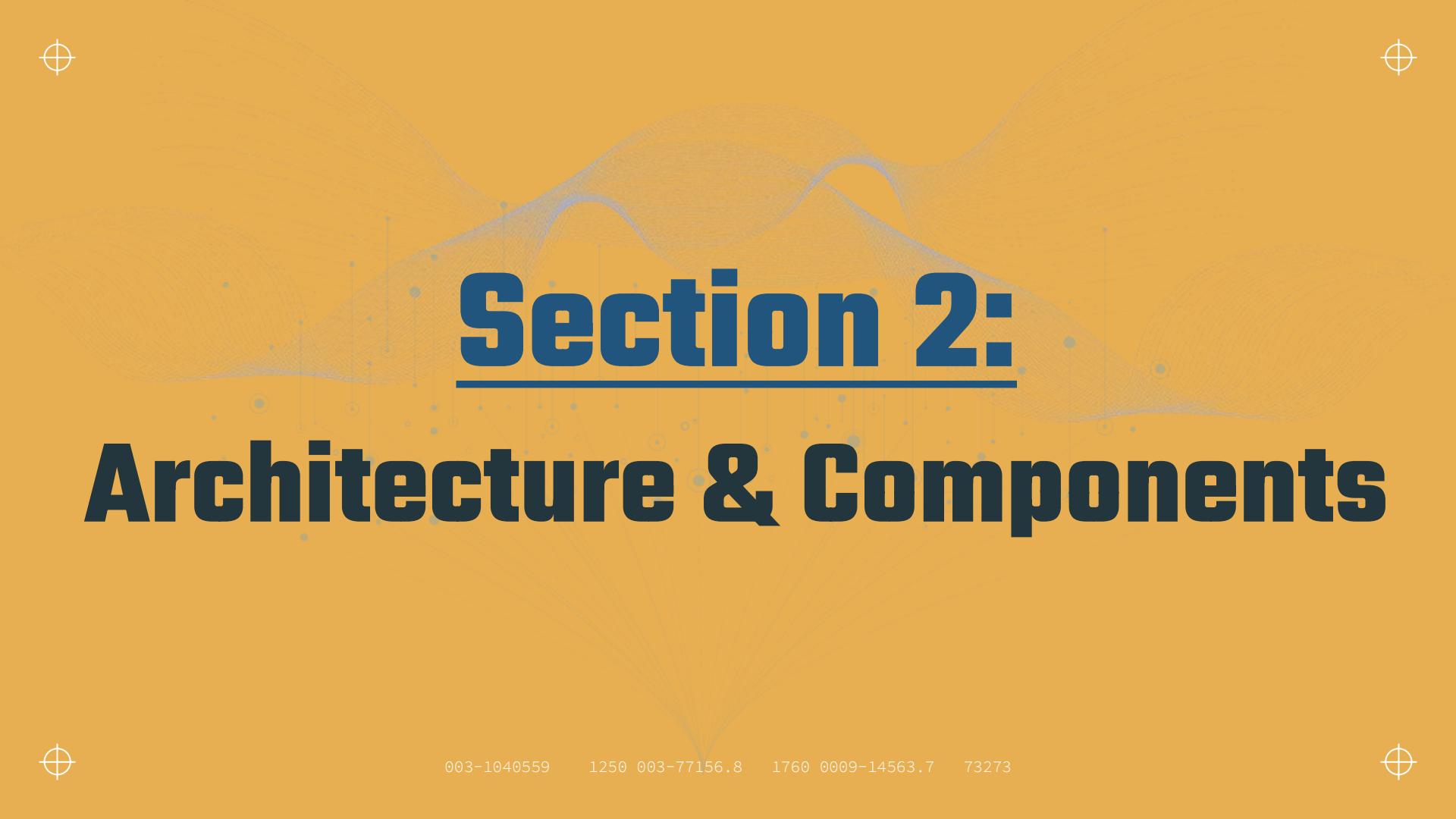


Lake House

A hybrid solution combining aspects of both data lakes and data warehouses.



let's get started with our AWS setup



Section 2:

Architecture & Components



Designing a Data Lake Architecture



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273





Overview:

- High-Level Architectural Components
- Data Flow in a Data Lake
- Requiring careful management for data lake success





Storage

- Scalable
- Robust Storage
- Capable of storing petabytes
- Amazon AWS provides a reliable solution with:
 - Durability
 - Availability
 - Scalability



Processing Capacity

- Robust processing capability
- Involves large-scale data transformations
- Utilizing the data lake



Governance

- Data integrity
- Security
- Spans across all aspect like data ingestion, security etc.
- Ensures responsible data management



Metadata Management

- Detailed catalog system
- AWS Glue used for data catalog
- Enhances data searchability



User Access

- Ensuring appropriate user access with authentication
- Architecture must include:
 - Authentication
 - Authorization
- Managed in AWS using **IAM** (Identity and Access Management)



Orchestration

- Coordination and management of data processing in more complex scenarios
- Involves designing, scheduling
- Ensures a coordinated workflow
- Amazon AWS provides a reliable solution with durability, availability, and scalability.



AWS Glue



IAM



High-Level Data Flow in a Data Lake



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273





Introduction

- Understand the journey of data
- Complex cycle
- Establish a data flow

01

02

03

04

05

06



01

02

03

04

05

06



Data Ingestion

Initial step, importing data from diverse sources

01

Sources

- IOT devices
- Online platforms
- Relational data
- Mobile apps

02

Data Acceptance

- Accepts:
- Shapes
- Sizes (structured, semi-structured, unstructured)

03

04

05

06



Data Storage

It's is a base of the data lake

01

Format

02

Schema on
Read

03

- Stored in raw format
- Schema imposed upon reading or processing

04

1

2

3

4

5

6

05

06



Metadata Management

Tagging data with content, source, format.

01

Importance

02

- Increases with the amount of loaded data

Purpose

03

- Organize data library
- Enhance usability
- Facilitate data discovery

04

05

06



Data Governance

Ensures quality and compliance

01

Components

02

- Involves data privacy
- Security measures
- Compliance with regulations

Framework

03

- Establishing Data Lake that is:
 - Secure
 - Compliant
 - Quality-driven

04

05

06



Data Consumption

Final step where business intelligence tools come into play

01

Goal

02

- Users access curated data
- Perform analysis
- Generate reports
- Visualizations
- Derive insights

Examples

03

- Power BI
- Data analytics platforms
- Machine learning models

04

For practical understanding, let's implement this with specific tools into more practice

05

5

6

7

8

9

10

06





Different Zones in Data Lake



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Overview

- Single-layer vs. Multi-layer (with multiple zones)
- Multi-layer recommendation:
 - ✓ Structured organization
 - ✓ Quality control
 - ✓ Enhanced governance

01

02

03

04

05

06

01

02

03

04

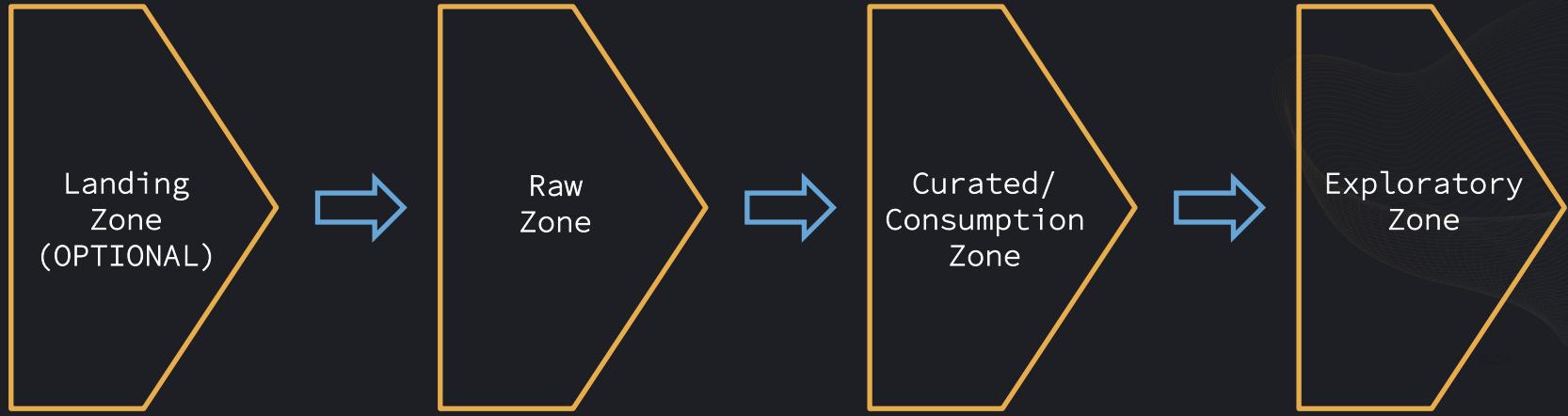
05

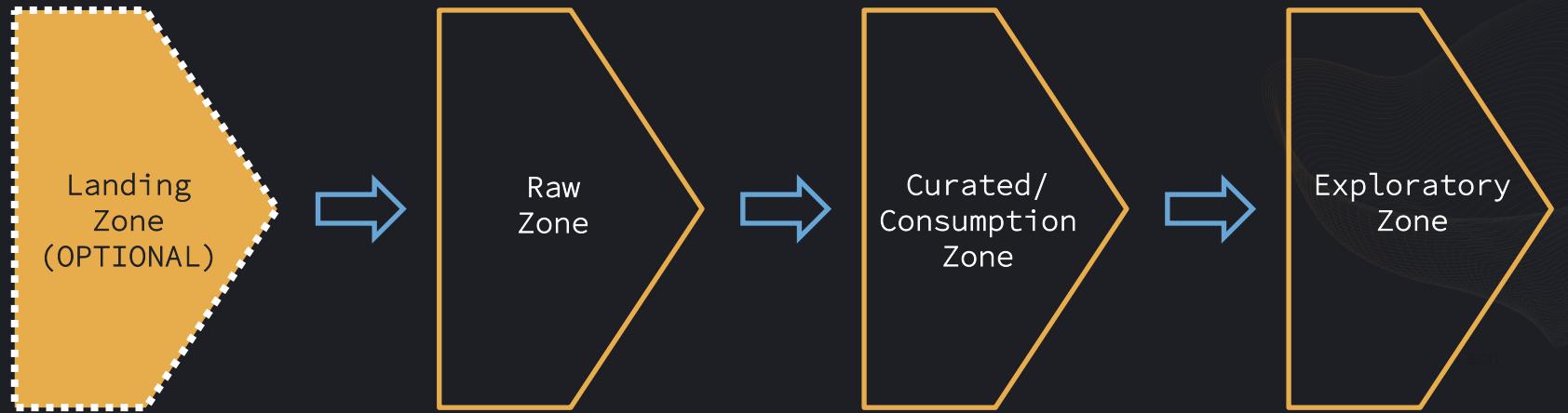
06



Planning Stage

- **Importance:**
Plan structure before data arrival.
- **Flexibility:**
It is not set in stone, structure can be adjusted.





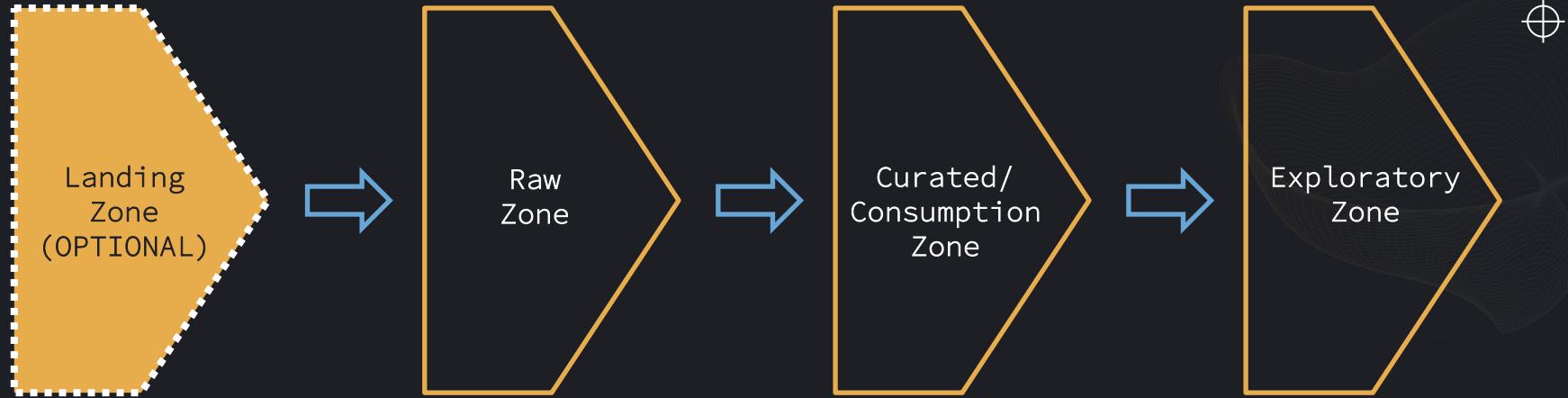
“Acting as a transient layer”

- Preserve data in:

- ✓ Native format
- ✓ Add metadata
- ✓ Timestamps
- ✓ Basic validation



- AWS S3 bucket with folders per source system



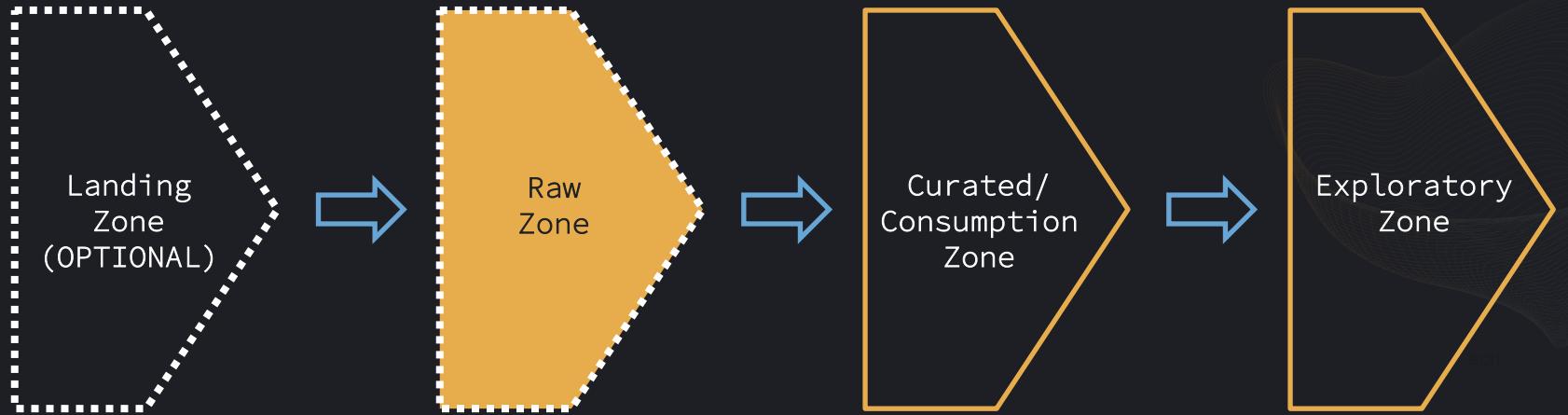
“Acting as a transient layer”

- Preserve data in:

- ✓ Native format
- ✓ Add metadata
- ✓ Timestamps
- ✓ Basic validation

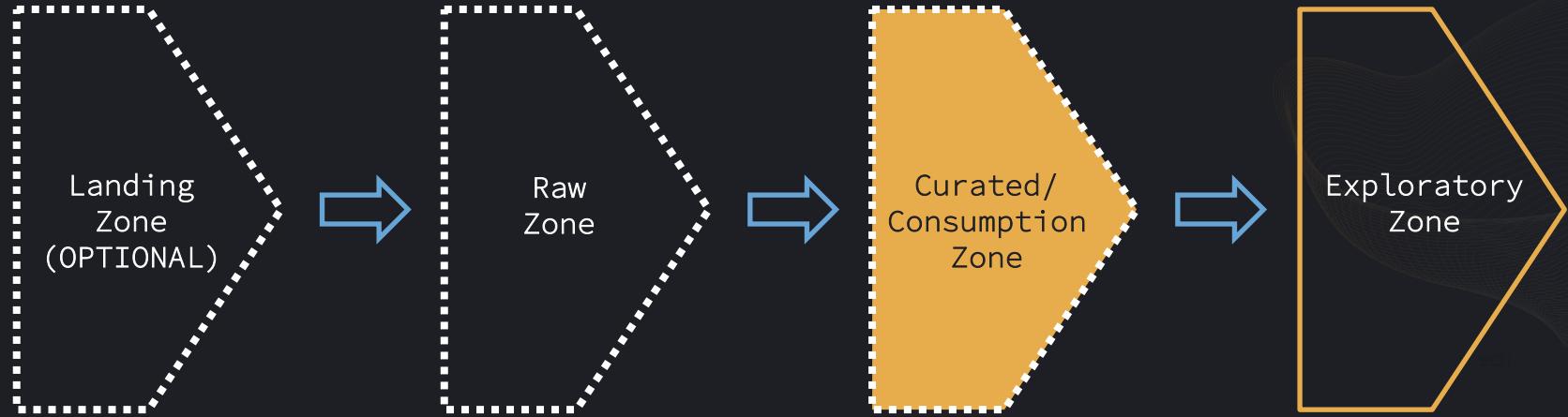


- AWS S3 bucket with folders per source system



- Contains **quality-checked** data
- No **pre-defined** structure
- **Schema** applied on reading
- Available to **users**
- Maintaining **raw format**

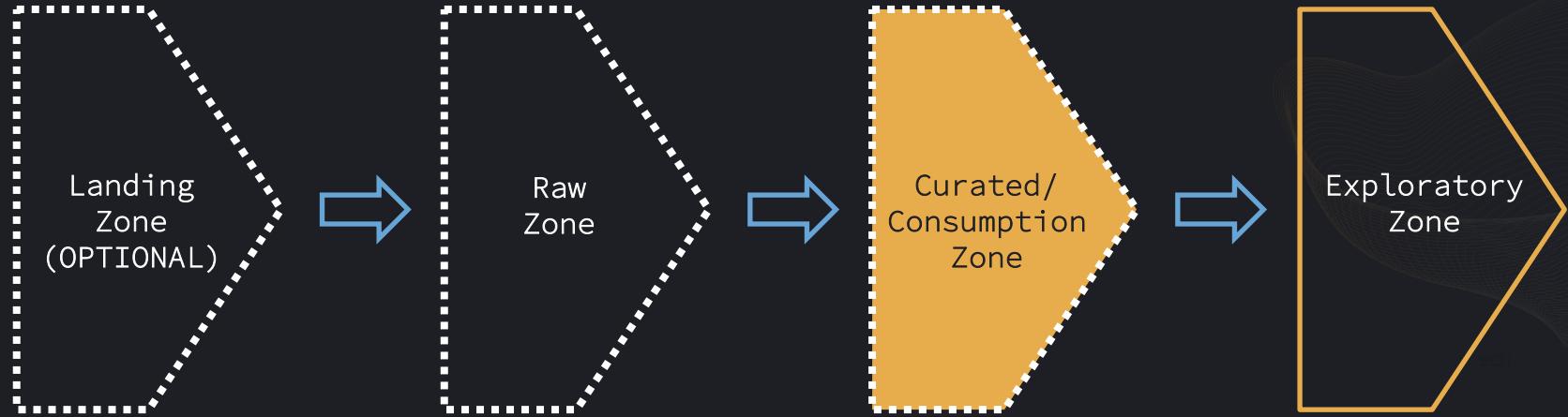




“Main consumption area for analysts, data scientists, and BI tools.”

