Comprehensive AWS data engineering with Python and Lambda
Duration: 80 Hrs

Phyton:

Module 1: Introduction to Data Engineering
- What is Data Engineering?
- Challenges of Big Data and Traditional Data Processing
- The Role of Python and PySpark in Data Engineering
- Introduction to Distributed Computing Frameworks

Module 2: Python Fundamentals for Data Engineering
- Python Programming Basics (Data Types, Variables, Control Flow)
- Working with Functions, Modules, and Packages
- Data Structures in Python (Lists, Tuples, Dictionaries)
- Handling Text Data with Strings and Regular Expressions
- Setting up an Python environment for training.

Module 3: Apache Spark Introduction
- Introduction to Apache Spark and its Architecture
- Setting Up a Spark Environment (Local vs. Cluster)
- SparkSession: Interacting with Spark
- Creating and Working with Spark DataFrames

Module 4: Data Manipulation with PySpark
- Data Loading and Saving: Working with Various Data Sources (CSV, JSON, Parquet)
- Data Cleaning Techniques: Handling Missing Values, Outliers, and Inconsistencies
- Data Transformations with PySpark Functions (Filtering, Mapping)
- Working with Complex Data Structures (Nested DataFrames)

Module 5: Data Analysis and Aggregation
- Exploring and Understanding Data with PySpark
- Statistical Operations (Mean, Median, Standard Deviation)
- Grouping and Aggregation with PySpark Operations
- Window Functions for Advanced Data Analysis

Module 6: Data Joining and Combining
- Different Join Types in PySpark (Inner, Left, Right, Outer)
- Combining DataFrames using Union and Intersection
- Broadcasting for Efficient Joins with Small Datasets

Module 7: Working with Structured Streaming
- Introduction to Real-time Data Processing with Structured Streaming
- Building Streaming Data Pipelines in PySpark
- Windowing Operations in Streaming Data

Module 8: Introduction to Cloud Integration
- Deploying PySpark Applications on Cloud Platforms (AWS EMR, Databricks)
- Leveraging Cloud Storage for Data Management

Module 9: Hands-on Labs
- Setting Up Python and PySpark Environment
- Lab 1: Data Loading, Cleaning, & Transformation with PySpark
- Lab 2: Data Analysis & Aggregation with PySpark
- Lab 3: Data Joining & Combining Techniques
- Lab 4: Introduction to Structured Streaming

Lab 5: Cloud Integration with PySpark

Module 10: Resources and Next Steps

Python and PySpark Documentation
Best Practices for Data Engineering with PySpark
Additional Learning Resources
Projects and Case Studies in Data Engineering
Introduction to Machine Learning Libraries with PySpark (e.g., PyML)

AWS Glue:

Module 1: Introduction to Data Engineering and AWS Glue

What is Data Engineering?
The Role of AWS Glue in Data Engineering
Key Concepts: Data Lakes, ETL/ELT Pipelines, Data Catalog
AWS Glue Services Overview

Module 2: Migrating from Talend to AWS Glue

Challenges and Considerations in Migrating from Talend
Mapping Talend Components to AWS Glue Features
Automating ETL Job Conversion with Tools
Strategies for Handling Only EL
Strategies for Handling different types of Sources (API/Database/S3) for data extraction
Strategies for Handling Complex Transformations

Module 3: Building Data Pipelines with AWS Glue

Data Ingestion with AWS Glue: Crawlers and Jobs
Extracting, Transforming, and Loading (ETL) with Glue ETL Jobs
Data Transformation using Apache Spark in Glue
Scheduling and Monitoring Glue Jobs

Module 4: Advanced Topics in AWS Glue

Working with Dynamic Data: Delta Lake on AWS Glue
Using Glue Workflows for Orchestration
Integrating Glue with other AWS Services (S3, Redshift, Athena)
Security and Best Practices for AWS Glue

Module 5: Hands-on Labs

Setting Up the AWS Environment
Lab 1: Migrating a Simple Talend Job to Glue
Lab 2: Creating a Data Catalog and Crawler
Lab 3: Building an ETL Pipeline with Glue
Lab 4: Data Transformation using Spark in Glue
Lab 5: Scheduling and Monitoring Glue Jobs

Module 6: Resources and Next Steps

AWS Glue Documentation
AWS Glue Best Practices Guide
Additional Resources for Data Engineering on AWS
AWS Certifications for Data Engineering
Case Studies: Using AWS Glue in Real-world Scenarios

AWS Lambda:

Module 1: Introduction to Serverless Computing and AWS Lambda
- The Rise of Serverless Computing and its Benefits
- Understanding AWS Lambda: Event-Driven, Scalable Functions
- Key Concepts: Triggers, Handlers, Runtime Environments
- Benefits of Using Lambda for Data Engineering

Module 2: Building Data Processing Functions with Lambda
- Introduction to Python for AWS Lambda
- Writing Your First Lambda Function: Data Ingestion Example (e.g., S3 Trigger)
- Error Handling and Logging in Lambda Functions
- Implementing Data Transformation Logic within Lambda

Module 3: Integration with AWS Services for Data Engineering
- Using Lambda with S3 for Event-Driven Data Processing (e.g., New File Upload)
- Interacting with DynamoDB for Data Storage and Retrieval
- Integrating with Kinesis for Real-time Data Streams
- Leveraging Step Functions for Orchestrating Complex Workflows

Module 4: Advanced Lambda Techniques for Data Engineering
- Optimizing Lambda Performance: Memory, Timeouts, and Concurrency
- Asynchronous Programming with Lambda and SQS
- Unit Testing and Debugging Lambda Functions
- Security Considerations and Best Practices for Data Engineering with Lambda

Module 5: Hands-on Labs
- Setting Up the AWS Environment and AWS Toolkit
- Lab 1: Building a Simple Data Ingestion Lambda Function (S3 Trigger)
- Lab 2: Data Transformation with Lambda (e.g., CSV Processing)
- Lab 3: Integrating Lambda with DynamoDB
- Lab 4: Building a Data Processing Workflow with Step Functions (Optional)

Module 6: Resources and Next Steps
- AWS Lambda Documentation and Developer Guide
- Best Practices for Data Engineering with AWS Lambda
- Case Studies: Using Lambda for Real-world Data Workflows
- Exploring Advanced Serverless Architecture Patterns
- Building Serverless Data Pipelines with Lambda and Glue
- Cost Optimization Strategies for Serverless Data