- Data organized and optimized for query performance
- Access control for different users
- Data can potentially be enriched with additional features

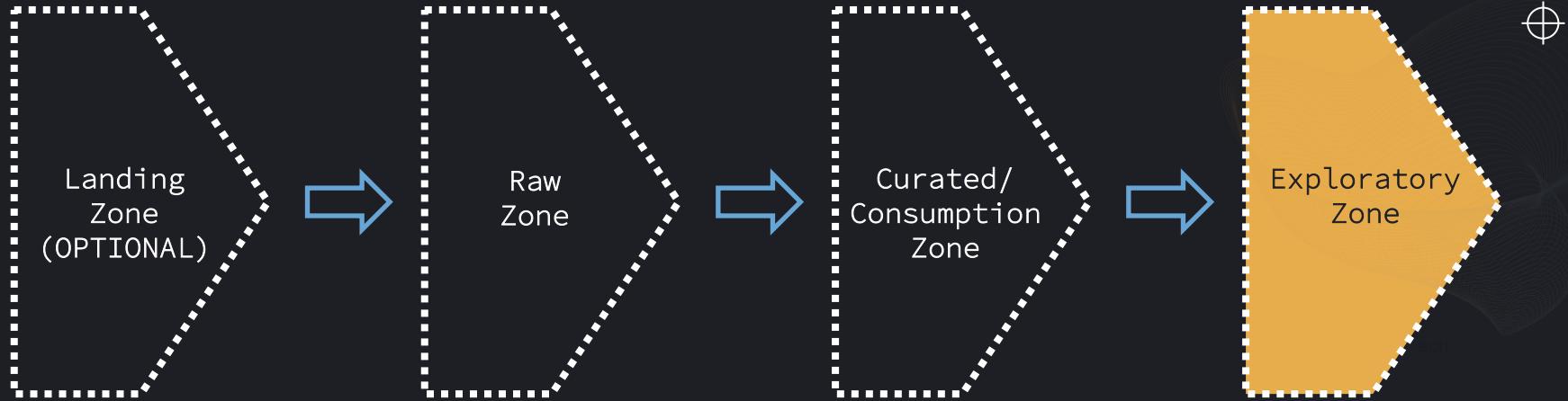




“Main consumption area for analysts, data scientists, and BI tools.”

- Data organized and optimized for query performance
- Access control for different users
- Data can potentially be enriched with additional features





- Non-productive environment for experimentation
- Experimental ML scenarios
- Different policies for storage and data quality
- Separate environment for development purposes





Conclusion

- Zones are not fixed
- Serves as general inspiration
- Single-layer is simpler
- Multi-layer preferable for complex and large-scale solutions



Different Tools & Services



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Storage: AWS S3 Buckets

- Base storage for the data lake.
- Different zones can be implemented using separate S3 buckets.



Data Ingestion



AWS Glue

AWS Glue

- Extracts data from various sources.
- Catalogs and classifies data as it enters the data lake.
- Maintains a centralized metadata repository.
- Can apply transformations to data.

01

02

03

04

05

06



AWS Lambda

AWS Lambda

- Serverless computing for running code without provisioning servers.
- Can be used for specific data ingestion tasks and processing.





Query and Analysis



AWS Athena

Amazon Athena

- Allows querying and analyzing data directly from S3 buckets.
- Ideal for ad hoc querying and exploratory data analysis.
- Provides quick SQL-like queries with good query performance.
- No need to load data into a separate database or data warehouse.

01

02

03

04

05

06

Data Formats in a Data Lake

	Purpose	Characteristics	Use Case
01	CSV (Raw Format)	Initial raw format for data storage.	Simple, human-readable, easy to create. Suitable for raw zone or landing zone.
02	Parquet (Optimized Format)	Optimized for consumption and analysis.	Columnar storage, compression, efficient for analytics. Suitable for curated or consumption layer.
03			
04			
05			
06			

In the next lecture, we'll focus on the structure of data lakes



Data Formats in a Data Lake



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273





01

02

03

04

05

06

Importance

- Underrated aspect in data lake design
- Crucial **for performance, storage cost, and functionality**
- Formats change across data lake zones

01

02

03

04

05

06





Primary Data Format Categories



A large, semi-transparent, light blue wireframe geometric shape, possibly a polyhedron, located in the top-left corner of the slide.

Row
Formats

Columnar
Formats





Primary Data Format Categories

Row
Formats

Columnar
Formats



AVRO



Parquet



ORC





Row Formats

- Store data row by row
- Simple for data generation and ingestion
- Inefficient for analytical purposes





Columnar Formats

- Physically store data in columns
- Efficient access to specific columns
- Enables effective columnar compression
- Allows parallel processing

Performance Advantage of Columnar Formats

- Columnar formats excel in read-intensive scenarios
- Greater storage efficiency



AVRO



Parquet



ORC



Data Lake Zones

Landing Zone/Raw Data Zone

- Store data in raw formats (CSV, JSON).
- Retain data in its original form.





Data Lake Zones

Curated Zone

- Convert data into columnar formats
- Optimized for efficient querying and analytics





Data Lake Zones

Exploratory Zone

- Flexibility to have both row and columnar formats.
- Coexistence to support various exploratory tasks.





Section 3:

Data Ingestion

003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Data Ingestion Methods

003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Overview

- Fundamental step in data lake management
- Process of moving data from various sources
- Crucial for subsequent processing

01

02

03

04

05

06

01

02

03

04

05

06



Streaming Ingestion

- Enables real-time ingestion
- Ideal for time-sensitive data
- Implemented using services like Amazon Kinesis for streaming data.

VS

Batch Ingestion

- Ingests data periodically in batches.
- Cost-effective and efficient
- Tools like AWS glue or azure data factory commonly used





ETL (Extract, Transform, Load) Paradigm

- Traditional approach
- Transformation occurs before loading into the data lake
- Used when data has a rigid structure, Commonly in data warehouses.

ELT (Extract, Load, Transform) Paradigm

- Modern approach
- Transformation occurs on-the-fly when needed
- Flexible approach suitable for data lakes

VS





Blurring Lines between ETL and ELT

- Hybrid approach adapts capabilities traditionally reserved for data warehouses.
- Data lakes allow for flexibility in choosing the most appropriate method based on specific data needs.





Flexibility in Data Lake Approach

- Data lakes provide flexibility
- Consider:
 - Batch ingestion
 - Real-time ingestion
 - ETL, or ELT
- Multiple options depending on different needs



Now we will dive deeper into the basics of batch ingestion



Basics of Batch Ingestion



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Overview

- Common process for ingesting data into a data lake
- Involves loading data in batches over set intervals
- Efficient for large blocks of data

01

02

03

04

05

06

01

02

03

04

05

06



Use Cases of Batch Ingestion

- Ideal for scenarios
- Cost-efficient
- Little time and load on productive source systems





Benefits of Batch Ingestion

- Cost-efficient and time-saving
- Optimization of resource utilization
- Suitable for more complex transformations





Batch Ingestion Process





Batch Ingestion Process



- Identify various source systems
- Determine the data extraction requirements.





Batch Ingestion Process



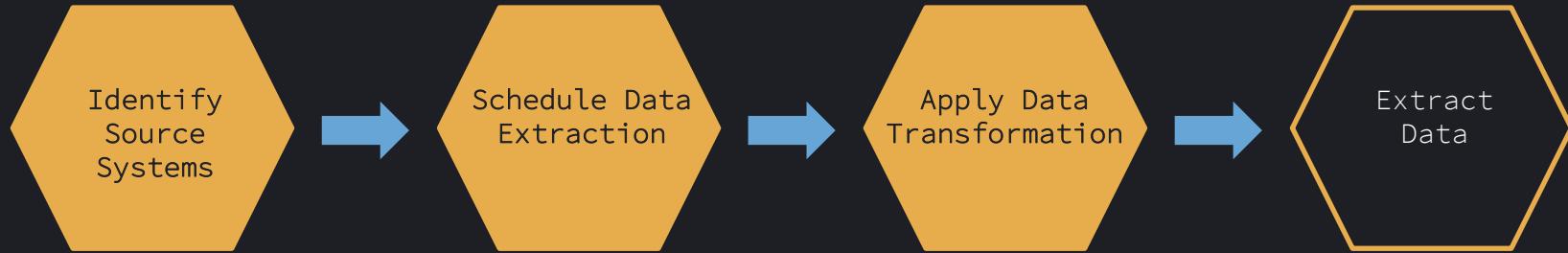
- Identify various source systems
- Determine the data extraction requirements.

- Extract data from source systems at scheduled intervals.
- Choose appropriate times to avoid overloads on productive systems.





Batch Ingestion Process



- Identify various source systems
- Determine the data extraction requirements.

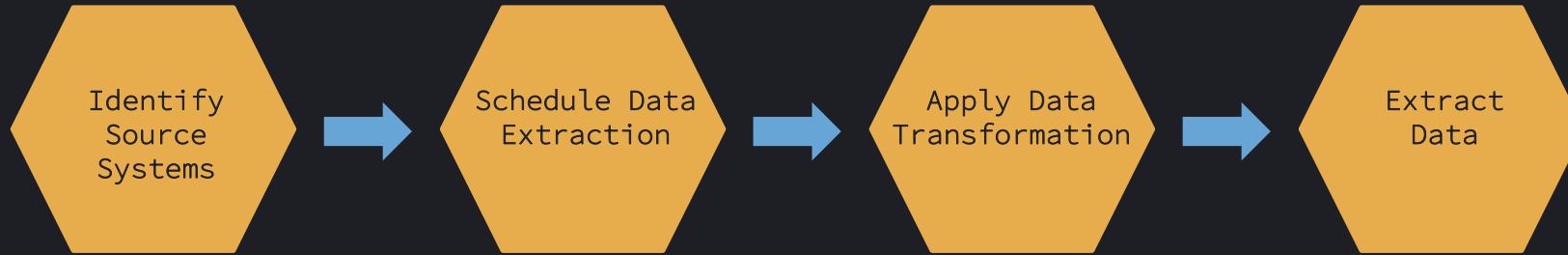
- Extract data from source systems at scheduled intervals.
- Choose appropriate times to avoid overloads on productive systems.

- Apply cleansing, formatting, or aggregation, if needed.
- Transformation steps are typically applied after the initial extraction.





Batch Ingestion Process



- Identify various source systems
- Determine the data extraction requirements.

- Extract data from source systems at scheduled intervals.
- Choose appropriate times to avoid overloads on productive systems.

- Apply cleansing, formatting, or aggregation, if needed.
- Transformation steps are typically applied after the initial extraction.

- Load extracted and transformed data into the data lake.
- Determine whether to store data in the landing zone or raw data zone initially.





Tools for Batch Ingestion

- ETL tools like Pentaho or Talend
- AWS Glue for batch ingestion





Considerations for Batch Ingestion

Appropriate Batch Size
and Frequency

- Align frequency
- Coordinate with administrators

Stakeholder
Communication

- Avoid overloads on source systems
- Schedule batches during non-peak hours

01

02

03

04

05

06





Data Catalog - Profiling



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Importance of Metadata Management

- Key element for efficient and organized data lake environment.
- Adding tags during data ingestion
- Captured metadata includes:
 - Source
 - Format
 - Structure
 - Columns
 - Data types



Data Catalog as a Metadata Repository

- Data catalog serves as a repository for all metadata
- Organizes metadata in a way that facilitates searchability
- AWS services like **Athena** can be used to query data based on metadata.





Metadata Extraction with AWS Glue



- AWS Glue offers automated data catalog capabilities
- Using features like Glue crawlers, metadata can be automatically discovered and cataloged,

1

Automated and Managed Process

Utilizing AWS Glue crawlers for scheduled automated discovery and cataloging.

2

Custom Approach

Establishing custom metadata systems with AWS Lambda functions triggered by specific events to update a custom data catalog.



Data Profiling During Ingestion



Data profiling involves examining data during ingestion for:

- ✓ Content
- ✓ Quality
- ✓ Identifying missing values
- ✓ Errors
- ✓ Inconsistencies

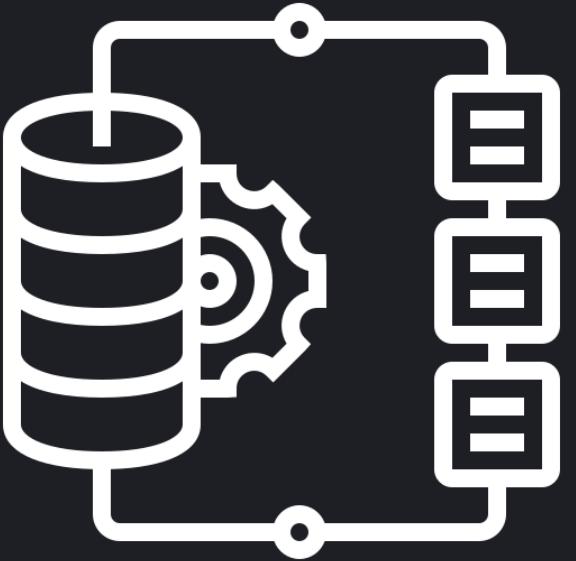
Data profiling can be integrated into the ingestion process to perform basic profiling checks.

Practical Implementation

- Demonstrating the use of AWS Glue crawlers for automated metadata discovery and cataloging.
- Emphasizing the ongoing nature of the process to keep metadata up to date.



Integration



Metadata Management

Data Profiling

Data Catalog



Project Scenario

003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Objective

- Hands-on experience in setting up a basic data lake
- Focus on the data ingestion process

01

02

03

04

05

01

02

03

04

05

06





Project Overview

Company's
Objective

Building a data lake for a
retail company





Project Overview



- Set up the data ingestion pipeline
- Perform basic data processing
- Conduct quality checks
- Implement metadata management





Ingestion Patterns

003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Change Data Capture (CDC)

Use Case

Commonly used in data warehousing

How it Works

- Monitors changes in a data source, such as a database with tables.
- Changes (inserts, updates, deletes) are captured and forwarded to the data lake or target (e.g., data warehouse).
- Database triggers or other mechanisms detect changes and push them to the target, keeping data synchronized in near real-time.





Log Ingestion

Use Case

Extracts data from log files for auditing or monitoring purposes.

How it Works

- Log files generated by applications or devices are analyzed to gather detailed records.
- Useful for tracking and monitoring activities.





Event-Driven Ingestion



Important for real-time insights and data ingestion in data lakes.

Step 1 Identify the Trigger Event

Select an event that will trigger the ingestion process, such as a new file appearing in an S3 bucket.

Step 2 Configure Trigger

Set up a trigger for the identified event. For example, in AWS, use an S3 bucket as the event source.

Step 3 Create Lambda Function

Configure a Lambda function to process the triggered event. This function can be written in languages like Python and defines how the data is processed.

Step 4 Data Processing

Within the Lambda function, define custom processing steps. This may include parsing files, transforming data formats, performing analysis, or conducting data quality checks.

Step 5 Data Movement

Use the Lambda function to move the processed data to the data lake or target storage.





ech

Implementation in AWS

- Use an S3 bucket as an event source.
- Set up a Lambda function to process the triggered event.
- Write a Python script within the Lambda function to define data processing steps.
- Configure the Lambda function to move the processed data to the data lake.

Next lecture will provide a hands-on demonstration of implementing event-driven ingestion in AWS





In-Place Querying

003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



In-Place Querying: Bridging Data Ingestion and Analysis

- Analyzing data involved setting up additional pipelines to transfer data into separate systems like data warehouses or analytic platforms.
01
- **Athena** simplifies the process by enabling in-place querying, circumventing the need for setting up additional systems.
02
03
04





Key Benefits

Efficiency

- Eliminates the need for setting up another system
- Directly analyzes stored data

Flexibility

- Enables quick access to data insights
- Facilitates timely and data-driven decision-making

Cost-Effective

- Eliminates the requirement for additional data processing
- Provides direct access to the data stored

01

02

03

04

05

06





How Athena Works

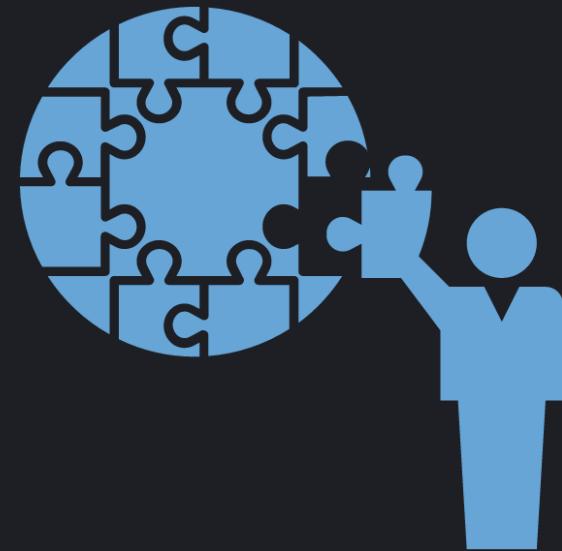
- Serverless query service
- Utilizes standard SQL for querying
- Supports various data formats
- Performance enhancements can be achieved through techniques





Benefits of In-Place Querying with Athena

- Quick access to data insights without the need for additional processing layers.
- Utilize standard SQL queries for data analysis.
- Query data in various formats directly from the data lake.





Understand Data Streaming



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273





Overview

- Involves processing data in real-time
- Ideal for applications

01

02

03

04

05

06



01

02

03

04

05

06



Common Platforms

Apache Kafka

- Open-source streaming platform
- Facilitates publishing and subscribing
- Allows the construction of real-time data pipelines

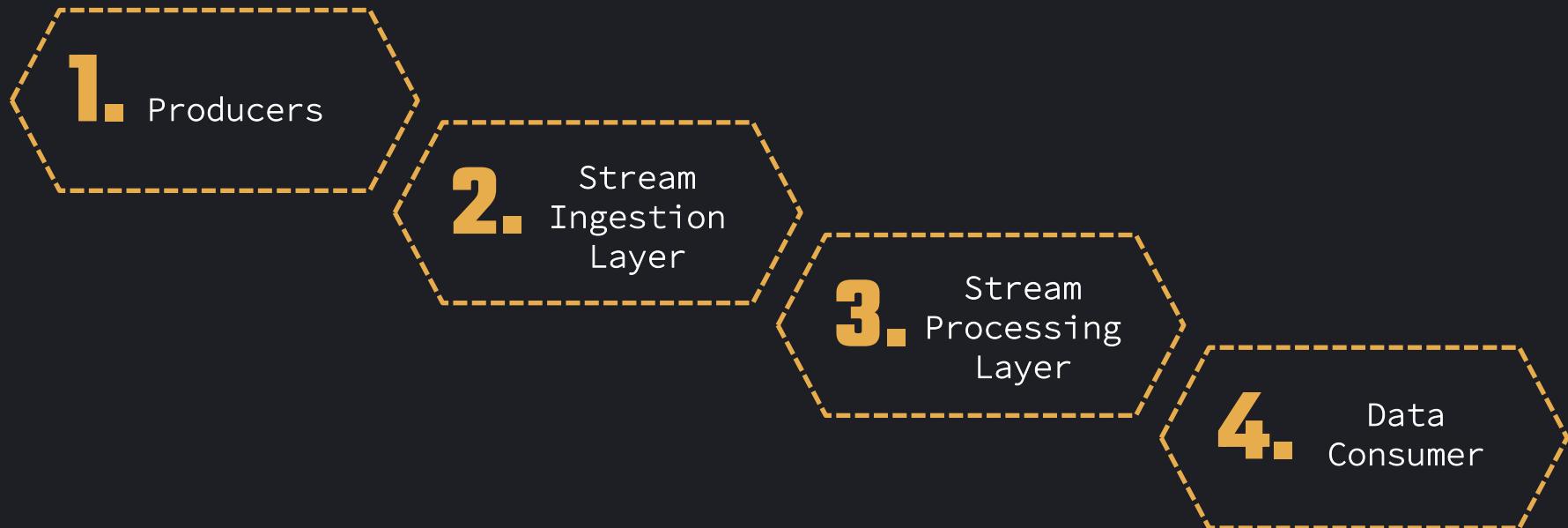
Amazon Kinesis

- Managed service in AWS
- Simplifies the handling of real-time streaming data





Key Components in Data Streaming





Important Terms and Concepts

Shards

- Containers for data in a stream
- Multiple shards handle specific amounts of data simultaneously
- Number and size of shards can be adjusted

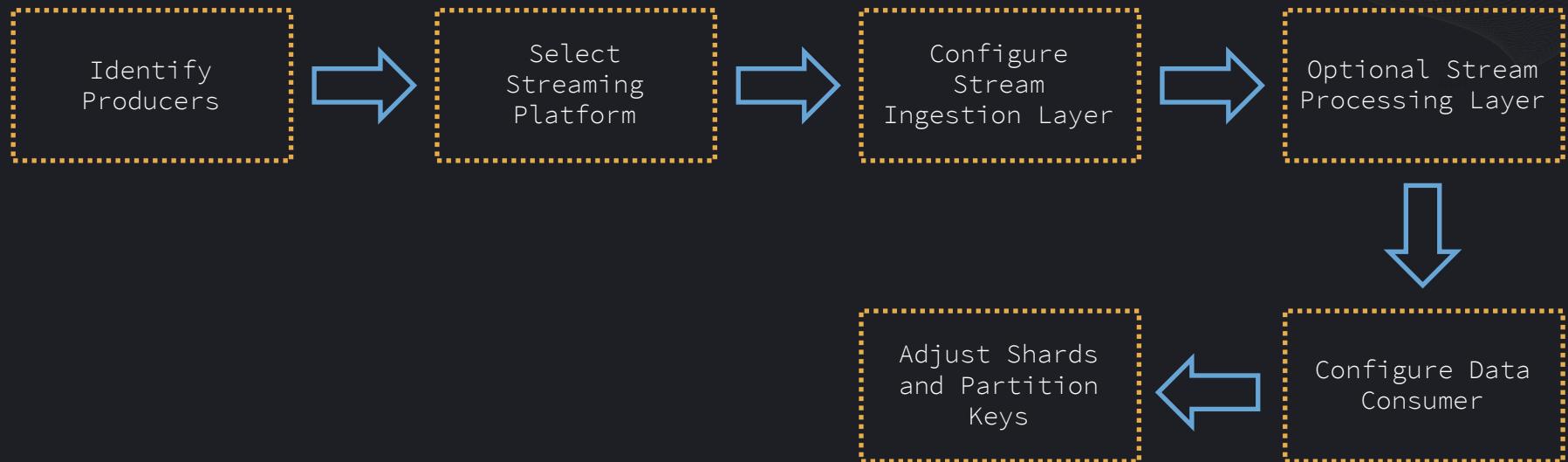
Partition Keys

- Ensure efficient data distribution
- Data labels
- Facilitate sorting





Implementation Steps





Benefits of Data Streaming

- Enables real-time or near real-time processing.
- Streaming platforms facilitate the handling of streaming data.
- Understanding terms is crucial in setting up a data streaming pipeline.



Monitoring and Troubleshooting

003-1040559

1250 003-77156.8

1760 0009-14563.7

73273



Monitoring Data Ingestion

Continuous observation of the performance and health of the data ingestion pipeline



01

Objectives:

- Ensure correct and efficient operation
- Track key metrics and performance indicators

02

03

04

05



01

02

03

04

05

06

07

Key Metrics to Track

Data Throughput

Error Rates

Resource Utilization





Components of Monitoring

1. Track Key Metrics

2. Set Up Alerts

3. Collect Detailed Logs

4. Create Dashboards



Best Practices

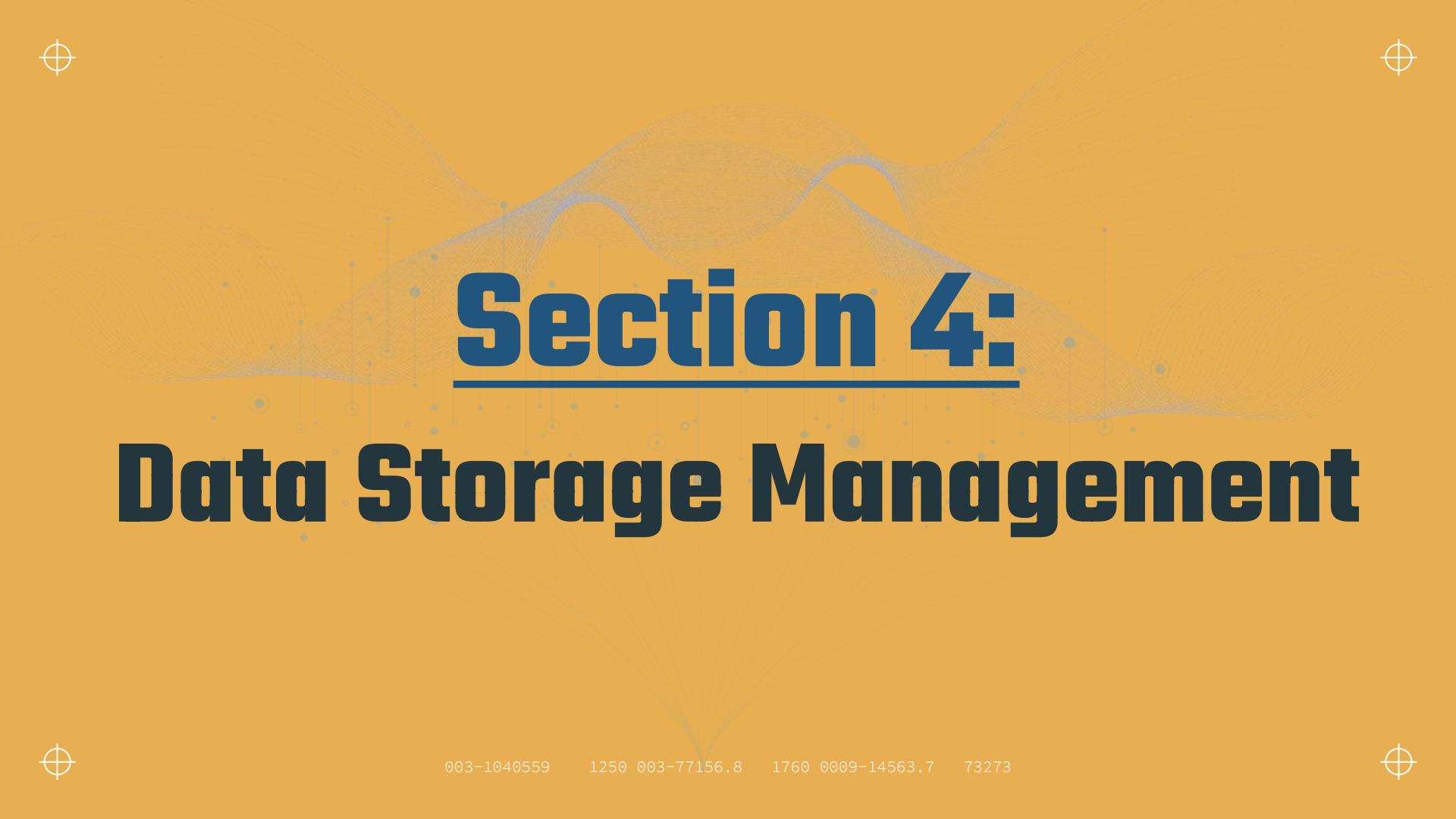




Demonstration in Practice

- Set up monitoring components using AWS CloudWatch.
01
- Configure key metrics, alerts, detailed logs, and dashboards.
02
- Emphasize the importance of proactive monitoring for the health and efficiency of the data ingestion pipeline.
03
- 04
- 05
- 06





Section 4:

Data Storage Management



Key Concepts for Data Storage Management





01

02

03

04

05

06

Key Principles of Data Storage Management

- Use of S3 as a popular choice for a data lake
- S3 buckets form the backbone of our data lake

01

02

03

04

05

06





Important Concepts of S3 Buckets

- Understanding the organization of data into buckets and objects.
- Consideration of effective folder structures.



01

Data Lifecycle Management

- Exploration of how S3 manages the lifecycle of data.
- Automation of archival and retrieval processes.

02

03

04

Data Redundancy and Replication

- Methods employed by S3 for redundant storage.
- Ensuring easy recoverability of data through replication.

05

06





Integration in Data

Data Ingestion

- Transfer of data into S3 buckets.

Data Cataloging

- Organization and cataloging of data within S3 buckets.

Querying Technology

- Understanding techniques and technologies for data extraction and querying from S3 buckets.





Specific Tasks and Purposes

- Understanding the purposes related to S3 buckets.
- Considerations when using other cloud providers like Azure and their equivalent containers.





Setting Up in AWS

- Mastering key principles and concepts in AWS.
- Establishing a well-organized setup in the data lake to leverage the benefits of S3.





Cloud Provider Agnosticism

- Possibility of using Azure containers in the context of a different cloud provider.
- Highlighting the similarities between Azure containers and S3 buckets in terms of functionality.





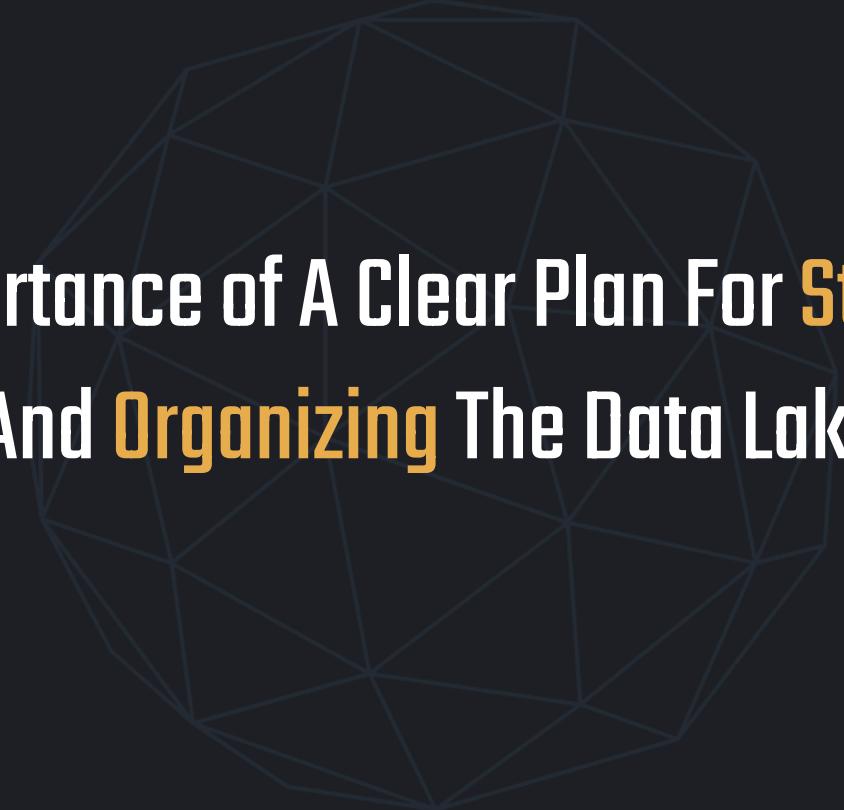
Environment Overview



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



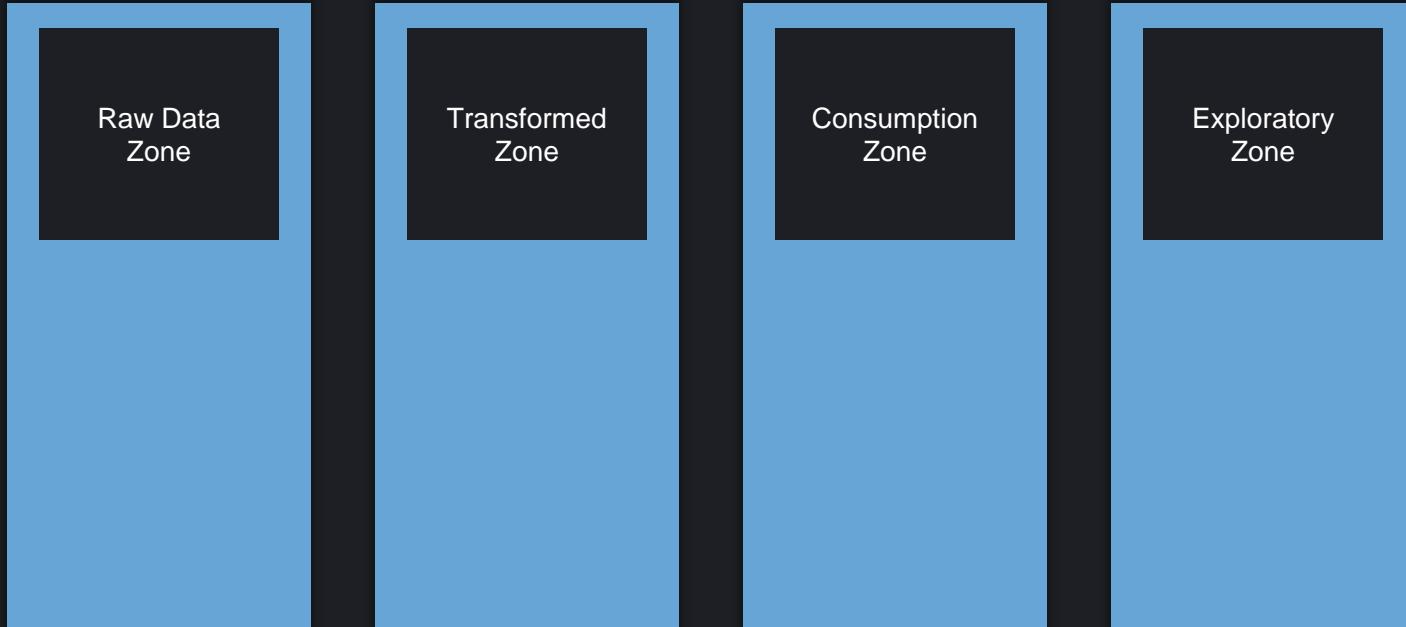
The Importance of A Clear Plan For Structuring And Organizing The Data Lake





Zone Strategy Overview

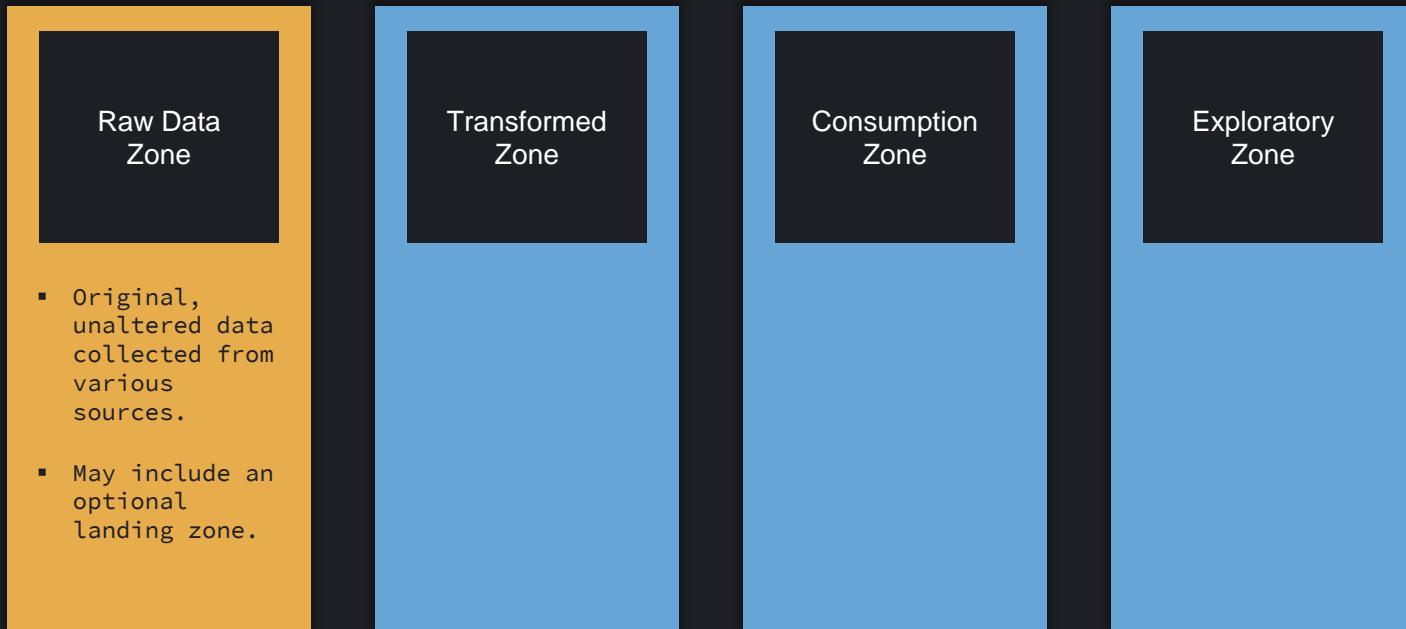
This structure of those different zones is not set in stone - this is just one way of doing it





Zone Strategy Overview

This structure of those different zones is not set in stone - this is just one way of doing it





Zone Strategy Overview

This structure of those different zones is not set in stone - this is just one way of doing it

Raw Data Zone

- Original, unaltered data collected from various sources.
- May include an optional landing zone.

Transformed Zone

- Data modified for specific purposes.
- Possible conversion into formats like Parquet or ORC for efficiency.

Consumption Zone

Exploratory Zone





Zone Strategy Overview

This structure of those different zones is not set in stone - this is just one way of doing it

Raw Data Zone

- Original, unaltered data collected from various sources.
- May include an optional landing zone.

Transformed Zone

- Data modified for specific purposes.
- Possible conversion into formats like Parquet or ORC for efficiency.

Consumption Zone

- Data enrichment, additional transformations, and optimization for analysis.
- Integration with tools like Amazon Athena or Amazon Redshift.

Exploratory Zone





Zone Strategy Overview

This structure of those different zones is not set in stone - this is just one way of doing it

Raw Data Zone

- Original, unaltered data collected from various sources.
- May include an optional landing zone.

Transformed Zone

- Data modified for specific purposes.
- Possible conversion into formats like Parquet or ORC for efficiency.

Consumption Zone

- Data enrichment, additional transformations, and optimization for analysis.
- Integration with tools like Amazon Athena or Amazon Redshift.

Exploratory Zone

- Dedicated to experimentation and development.
- Testing ground for data scientists and the development of new products.





Importance of AWS Account Distinction

Recommendation for a distinct AWS account setup

Keeping production and development environments separated for clarity and efficiency.





Production and Development Separation

- Use of separate buckets and accounts for production and development.
- Ensuring a clear understanding of when actions are in the production environment.

Further partitioning data through a combination of:
👉 Buckets 👉 Folders 👉 Metadata

Next lecture will be on a closer examination of how data is structured and organized in the data lake



Partitioning



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Overview

- Partitioning data through a combination of bucket and folder structure.

01

02

03

04

05

06

01

02

03

04

05

06





Importance of Folder Structure

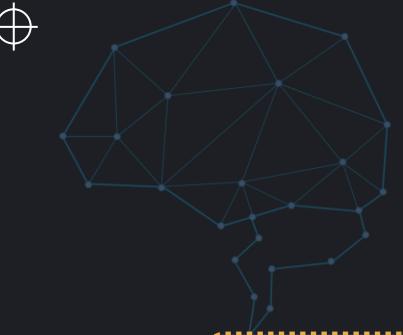
Data Management

- Simplifying tasks such as data retention and archival.
- Facilitating easy archiving or deletion of older data partitions

Query Performance

- Improving query performance by allowing queries to process only relevant data subsets.
- Reducing the amount of data scanned in queries, leading to cost savings.





Implementation of Partitioning

Organizing Data



- Organizing data into folders and subfolders, including the use of buckets.
- Enabling the use of Glue crawlers to automatically create partition keys.

Partitioning Examples

2021 /month=11/day=05/filename.txt

- Time-based partitioning example with S3 keys.
- Introduction of metadata tagging for custom metadata attachment.

01

02

03

04

05

06



Importance of Maintaining The Glue Catalog for Managing Metadata and Defining Partitions





Amazon Athena



Query using Athena.



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Amazon Athena



Query using Athena

The screenshot shows the Amazon Athena console interface. On the left, the 'Data' sidebar displays the 'Data source' as 'AwsDataCatalog' and the 'Database' as 'retail_db'. Below this, the 'Tables and views' section lists several tables: 'nikolai_12345_test' (Partitioned), 'parquet', 'raw_sales', 'sales', and 'sales_sources' (Partitioned). The 'Views (0)' section is also listed. On the right, the main area shows a query editor with four tabs: 'Query 1', 'Query 2', 'Query 3', and 'Query 4'. The 'Query 4' tab is active, containing the SQL command: `1 SELECT * FROM "retail_db"."parquet" limit 10;`. Below the editor are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. The 'Query results' tab is selected, showing a green status bar with 'Completed'. The 'Results (10)' section displays a table with three rows of data:

#	date	product_id	quantity
1	2023-02-02 22:36:09.000	P319	2
2	2023-03-11 10:02:15.000	P896	20
3	2023-12-14 18:06:50.000	P157	14



003-1040559

1250 003-77156.8

1760 0009-14563.7

73273





Efficient Querying

- Focusing on specific criteria, through partitioning for more efficient querying.
- Ensuring data is partitioned according to how it is typically filtered or aggregated.





Folder Structure

003-1040559

1250 003-77156.8

1760 0009-14563.7 73273





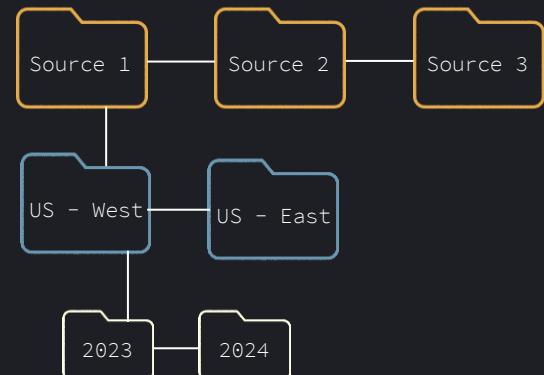
Folder Structure

Bucket Level

Using buckets to define different zones

Raw Zone Folder Structure

- Utilizing source systems as subfolders
- Partitioning by platforms, systems, locations, and date



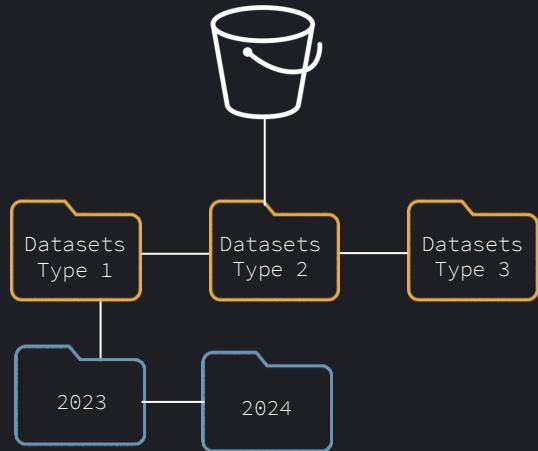


Folder Structure

Transformed Zone Folder Structure

- Optional zone containing cleaned and transformed data
- Subfolders based on specific use cases
- Consideration of how data is commonly used, retrieved, and filtered for efficient partitioning.

Sales Transactions Data (use case 1)



Marketing Data (use case 2)

Each folder holds only datasets of the same Schema structure

Partitioning of the data by date, location, or anything that is used when filtering or grouping the data => better performance

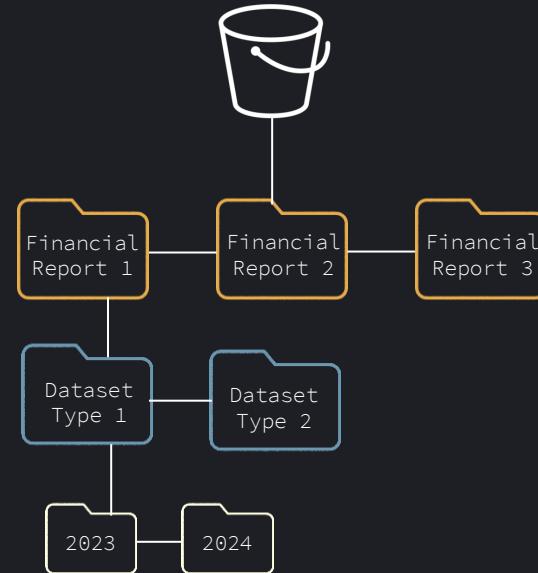




Folder Structure

Curated Zone Folder Structure

- Highly processed and enriched data ready for analytics and business intelligence.
- Organization based on different use cases or analytical purposes (e.g., Curated/Financial Reports).
- Use of Athena for querying data based on specific partitions.

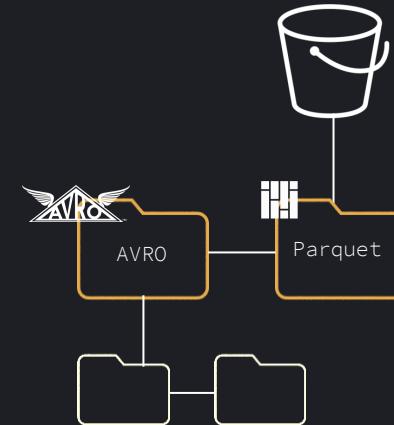




Folder Structure

Data Formats

- Usage of more readable formats like Parquet or ORC in Transformed and Curated zones.
- Ensuring files within the same subfolder have the same structure and format for efficient querying.





Efficient Data Lake Management





Data Lifecycle Management



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273





Importance

- Need for effective lifecycle management to control
 - Storage costs
 - Ensure compliance.

01

02

03

04

05

06



01

02

03

04

05

06



Based on usage patterns and specific use cases,
strike a balance between **Cost-Saving Measures** and
Data Accessibility.



Data Lifecycle Stages





Storage Classes Overview

S3 Standard

- Best choice for frequently accessed data.
- Ideal for regularly queried data

Intelligent Tiering

- For data with uncertain or variable access patterns
- Automatically moves between frequent and infrequent access tiers

Standard Infrequent Access

- For less frequently accessed data, providing quick query capabilities
- High durability due to data replication across multiple availability zones





Storage Classes Overview

One Zone Infrequent Access

- Less expensive but stores data in a single availability zone
- Suitable for non-critical data or with additional backup mechanisms.

S3 Glacier and Glacier Deep Archive

- Low-cost storage solutions for infrequently accessed data.
- Ideal for long-term archiving of historical data.





Cross Region Replication



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273

Overview:



01

02

03

04

05

06

01

02

03

04

05

06

Overview:



01



02

03

04

05

06

01

02

03

04

05

06



Benefits of Cross Region Replication

Disaster Recovery

- Safeguarding critical data in case of outages or disruptions.
- Additional redundancy ensuring data availability.



Latency Reduction

- Reducing data access times for users in different regions.
- Improving user experience by placing data closer to users.





Use Case Demonstrations

Disaster Recovery

- Health care company with critical data in **Sydney** region replicating to **Tokyo** for redundancy.
- Cross region replication **prevents** data loss during regional disruptions.



Latency Reduction

- Gaming company in the United States replicating data to an **S3 bucket in the EU**.
- Reducing data access times for European players, **enhancing user experience**.





Additional Costs and Considerations

Storage Costs

Cross region replication comes with additional storage costs.

Data Transfer Costs

Additional data transfer costs when moving data across regions.





Selective Replication

- Identifying critical data for disaster recovery.
- Considering applications with quick response time requirements.
- Replicating data frequently accessed by users in different regions.

Criteria for Replication





Implementation in AWS

01
Replication Rules and Policies

- Setting up replication rules and policies in the source bucket.

02

03
Automated Replication

- Automatic replication of new objects added to the source bucket based on configured rules.

04

05
Direction of Replication

- Replication occurs only when adding files to the source bucket, not vice versa.

06





Implementation in AWS

01 Deletion Considerations

- Deletion in the source bucket does not delete the object in the destination bucket by default.

02 Versioning Requirement

- Versioning must be enabled in both the source and destination buckets for replication.

03 Practical Implementation

- Implementing cross-region replication in AWS.





Backups & Recovery

003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Data Protection Strategy

- Complementary to versioning and replication, providing a dedicated recovery strategy.

01

02

03

04

05

06

01

02

03

04

05

06





Versioning

- Granular recovery option by tracking all data changes.
- Reverts to specific versions, offering a detailed history.



Replication

- Enhances data availability and accessibility across regions.
- Provides geographical redundancy and disaster recovery options.



Backups

- Created on demand or scheduled for recovery after data loss or corruption events.
- Serves the specific purpose of recovery, integrating with versioning and replication.

Implementation of Backups and Recovery Strategy

Identify Critical Data

- Assess the impact on business functions.
- Consider compliance and regulatory requirements.
- Categorize data based on importance and sensitivity.

1

Implementation of Backups and Recovery Strategy

Determine Backup Frequency

- Understand data change frequency.
- Consider the impact of data loss.
- Adjust backup frequency based on criticality (e.g., every few hours or real-time for sensitive data).
- Define Key Performance Indicators (KPIs).

2

Implementation of Backups and Recovery Strategy

Define Metrics

Recovery Time Objective (RTO) :

- Maximum acceptable time for the recovery process.
- The time it takes until data is restored after a failure.
- Critical data might have a shorter RTO.

Recovery Point Objective (RPO) :

- Maximum tolerable age of data that must be recovered.
- Determines the backup frequency.
- Example: An RPO of 30 minutes requires backups at least every 30 minutes.

3

Implementation of Backups and Recovery Strategy

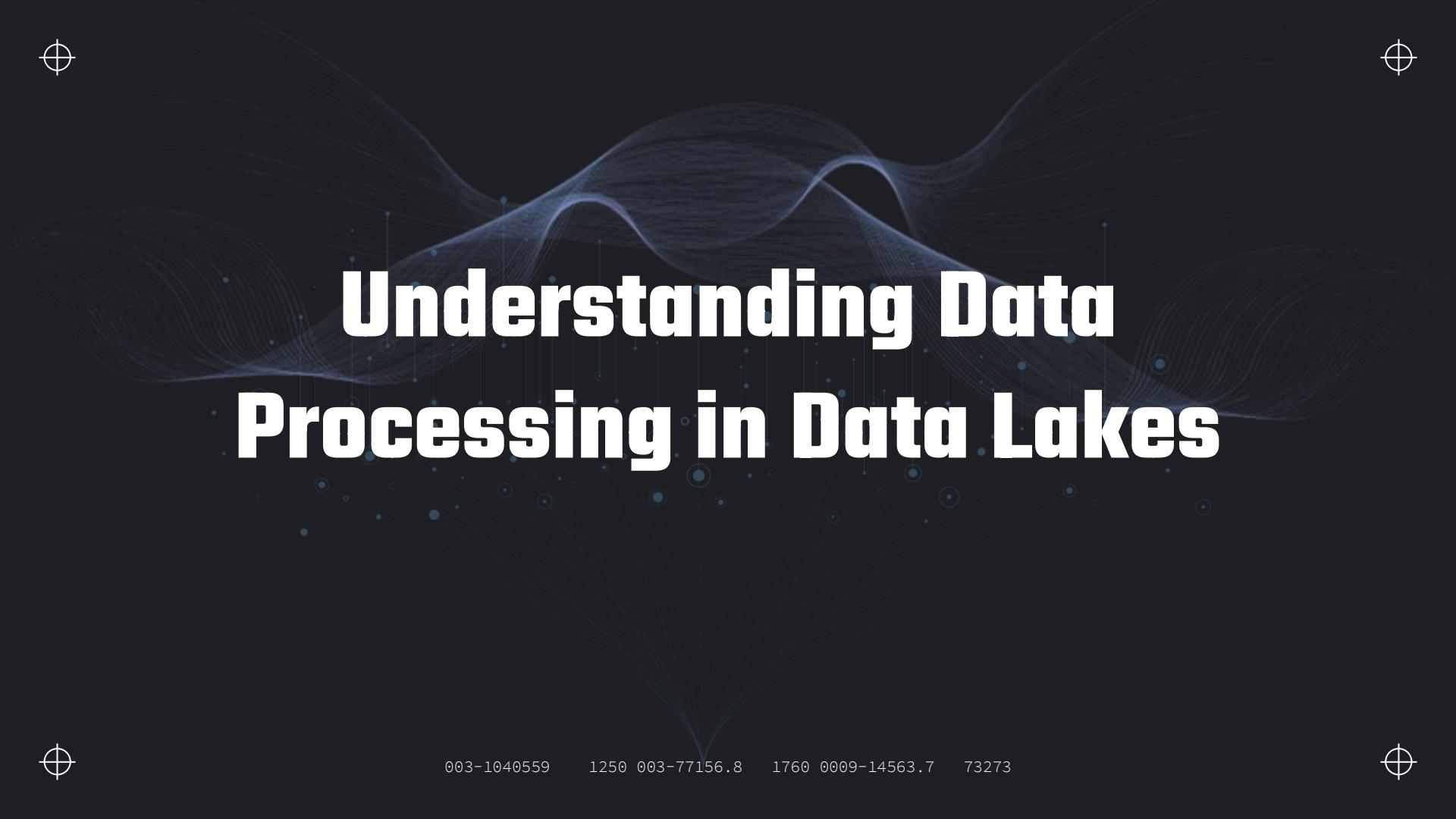
Regular Testing

- Ensure the strategy meets defined objectives.
- Regularly test the recovery process to validate RTO and RPO.

4

Section 5:

Data Processing & Transformation



Understanding Data Processing in Data Lakes

003-1040559

1250 003-77156.8

1760 0009-14563.7

73273



Data Ingestion



003-1040559

1250 003-77156.8

1760 0009-14563.7

73273





Data Ingestion

Initial step to bring data into the data lake from various sources.



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Data Storage



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273





Data Storage



Raw storage of data in its original format.



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273





Data Organization



003-1040559 1250 003-77156.8 1760 0009-14563.7 73273





Data Organization

Cataloging and organizing
data for easier
accessibility using
metadata.



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Data Cleaning and Transformation



003-1040559

1250 003-77156.8

1760 0009-14563.7

73273



Data Cleaning and Transformation

Light transformations,
typically part of ELT
approach, performed before
analysis.





Data Analysis and Transformation



003-1040559

1250 003-77156.8

1760 0009-14563.7

73273





Data Analysis and Transformation

Majority of transformations applied during data analysis, aligning with ELT and schema-on-read approaches.





Tools and Services



003-1040559 1250 003-77156.8 1760 0009-14563.7 73273





Tools and Services

- AWS services such as **Athena** or **Amazon Redshift** for flexible queries and transformations.
- **AWS Glue** for light transformations and handling larger datasets.
- Frameworks like **Apache Hadoop** and **Spark** for efficient processing.





Multi-Zone Data Lake



003-1040559

1250 003-77156.8

1760 0009-14563.7

73273





Multi-Zone Data Lake



- Different zones (Raw, Staging, Curated, and Analytics) serve specific purposes in the data processing lifecycle.
- ELT approach allows flexibility in handling diverse and unstructured data.





Overview of Zones



003-1040559

1250 003-77156.8

1760 0009-14563.7

73273



Overview of Zones

- **Raw Zone:**

Initial landing area for raw and unprocessed data.

- **Staging Zone (Optional):**

Light transformations for initial cleaning and structuring.

- **Curated Zone:**

Substantial transformations and enrichment of data.

- **Analytics Zone:**

Optimized data for specific business intelligence tools and analytics applications.



Tools for Transformation



003-1040559

1250 003-77156.8

1760 0009-14563.7

73273



Tools for Transformation

- **AWS Glue:**
ETL tool, also usable in ELT context for light transformations.
- **Apache Hadoop and Spark:**
Frameworks for executing efficient data processing.





Hadoop



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



01

02

03

04

05

06

Introduction

- Cloud-native services like S3 buckets in AWS are common in modern data lakes.
- Emphasis on scalability, ease of setup, management, and cost-effectiveness.
- Contrast with Hadoop, which can be complex, requires specialized skills, manual maintenance, and may involve hardware costs.





Relevance of Hadoop

- Hadoop remains relevant in on-premise data lakes or hybrid deployments.
- It can still play an important role in legacy Hadoop ecosystems.





Overview of Hadoop

- Hadoop is an open-source framework for storing and processing large volumes of data.
- Providing cost-effective and distributed solutions to big data challenges.





Hadoop Core Capabilities

Hadoop Distributed File System (HDFS).

- Stores large data distributed across commodity hardware.
- Commodity hardware includes standard processors, memories, and hard drives for cost-effectiveness.





Data Ingestion and Storage

- Data ingested in the form of files.
- HDFS splits large files into smaller blocks for distributed storage.
- Replication of data blocks across multiple nodes ensures fault tolerance.





MapReduce Programming Model

- Two main phases: Map phase and Reduce phase.
- Map phase processes data independently in parallel, defining rules in code.
- Shuffle and sort phase organizes results for the reduce phase.





Reduce Phase and Result Storage

- Reduce phase applies another set of rules to further process and transform sorted data.
- Results can be stored back in HDFS or exported to other storage systems.





Benefits of Hadoop

Distributed Processing

- Leverages distributed processing for faster processing of large datasets.
- Multiple machines in a cluster work simultaneously on data.

Fault Tolerance

- Automatic rerouting of work in case of machine failures ensures reliability.
- No interruption in jobs even if a machine fails during processing.

Scalability

- Easily scalable by adding more machines without significant code or architectural changes.

Introduction to Spark as the next topic for discussion in the upcoming lecture.





Spark

003-1040559

1250 003-77156.8

1760 0009-14563.7 73273





01

02

03

04

05

06

Introduction

- Apache Spark and Hadoop are commonly used together in big data processing and analytics workflows.
- They are considered complementary for comprehensive data processing.

01

02

03

04

05

06





Strengths of Apache Spark

- Known for super-fast processing, outperforming Hadoop's MapReduce.
- Utilizes in-memory processing, reducing the need to read from slow disk storage.
- Supports multiple programming languages: Scala, Python, and Java.
- Extends beyond batch processing to handle real-time data streaming, machine learning, etc.





How Apache Spark Works

- Utilizes a cluster of machines (nodes) for distributed data and processing tasks.
- Efficiently processes large datasets by splitting them into smaller, manageable partitions.
- Parallel processing across a cluster significantly speeds up data processing.





Resilient Distributed Datasets (RDDs)

- Core concept in Spark, serving as building blocks for data operations.
- Fault-tolerant, able to recover from errors or node failures by tracking data lineage.
- Created from data stored in various storage systems like HDFS and S3 buckets.



Advantages of RDDs

- Ensures data integrity and reliability through fault tolerance.
- Optimizes data processing speed by storing as much as possible in memory.
- Cluster expansion is easy by adding more machines without significant code changes.



Deployment Options for Apache Spark

Complex Setup

- Acknowledges potential complexity and configuration challenges.
- May create overhead, especially for those new to distributed computing.

Cloud Deployment

- Cloud providers like AWS offer deployment options.
- AWS services include Amazon Elastic MapReduce (EMR), EC2 instances, and AWS Glue.
- AWS Glue, a serverless option, incorporates Apache Spark in its processing engine.





AWS Glue Overview

Serverless Advantage

- AWS Glue is serverless, eliminating the need to manage underlying infrastructure.
- Integrated with Apache Spark in its ETL jobs.

Cloud Deployment Options

- In AWS, services like Amazon EMR, EC2, and AWS Glue provide flexibility.
- AWS Glue simplifies deployment with a serverless, managed approach.

Expect a detailed exploration of AWS glue in the next part of the presentation.





Data Integration with AWS Glue

003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



01

02

03

04

05

06

Background of AWS Glue



- Fully managed data integration service in the AWS ecosystem.
- Serverless nature eliminates the need for users to manage underlying infrastructure.
- Focus on data processing and job creation without worrying about server management.



01

02

03

04

05

06



01

02

03

04

05

06

Importance of Data Integration

- Data integration is crucial, even in data lakes, for smaller transformations and tasks.
- AWS Glue facilitates data integration, providing capabilities for connecting to different data sources and processing data streams.

01

02

03

04

05

06



Components of AWS Glue Ecosystem

Glue Data Catalog

- Central metadata repository storing metadata from data assets.
- Enables easy searchability and queryability of data.
- Automatically captures metadata from data assets using Glue Crawlers.

Glue Crawlers

- Populate the Glue Data Catalog by discovering and classifying data.
- Automatically identify format, schema, and other properties of data sources.
- Schedule or run on-demand to keep data catalog up to date.

Glue ETL Jobs

- Core functionality for preparing and transforming data.
- Three ways to create jobs: Visual ETL, Notebooks and Script Editor.
- Three job types: Python Shell Jobs, Spark Jobs, and Ray
- Streaming Spark for real-time data processing.



AWS Glue Cost Model

- Serverless and pay-as-you-go model.
- Important to manage costs, especially for large-scale processing tasks.





Practicalities of Using AWS Glue in Data Lakes

- Detailed exploration of practical aspects in the context of data lakes.
- Understanding how to create and set up jobs using different approaches.
- Importance of Glue ETL jobs in data integration within the data lake environment.



Expect a deeper dive into the practical usage of AWS glue in the next section.



Cost optimization in Data Lakes



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Optimizing Performance and Reducing Costs in Data Lakes

Tip 1: Optimize Storage Formats

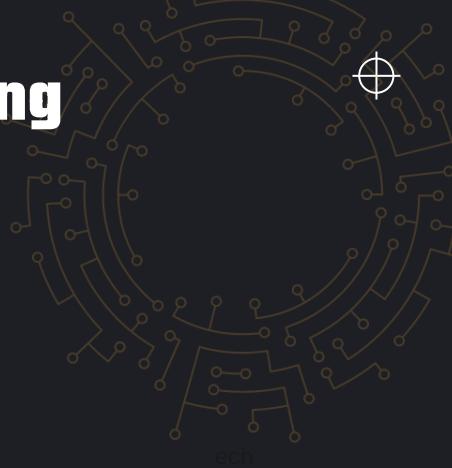
- Convert data to efficient storage formats like Parquet or ORC.
- Optimized for analytics, offer better performance, and reduce storage costs due to advanced compression capabilities.



Optimizing Performance and Reducing Costs in Data Lakes

Tip 2: Leverage Data Partitioning

- Partition data based on queried fields like date or region.
- Reduces the amount of data scanned during queries, lowering costs.
- Example: Partitioning historical sales data by year and month.



Optimizing Performance and Reducing Costs in Data Lakes

Tip 3: Implement Incremental Processing

- Use mechanisms like bookmarking in AWS Glue to process only new or changed data.
- Avoids redundant processing of the entire dataset or files that have already been loaded.



Optimizing Performance and Reducing Costs in Data Lakes

Tip 4: Automate Data Lifecycle Management

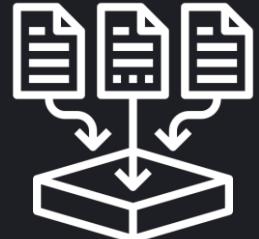
- Implement policies to automatically archive or delete old, infrequently accessed data.
- Utilize AWS S3 lifecycle rules to move data to Glacier or delete after a specified period.



Optimizing Performance and Reducing Costs in Data Lakes

Tip 5: Right Size Computing Resources

- Continuously monitor and adjust compute resources based on ETL job performance.
- Resize the number of DPU and workers in AWS Glue to match job requirements and avoid over-provisioning.



Optimizing Performance and Reducing Costs in Data Lakes

Tip 5: Use Pushdown Predicates

- Apply filters early in the data processing pipeline to reduce unnecessary data processing.
- Use pushdown predicates to filter data before loading, leading to faster execution and lower costs.





Strategic Planning for Cost Optimization

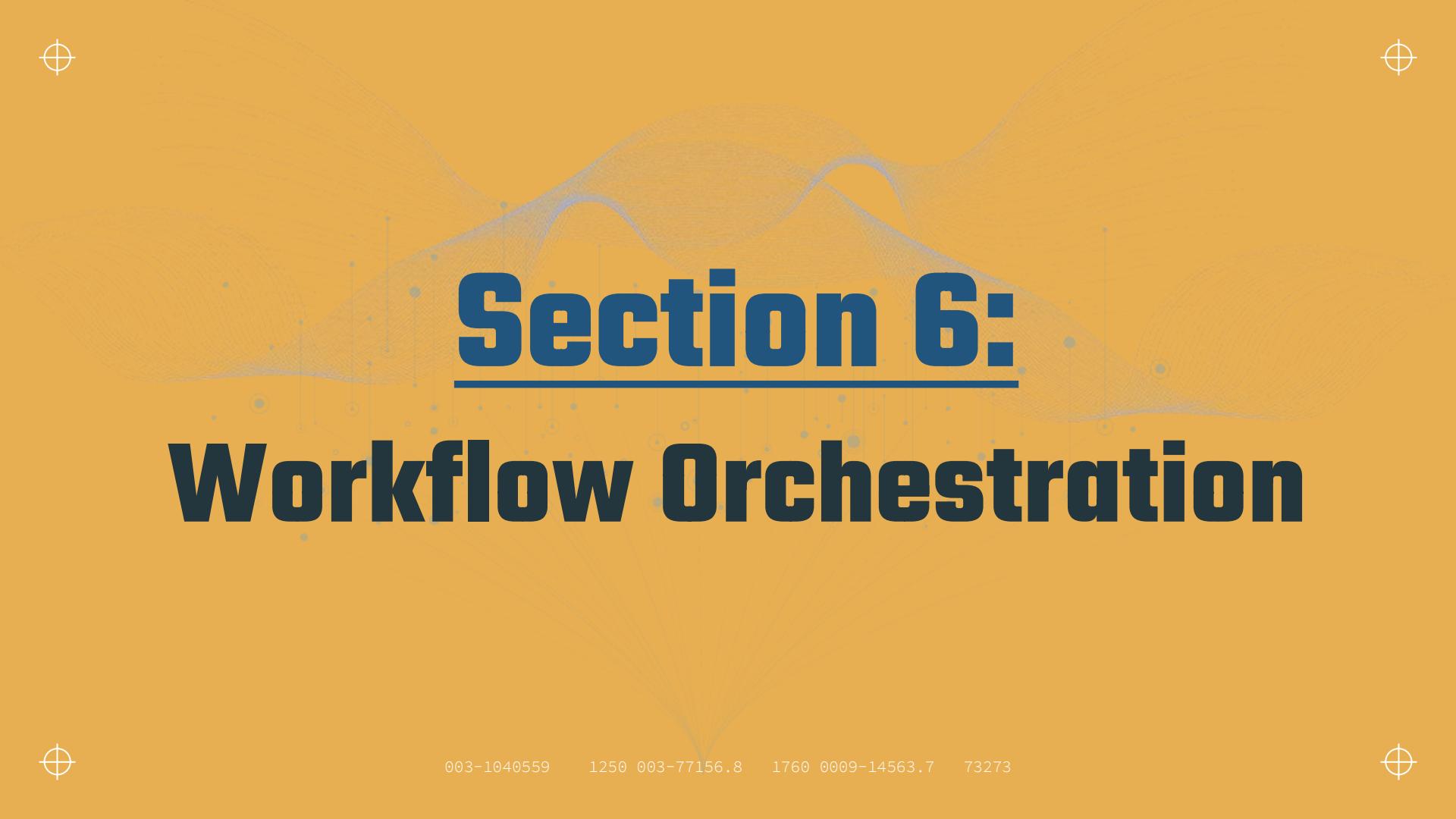
- Emphasizes using resources efficiently and having a strategic plan.
- Involves implementing effective storage formats, partitioning data, leveraging incremental processing, and managing the data lifecycle.
- Regularly monitor usage patterns and make adjustments for ongoing cost optimization.





Conclusion:

- Cost optimization in data lake processing is about strategic planning and efficient resource utilization.
- Following the provided tips can significantly reduce and optimize costs.
- Regular monitoring and adjustments based on usage patterns are crucial for maintaining a cost-optimized data lake environment.



Section 6: **Workflow Orchestration**



Understand Workflow Orchestration



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273





Automation in Data Lakes

- Schedule **ETL jobs** for data processing.
01
- Use **Lambda functions** for event-driven data ingestion.
02
- **Schedule crawlers** for data discovery.
03



01

02

03

04

05

06



05

06



Complex Workflows and Dependencies





Orchestration in Action



Systematic approach to automate and manage processes

Coordination of tasks and services for efficient data flow

Transition from raw data to usable, insightful forms





Components of Workflow Orchestration

AWS Step Functions as a central orchestration tool.

Lambda functions for lightweight processing tasks.

AWS Glue for complex data transformations.

S3 buckets for storage.

CloudWatch for monitoring.

Amazon Event Bridge for triggering events based on conditions or schedules.

Practical Implementation Steps

Designing the Workflow

- Outline the sequence of steps from ingestion to analysis.
- Visualize the workflow on paper.
- Identify triggers and task dependencies.



Setting up Ingestions

- Utilize Kinesis for data streaming or S3 buckets.
- Run ETL jobs for data processing.



Processing Jobs

- Implement AWS Glue jobs for complex transformations.
- Utilize Lambda functions for lightweight processing.

Practical Implementation Steps

Orchestration with Step Functions

- Create state machines for managing task sequences.
- Discuss the practical implementation of state machines.
- Use CloudWatch for workflow monitoring.



Event Triggering with Event Bridge

- Define triggers or schedules using Amazon Event Bridge.



Scenario Overview

- Understand the scenario
- Solve different tasks
- Make use of the big picture

01

02

03

04

05

06



01

02

03

04

05

06





Scenario Automating Retailer Data Lake

003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



01

02

03

04

05

06

Practical Scenario Overview

- Demonstrate the concept
- Acknowledge the advanced nature of the concept
- Improve understanding by simplifying the explanation



01

02

03

04

05

06



Objective

- 👉 Automate the data processing pipeline
- 👉 Focus on daily retail sales





Workflow Overview



STEP 01: Data Ingestion

- CSV files uploaded to an S3 bucket

STEP 03: State Machine and Lambda Function

- Utilize AWS Step Functions state machine
- Include a Lambda function

● STEP 02: Data Validation with EventBridge

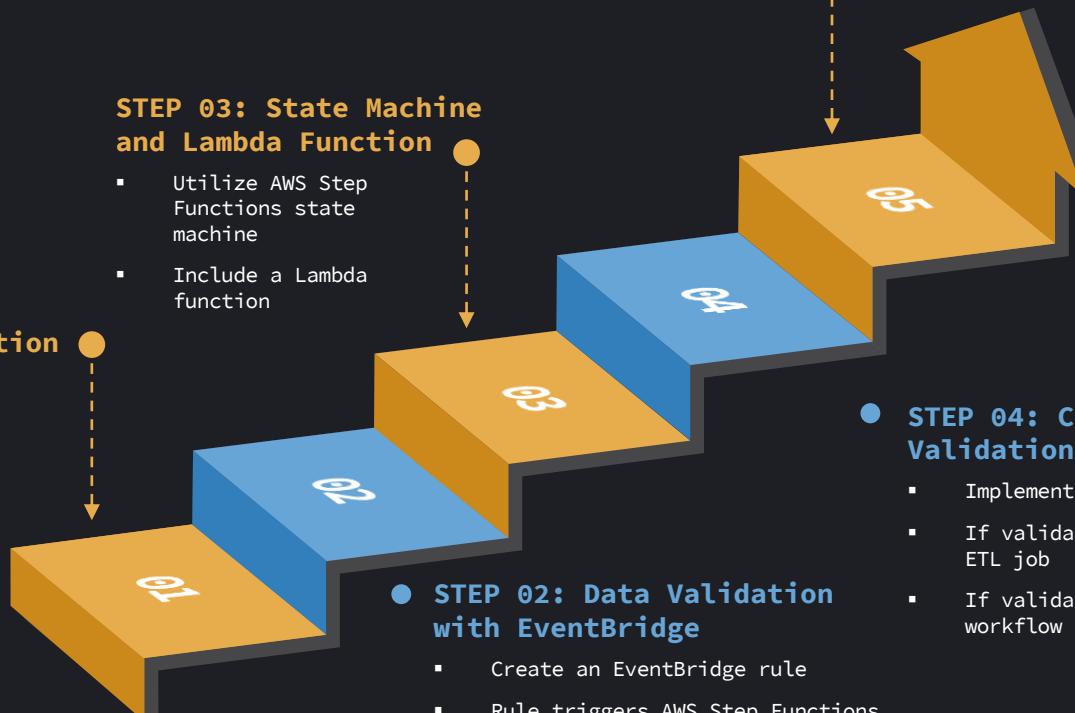
- Create an EventBridge rule
- Rule triggers AWS Step Functions workflow

STEP 05: ETL Job Execution

- Successful validation triggers the execution
- ETL job performs further data transformation

● STEP 04: Choice State for Validation

- Implement a choice state
- If validation passes, proceed to ETL job
- If validation fails, end the workflow





Outcome

Automation of the workflow

Reduction of manual overhead

Quick generation of daily reports.

Ensured data quality

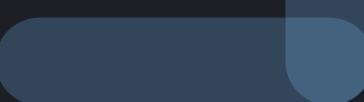




Implementation Steps

Step 1 Designing the Workflow:

- Outline the sequence
- Visualize the workflow



Step 2 Setting up Data Ingestion:

- Configure S3 bucket for sales data uploads.
- Define the structure for CSV files.



Step 3 EventBridge and Rule Creation

- Create an EventBridge rule
- Set up the rule to trigger the AWS Step Functions workflow



Step 4 AWS Step Functions State Machine

- Create a state machine
- Define the logic and flow





Implementation Steps

Step 5 Lambda Function for Data Validation

- Develop a Lambda function
- Check for expected header names and handle success/failure conditions

Step 6 Choice State for Validation

- Implement a choice state in the state machine.
- Define conditions for proceeding to ETL or ending the workflow.

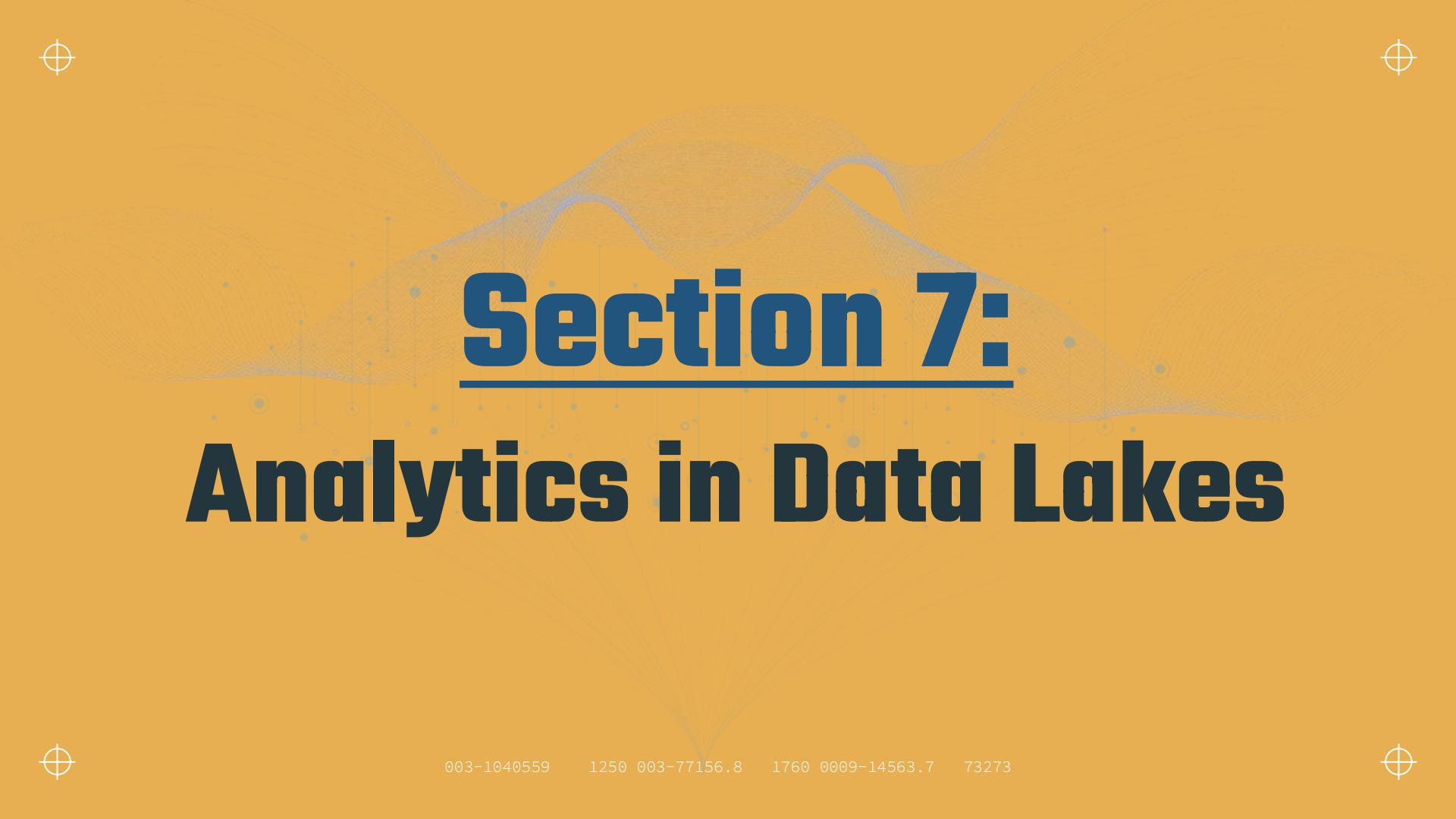
Step 7 ETL Job Configuration

- Set up the ETL job for further data transformation.
- Ensure it is triggered upon successful validation.

Step 8 Monitoring and Optimization

- Utilize CloudWatch for monitoring the workflow.
- Optimize the workflow for efficiency and performance.





Section 7:

Analytics in Data Lakes



Understanding Analytics in a Data Lake

003-1040559 1250 003-77156.8

1760 0009-14563.7 73273



01

02

03

04

05

06

Key Components

- Schema on Read Approach:
 - Structure applied when reading data
 - Allows flexible handling
- Decentralized Analytics:
 - Enables different departments to access and analyze relevant data
 - Eliminates isolated data silos



01

02

03

04

05

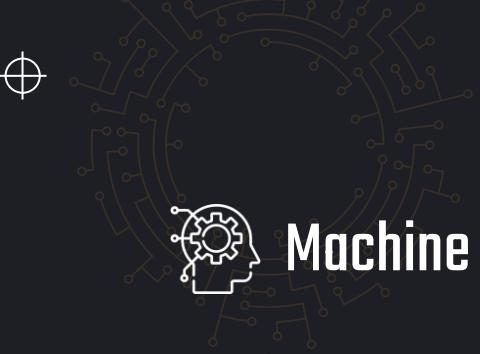
06



How it Works

Data Exploration and Discovery

- Identify potential use cases.
- Conduct exploration in raw, cleanse, or exploratory zones.
- Utilize tools like Athena for ad hoc analysis
- Maintain a data catalog for easy access and search.

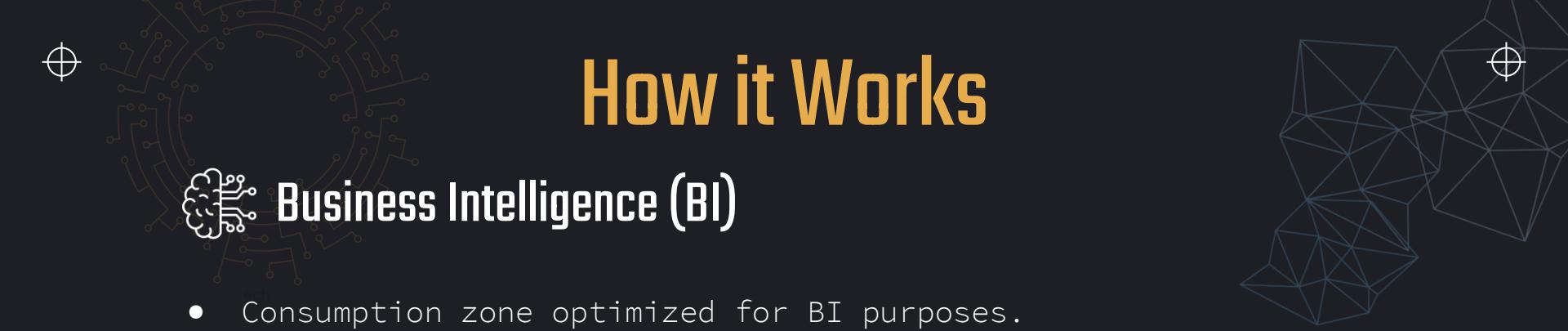


How it Works

Machine Learning

- Foundation for various machine learning models
- Leverage historical data stored in the data lake
- Models often trained with raw or cleansed data
- Experiment and build use cases in the exploratory or sandbox zone
- Use tools like Amazon SageMaker for training and deployment





How it Works

Business Intelligence (BI)

- Consumption zone optimized for BI purposes.
- Business users and analytical tools connect to the consumption zone.
- Perform reporting and interactive visualizations with tools like Power BI or QuickSight.
- Consumption zone feeds into a data warehouse for additional aggregated data.



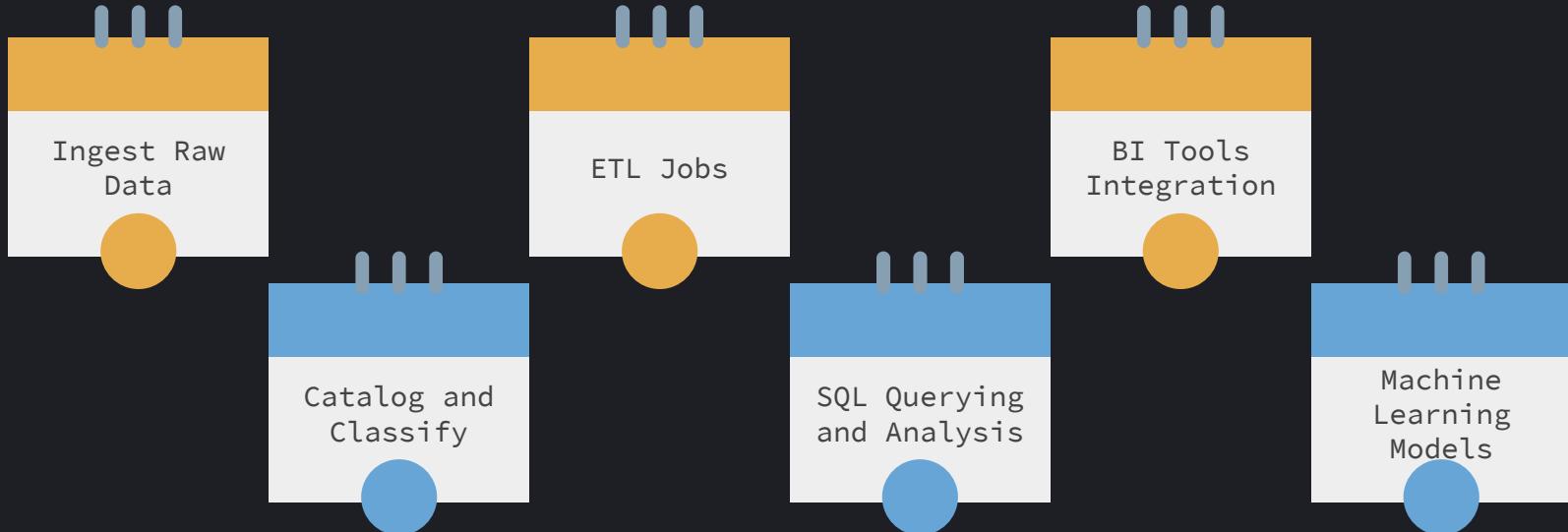
How it Works

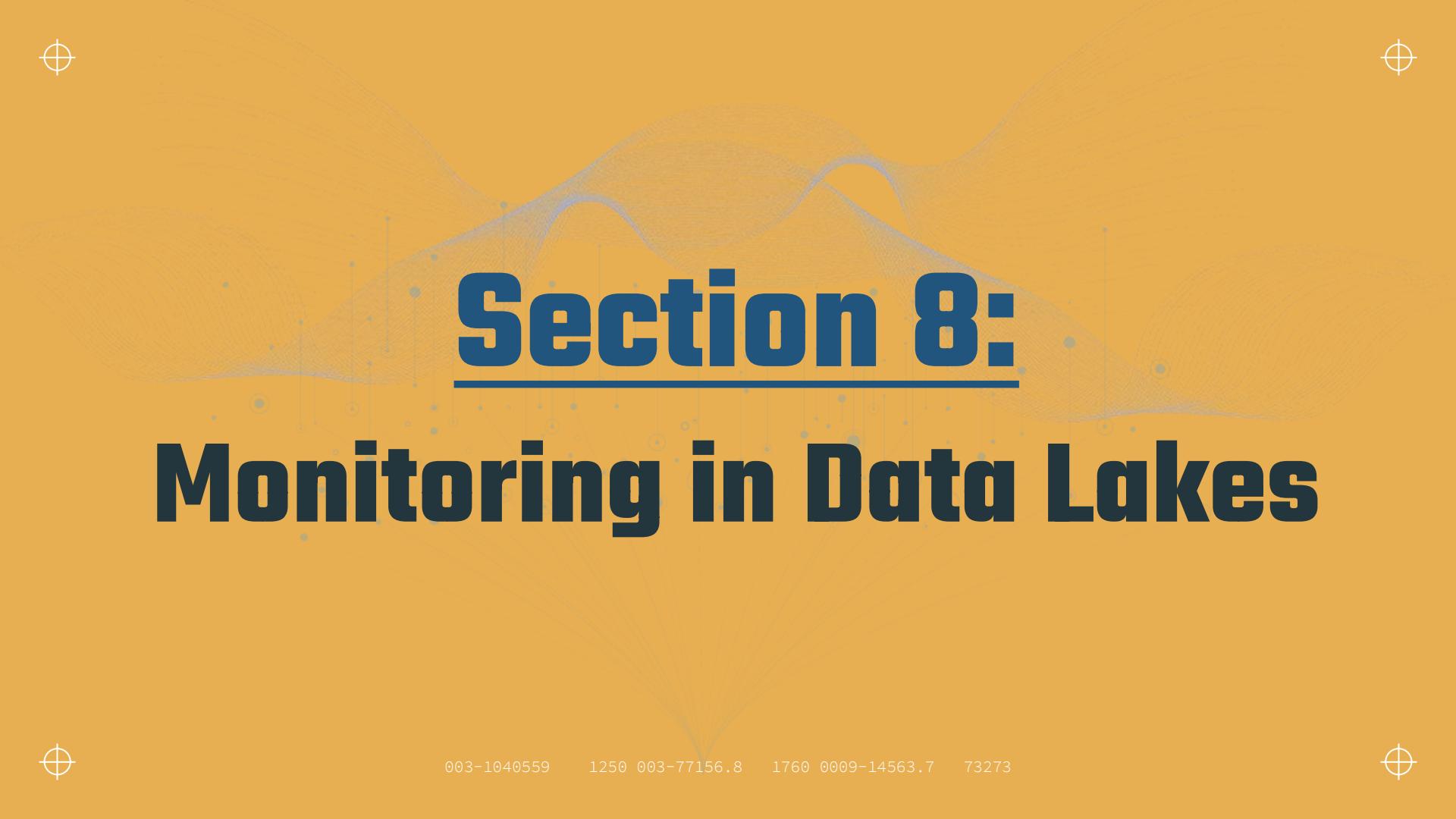
Real-Time Data Streaming and Analytics

- Capture and analyze data as it arrives.
- Useful for time-sensitive decisions.
- Tools like Amazon Kinesis for capturing and analyzing data streams.



Practical Workflow Example





Section 8:

Monitoring in Data Lakes



Need for Monitoring in Data Lakes

003-1040559

1250 003-77156.8

1760 0009-14563.7

73273



Overview

- Data lakes are flexible and scalable, handling large volumes of diverse data.
- Complexity leads to challenges that require robust monitoring.

01

02

03

04

05

06





Challenges Addressed by Monitoring Data Quality

- Diverse data makes ensuring consistency and reliability challenging.
- Monitoring needed to prevent inefficiencies and ensure data quality.
- Implementation of data quality checks.





Challenges Addressed by Monitoring

Performance and Resource Utilization

- Growing data volumes can challenge data retrieval speed.
- Slow query performance and bottlenecks may occur without monitoring.
- Monitoring needed for performance optimization and resource utilization.





Challenges Addressed by Monitoring Security and Compliance

- Ensure compliance with regulations.
- Detect and prevent security risks.
- Monitor unauthorized access and overall security.





Challenges Addressed by Monitoring Cost Control

- Scalability of data lakes may lead to scalable costs.
- Monitoring required to:
 - Control costs
 - Prevent budget overruns
 - And optimize spending





Summary

- Monitoring provides oversight over critical aspects
- Essential for the overall success

Explore the toolset available for effective monitoring in data lakes in the next lecture.



Toolset for Monitoring



003-1040559

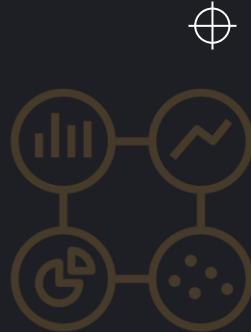
1250 003-77156.8

1760 0009-14563.7 73273



Metrics

- Quantitative data for understanding performance and health.
- Monitored through services like Amazon CloudWatch in AWS.
- Tracks metrics such as read/write requests, storage usage, query execution time, and data scanned.
- Allows setting up alarms for automated notifications based on predefined thresholds.
- Enables immediate reaction to potential issues in the data lake.



Dashboards

- Real-time comprehensive overview of data lake operations and state.
- Implemented using AWS CloudWatch.
- Customizable dashboards with specific metrics for proactive management.
- Provides immediate visibility for better decision-making.





Logs

- Detailed **record-keeping** for auditing, security analysis, and troubleshooting.
- Implemented through services like **AWS CloudTrail**.
- **Offers insights** into specific events, activities, and potential risks.
- Essential for **addressing problems** requiring in-depth details.





Comprehensive Monitoring Approach

Dashboards

High-level overview for quick insights

Alarms

Immediate alerts for proactive management

Metrics

Quantitative analysis for specific aspects

Logs

Detailed information for auditing, security, and troubleshooting





Implementation in AWS

- Utilize AWS CloudWatch for metrics, alarms, and dashboards.
- AWS CloudTrail for detailed logs and event recording.
- Integrate all components for a robust monitoring system.



Section 9:

Access Control in Data Lakes





Access Control in Data Lakes



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273





Authentication

Definition

Proving the user's identity, ensuring they are who they claim to be.

Methods

Credentials (username/password), multi-factor authentication (MFA).

Purpose

Verifying user identity before granting access to the data lake.

Authorization

Definition

Determining what resources a user can access and what actions they can perform.

Example

Specifying read-only or write permissions on specific datasets.

Purpose

Controlling user privileges and actions within the data lake.





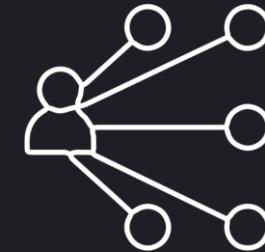
Implementation in AWS

- IAM (Identity and Access Management) users.
- IAM roles and policies.
- Users get the minimum access necessary for their tasks.



IAM (Identity and Access Management) in AWS

- Individual team members or entities with credentials.
- Collections of IAM users with shared policies.
- Assign policies defining allowed actions on resources.



Role-Based Access Management (RBAC)

Definition

Assigning roles to users based on their responsibilities.

Implementation

IAM roles and policies are assigned to users or groups.

Advantage

Simplifies access management by categorizing users based on roles.



Principle of Least Privilege

Definition

Users get the minimum access necessary for their tasks, reducing security risks.

Implementation

Assign only essential permissions required for specific roles.

Advantage

Limits potential damage from accidental or intentional misuse.





Principle of Least Privilege (PoLP)



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



**Grant users/systems the minimum access required
for their job function or specific tasks.**

**Reducing the risk of unauthorized access,
accidents, and data breaches.**



Benefits



Minimize
Attack
Surface



Limit
Impact of
Accidents



Data
Protection



Compliance



Practical Implementation



Data Analyst Example

Grant access only to a specific dataset relevant to the analyst's task.



Different Privileges

Assign read-only access to data analysts, while data engineers may need modification rights.





Challenges

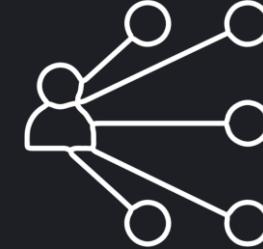
Identifying Minimum Necessary Access

Determining the precise access required for each role or task.

Balancing Security and Usability

Finding the right equilibrium between restricting access for security and enabling productivity.





Role-Based Access Control (RBAC)

Concept

Assigning roles to users based on their responsibilities.

Implementation

Defining roles with specific access rights; users assume roles based on their tasks.

Benefits

Efficiently manages access control, especially in complex environments with varying roles.



Role-Based Access Control **(RBAC)**



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Practical Steps





AWS Components in RBAC

- Entities in the AWS account, representing individuals or services interacting with AWS resources.
- Used to assign permissions to users or groups.
- Collections of users with distinct permissions, serving as a proxy for roles in RBAC.



Policies in AWS

- Pre-built policies that can be assigned and reused for multiple users or groups.
- Embedded directly into a user or group, tied specifically to that user or group.



IAM Groups in RBAC

Purpose

Serve as a proxy for roles in the concept of RBAC within AWS.

How it Works

Create groups like data analyst, data engineer, admins, assign policies, and add users to manage access.



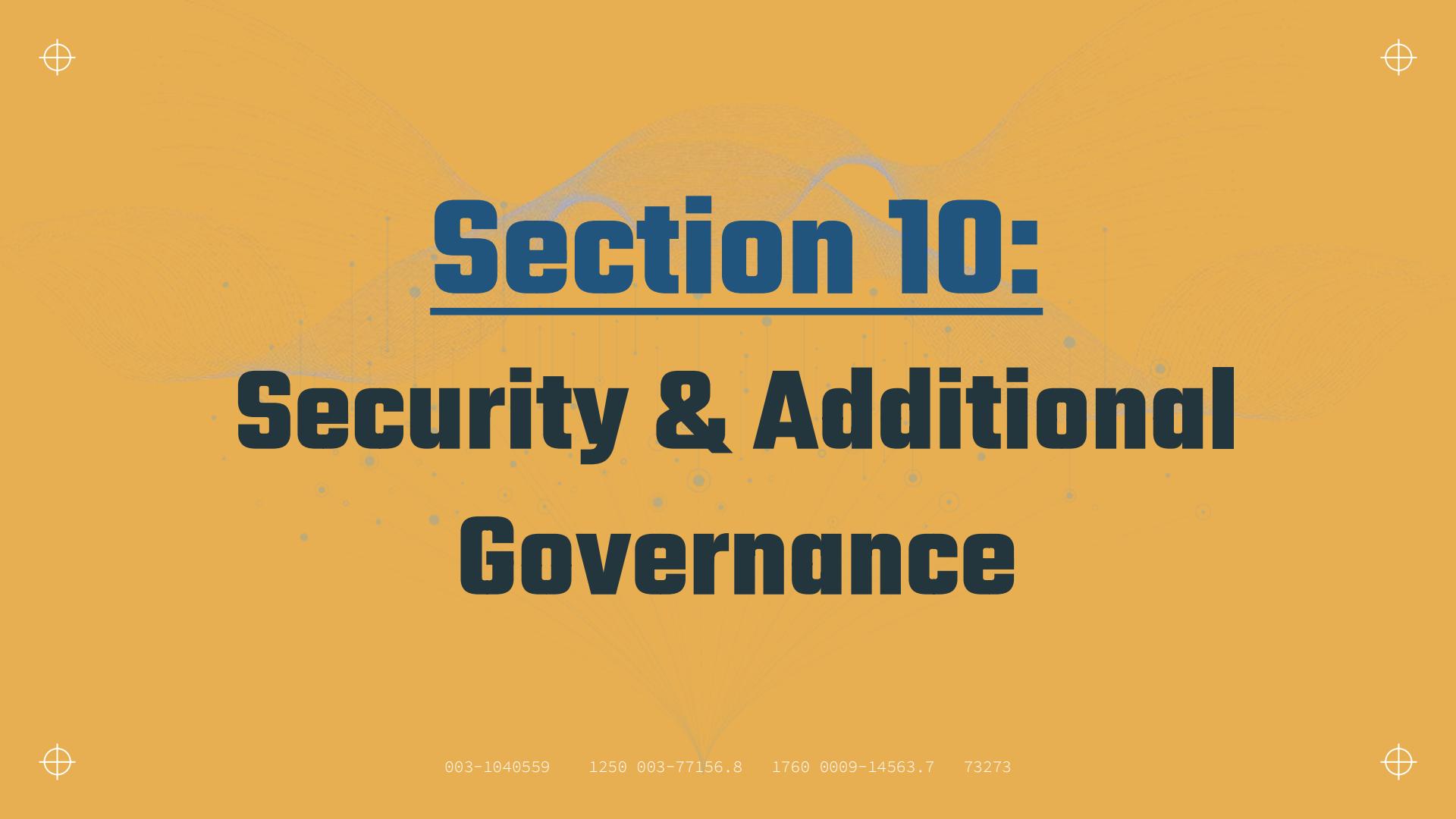
IAM Roles in AWS

Purpose

Often used for delegating permissions to services or for cross-account access.

How it Works

Useful for scenarios involving temporary access for external developers or permissions for AWS Lambda functions.



Section 10:

Security & Additional

Governance





Multi-Layered Security Strategy



003-1040559

1250 003-77156.8

1760 0009-14563.7

73273





01

02

03

04

05

06

Importance

- Protects sensitive data and resources
- Prevents unauthorized access and potential threats
- Implements a multi-layered structure

01

02

03

04

05

06





Defense in Depth Strategy

- Refers to a multi-layered approach in network security.
- Adds unique dimensions or layers, making it more difficult for attackers to breach systems.



Layers of Network Security



Perimeter Security

- Utilizes firewalls to filter incoming and outgoing traffic.
- Implements security groups and network ACLs for fine-grained access control.



Network Segmentation

- Uses Virtual Private Clouds (VPCs) for private, isolated sections in the public cloud.
- Divides VPCs into multiple subnets for further segmentation.



Access Control Layer

- Utilizes AWS Identity and Access Management (IAM) to manage users and applications.
- Adheres to the principle of least privilege, granting minimal necessary permissions.



Monitoring and Detection Layer

- Leverages Amazon CloudWatch for resource monitoring and setting up alarms.
- Employs AWS GuardDuty to automatically detect suspicious activity and unauthorized access.



Data Protection Layer

- Implements encryption for data in transit and data at rest.
- Ensures robust security by default in AWS buckets.



Comprehensive Security Measures

- Combination of measures required for a comprehensive security approach.
- Security strategies should be tailored to the specific needs of the organization. There is no one-size-fits-all solution.



Encryption



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Importance

- Essential for protecting sensitive data
- Necessary for regulatory standards
- Two main types: **Encryption At Rest** and **Encryption In Transit**



01

02

03

04

05

06

01

02

03

04

05

06

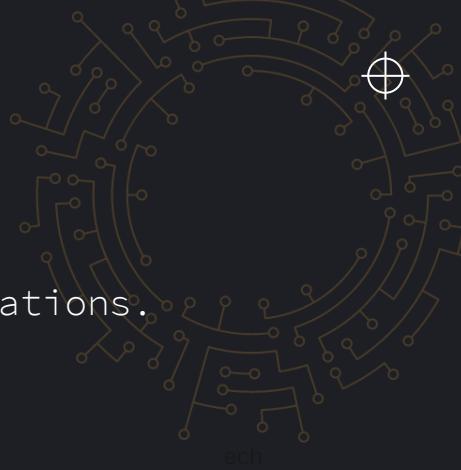
Encryption at Rest

- Data stored in services like **S3 buckets** is encrypted by default in AWS.
- Server-side encryption with **AWS managed keys**.
- Key management method can be chosen based on needs.
- **AWS KMS** (Key Management Service) allows users to manage their own keys.
- Dual layer server-side encryption for an extra layer of security.
- Key size of **AES 256** provides a strong encryption standard.



Encryption in Transit

- Required when **transferring data** between networks or applications.
- Typically uses **SSL** or **TLS** encryption, supported by **HTTPS**.
- Enabled by default for most managed services like **S3 buckets**.





Practical Scenario - Data Classification and Sensitivity Categories



01

- Identify different data types and classify them based on sensitivity.



02

- Example categories:** Highly confidential, confidential, internal use, public data.

03

- Classification ensures appropriate encryption measures for each data type.

04

05

06



Implementation in AWS

- AWS provides default encryption
- Users can choose key management methods
- AWS enables encryption in transit by default
- Data classification and sensitivity categories guide encryption implementation



Conclusion

- Encryption is a critical layer in safeguarding data lakes.
- AWS offers robust encryption options, both at rest and in transit.
- Data classification ensures tailored security measures based on sensitivity levels.



Using Tags



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273

What are Tags

Tags are user-defined labels in the form of key-value pairs.

Example:

Key "department" with value "finance department."





01

02

03

04

05

06

Usage of Tags

- Tags are applied to **resources and data objects** in the data lake.
- Can be added to ETL jobs, files in S3 buckets, etc.
- Up to 10 tags can be added to a file, and they can be modified or removed.
- Automation with AWS Lambda functions is possible for tag management.



1

2

3

4

5

6



Tag Use Cases

Classification of Data

- Classify data by source, purpose, data owners, department, or security tags.
- Enhances data understanding, especially in larger data lakes.

Access Control

- Tags can be used to define access controls in AWS policies.
- Restrict or allow specific user actions based on tags, e.g., restricting access to confidential data.

Data Lifecycle Management

- Trigger actions like archiving or deletion based on tags.
- Automatically move data to different storage classes after a specified period.

Cost Allocation and Management

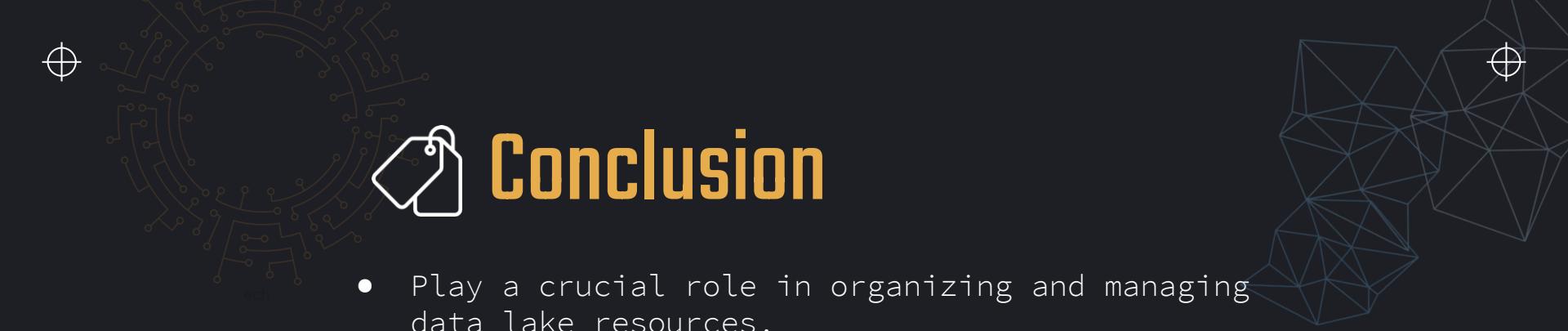
- Associate costs with projects, departments, or business units using tags.
- Useful for budgeting and understanding cost allocation.





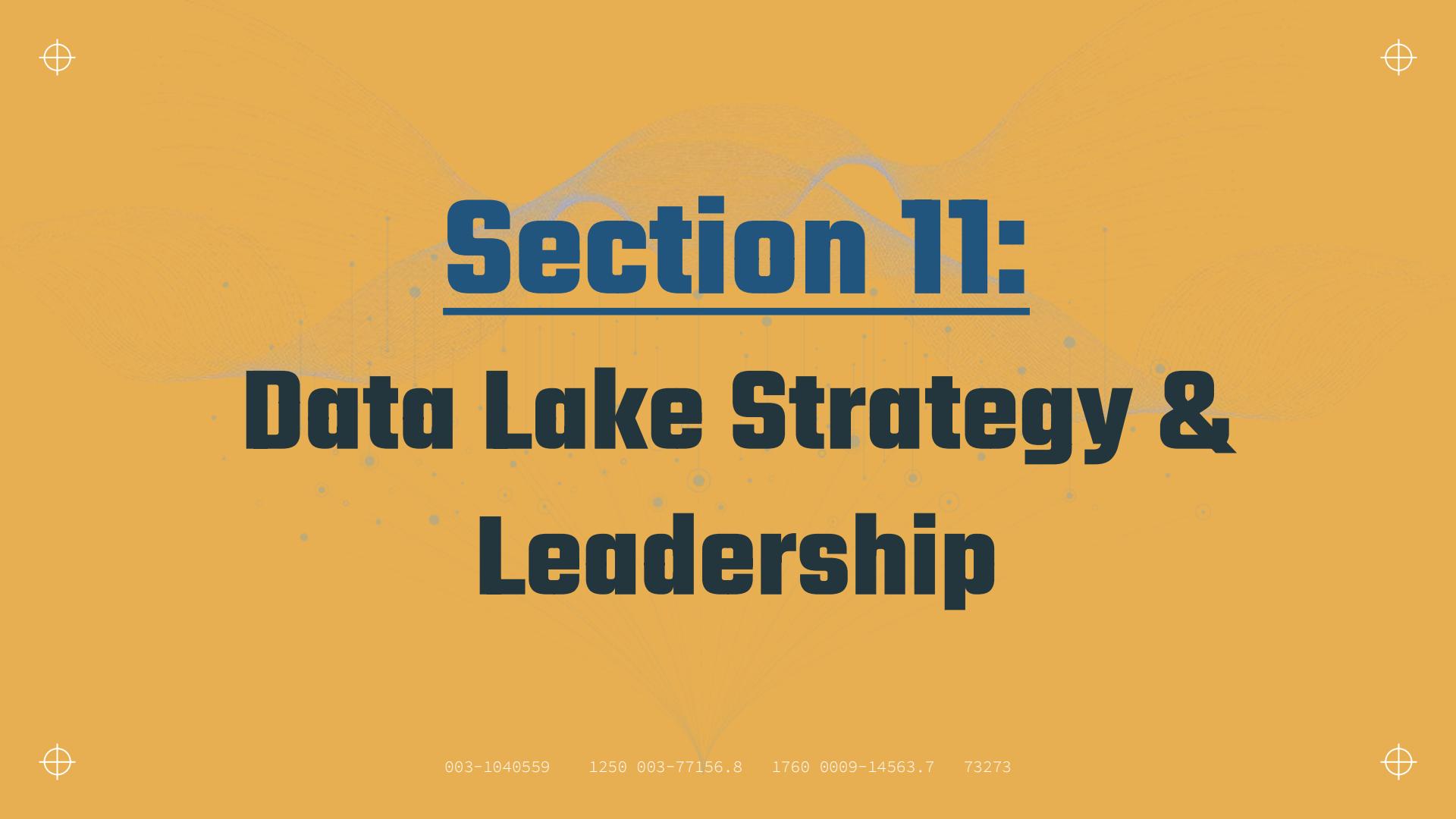
Practical Demonstration

HOW TO add, modify, and remove tags



Conclusion

- Play a crucial role in organizing and managing data lake resources.
- Enhance:
 - Data understanding
 - Assist in access control
 - Automate data lifecycle management
 - Contribute to cost allocation



Section 11:

Data Lake Strategy &

Leadership





Data Lake

Strategy & Leadership



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Importance

- Usefulness for Aspiring Managers
- Benefits for Aspiring Team Leads
- Value for Technical Roles
- Advice to Decision Makers



01

02

03

04

05

06

01

02

03

04

05

06



Assessment of Needs in Data Lake

003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



01

02

03

04

05

06

Understanding Business Objectives

- Define the overarching goal and vision of the data lake.
- Identify specific problems the data lake aims to address.
- Align data lake objectives with business organization goals.



01

02

03

04

05

06



Implementing Needs Assessment

- Conduct a thorough assessment of business needs and pain points.
- Use interviews with department heads and relevant users to understand requirements.
- Analyze current data usage patterns to identify issues and bottlenecks.





Example Scenario





Example Scenario





Example Scenario



Issues Addressed



Defining Clear Objectives

Example:

Consolidate customer, sales, and inventory data from all systems into the data lake within six months.



The next lecture's focus on stakeholder identification and involvement.





Identifying and Involving Key Stakeholders





01

02

03

04

05

06

Importance

- Critical for data lake implementation success.
- Identifying stakeholders crucial for understanding needs and aligning with business goals.
- Neglecting stakeholders can lead to resistance and poor adaptation.



01

02

03

04

05

06



Stakeholder Identification

- Begin with creating a comprehensive stakeholder list.
- Include professionals from various departments, such as marketing, sales, and IT.
- Mention specific names, roles, and influence on the project.



Meeting Preparation

- Schedule individual meetings with department heads and influential team members.
- Prepare tailored questions to understand department data usage patterns.
- Keep communication simple and clear, avoiding technical jargon.



Communication Plan

- Implement a simple and consistent communication plan.
- Example: Monthly email updates with project milestones, updates, and upcoming decisions.
- Encourage stakeholder input on important decisions.



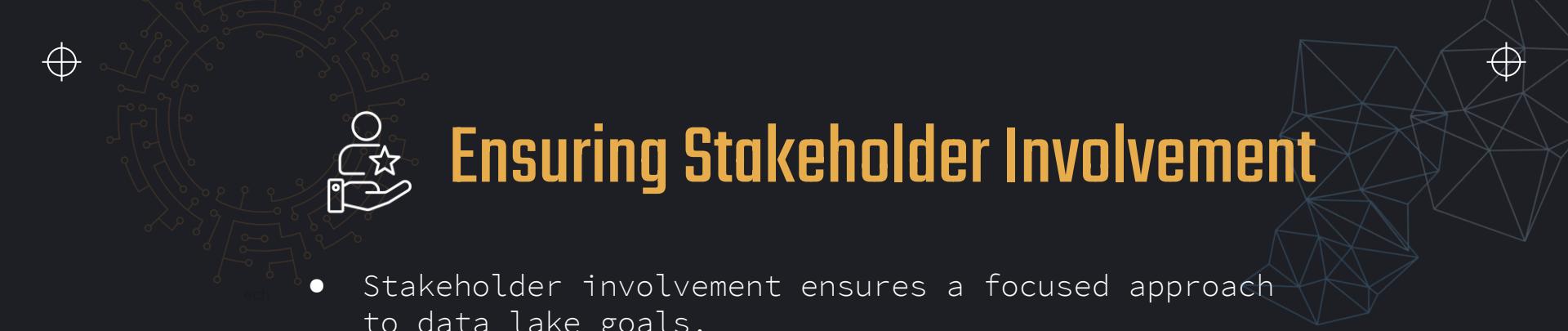
Assessment of Data Landscape

- Connect with stakeholders to assess the current data landscape accurately.
- Create an inventory of data sources, including CRM systems and external databases.
- Document data details:
 - Formats
 - Storage
 - Usage
 - Structured or unstructured.



Identifying Data Quality Issues

- Collaborate with stakeholders to identify data quality issues.
- Use tools to detect missing values, duplicates, or inconsistencies.
- Document data quality challenges to understand the scope of improvement needed.



Ensuring Stakeholder Involvement

- Stakeholder involvement ensures a focused approach to data lake goals.
- Understand actual pain points and issues for smoother implementation.
- Establishes a connection between technical implementation and department-specific needs.

The next lecture will delve into implementing a data governance framework.





Data Governance Framework



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Data Governance Team Roles

Manager/Data Governance Lead

- Main coordinator for the data governance framework.
- Oversees daily operations, coordinates between roles, and reports progress.

Data Stewards

- Assigned from various departments
- Understand data and business context.
- Acts as a link between data governance team, data engineers, and end-users.

Data Architects/Data Engineers

- Technical experts ensuring the data lake supports defined policies.
- Bridge the gap between technical aspects and defined standards.





Data Governance Team Roles

Legal/Compliance Officer

- Specializes in legal and compliance aspects of data.
- Ensures adherence to relevant laws and regulations, particularly when dealing with sensitive information.

Information Security Officer

- Oversees security aspects of the data lake.
- Ensures alignment of data governance policies with security standards.





Role Dynamics

Roles can be combined or taken individually, depending
on the specific needs of the data lake.

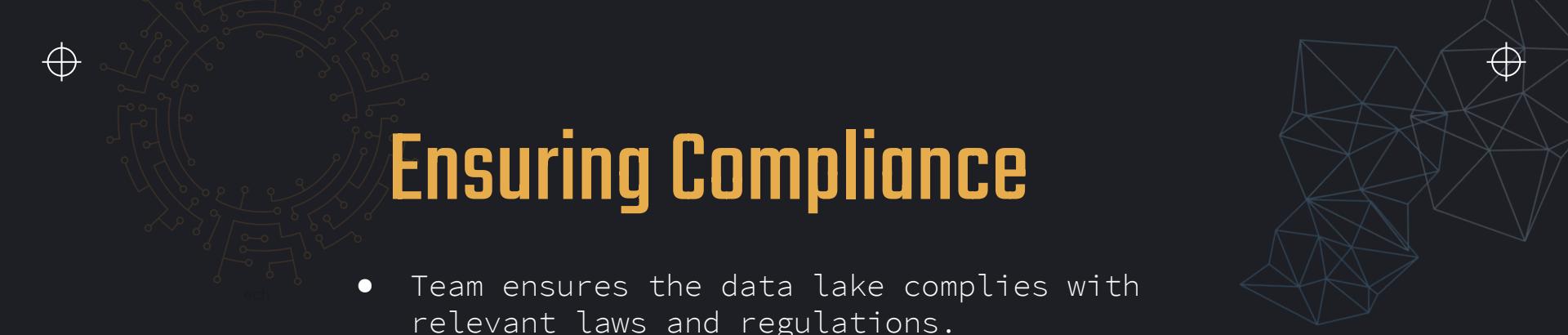




Defining Data Governance Standards

- Responsibilities of the data governance team include defining standards for:
 - ✓ Data access
 - ✓ Data quality
 - ✓ Security
 - ✓ Privacy





Ensuring Compliance

- Team ensures the data lake complies with relevant laws and regulations.
- Processes established for maintaining high data quality.

Next lecture will delve into how to define the exact data governance framework.





Defining Governance Standards



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



Data Quality Standards

- Develop a checklist for data quality attributes
- Implement routine processes for regular data quality checks.
- Use data quality tools or manual checks for duplicates, missing values.
- Generate monthly data quality reports for key datasets.



Data Access Policies

- Identify roles requiring access
- Define requirements for each role's data access and manipulation level.
- Implement Role-Based Access Control (RBAC) to enforce policies.
- Regularly review and update access roles and policies.





Data Security Standards

- Collaborate with data security officers to establish encryption standards.
- Conduct regular security audits to ensure compliance and effectiveness.





Data Privacy Compliance

- Align data governance policies with privacy regulations
- Develop standards for data storage, access, and processing in line with regulations.
- Consider data anonymization for sensitive personal information.
- Regularly review and update policies in compliance with regulations.





Documentation and Accessibility

- Document all relevant activities, audits, and policies.
- Create a central repository for storing and sharing governance documentation.
- Establish a data governance section on the company intranet for easy accessibility.

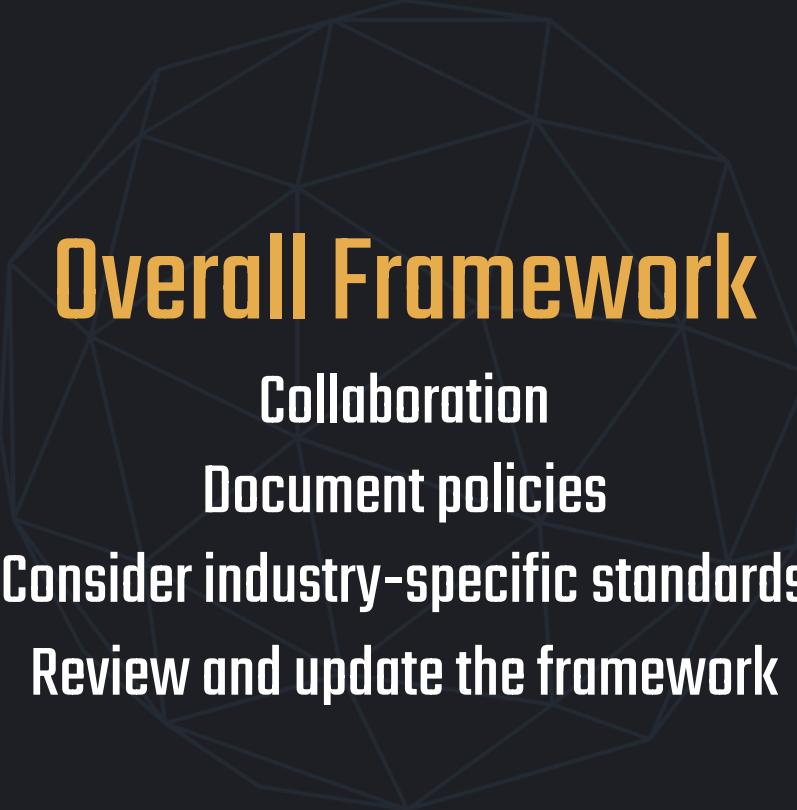




Training Program

- Develop a training program for relevant personnel on data governance policies.
- Ensure understanding of established procedures and standards.





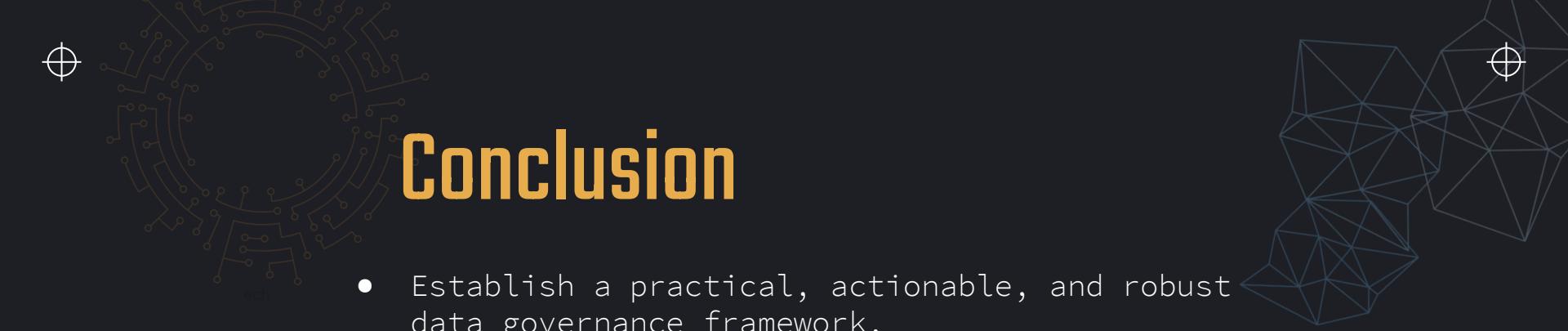
Overall Framework

Collaboration

Document policies

Consider industry-specific standards

Review and update the framework



Conclusion

- Establish a practical, actionable, and robust data governance framework.
- Ensure the framework is valuable, secure, and compliant with relevant regulations.
- Regularly review, update, and communicate policies to maintain effectiveness.



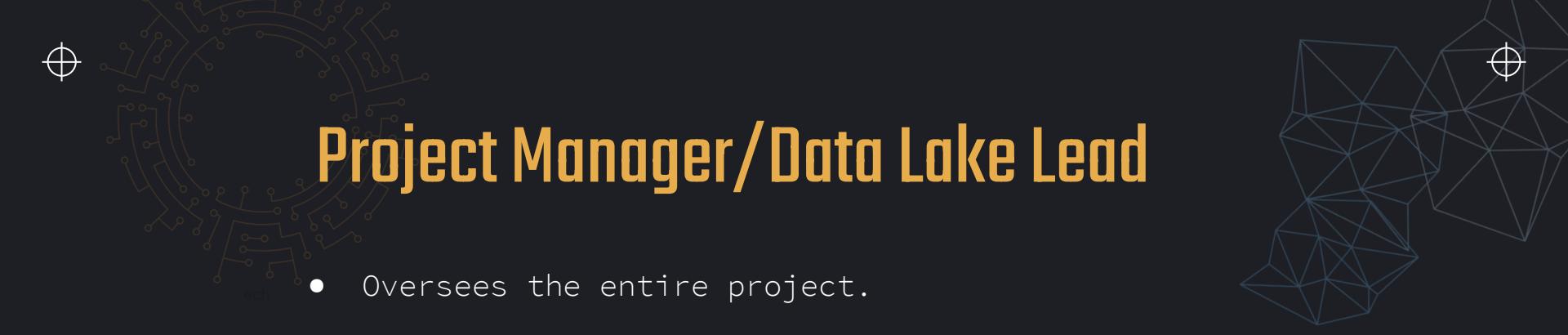
Setting up Data Lake Team



003-1040559

1250 003-77156.8

1760 0009-14563.7 73273



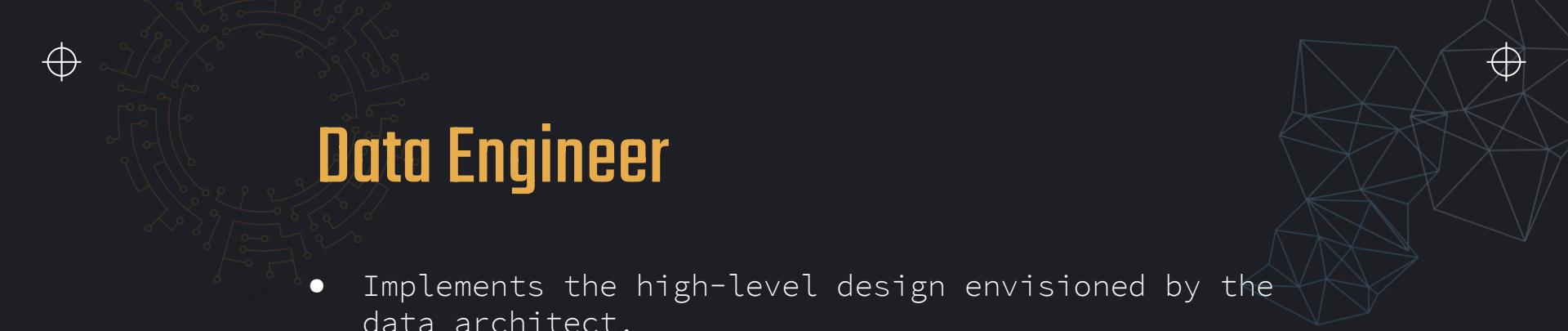
Project Manager/Data Lake Lead

- Oversees the entire project.
- Manages timelines, resources, and coordination between teams.
- Ensures project objectives are met.



Data Architect

- Designs the overall structure and strategy of the data lake.
- Focuses on high-level design, including data modeling and flow.



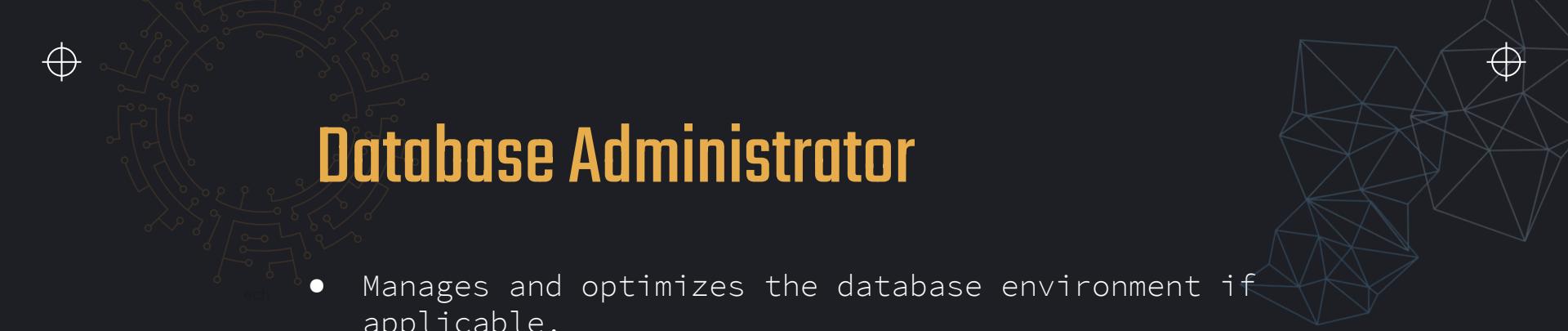
Data Engineer

- Implements the high-level design envisioned by the data architect.
- Responsible for building and maintaining data pipelines, ingestion, and storage.



Data Scientist

- Analyzes and derives insights from the data.
- May include data scientists building predictive or machine learning models.



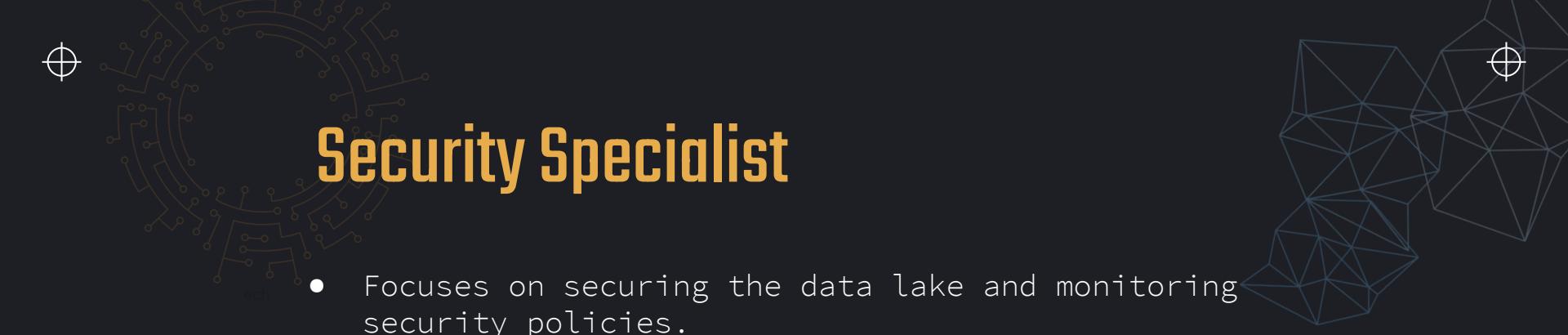
Database Administrator

- Manages and optimizes the database environment if applicable.



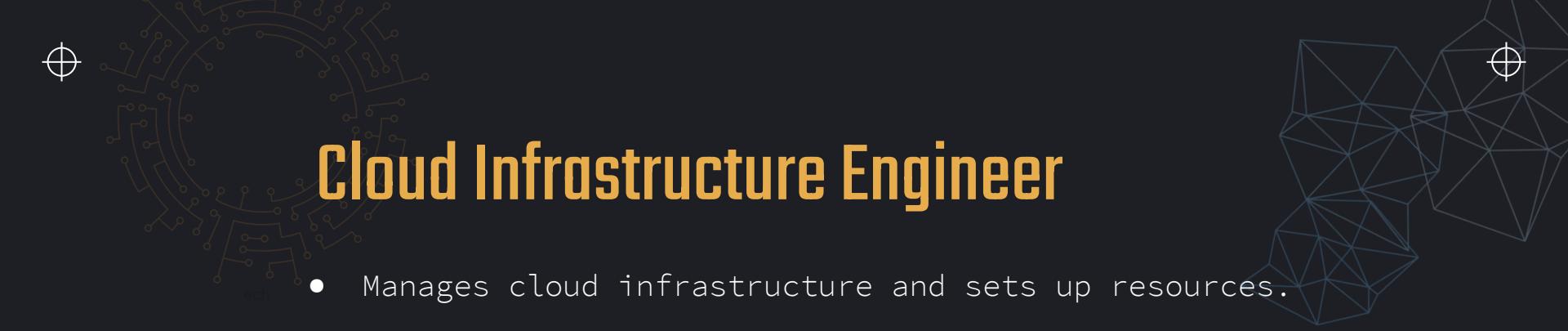
Data Governance Team

- Develops and enforces governance policies.
- Focuses on data quality, security, privacy, and compliance.



Security Specialist

- Focuses on securing the data lake and monitoring security policies.



Cloud Infrastructure Engineer

- Manages cloud infrastructure and sets up resources.

Setting Up an Effective Data Lake Team



Define Objectives



Balance Technical Aspects



Diversity of Roles



Encourage Collaboration



Soft Skills



Regular Meetings



Open Communication Channels



Success Factors



Diversity
Address various aspects of data management.

Communication
Foster open communication and collaboration.

Soft Skills
Recognize the importance of leadership and soft skills.





Roadmap Development



003-1040559

1250 003-77156.8

1760 0009-14563.7

73273

Planning and Design Phase

- Define architecture and goals.
- Plan for data integration and security.
- Develop data governance policies.
- **Set milestones:** Finalize data governance policies, select data storage types.
- **Define KPIs:** Number of approved design documents, time to finalize data governance standards.



Pilot Implementation Phase

- Start with a pilot project targeting a specific use case or limited set of data.
- Test design, gather initial feedback.
- Start small and evaluate the approach in a controlled manner.
- **Set milestones:** Completion of initial data ingestion, first round of user feedback on the pilot.
- **Define KPIs:** Time to deploy the pilot data lake, user satisfaction score.



Expansion Phase

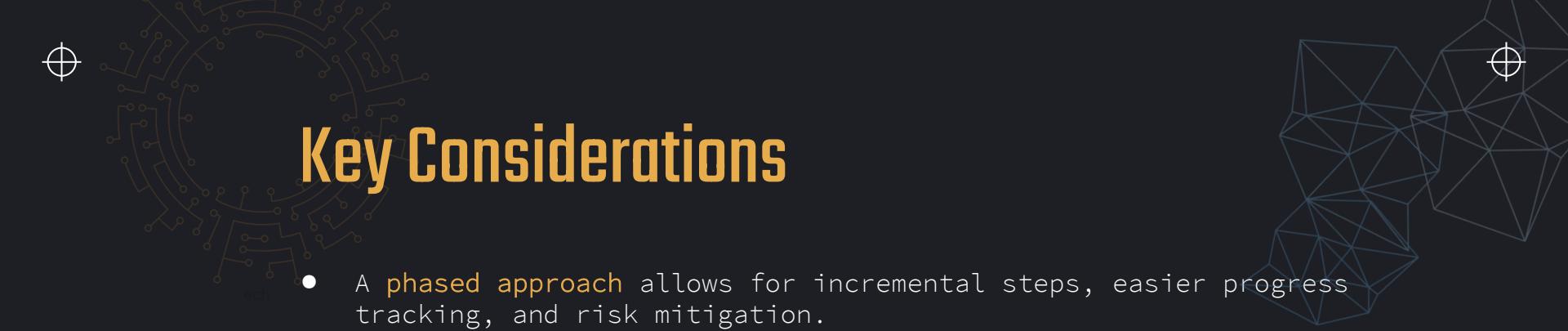
- Gradually expand the data lake to include more data sources and use cases.
- Based on the success of the pilot, expand to other departments or use cases.
- **Set milestones:** Implementation of additional data sources for specific departments.
- **Define KPIs:** Number of data sources fully integrated.



Optimization and Scaling Stage

- Continuously optimize and monitor the data lake.
- Adapt to new data, use cases, governance rules, and performance issues.
- **Set milestones:** Continuous improvement initiatives.
- **Define KPIs:** Adaptability to change, performance improvements.





Key Considerations

- A **phased approach** allows for incremental steps, easier progress tracking, and risk mitigation.
- **Milestones** help track the completion of key objectives at each phase.
- **KPIs** provide measurable indicators of success and progress.
- The **optimization and scaling stage** acknowledges the ongoing nature of data lake management and the need for continuous improvement.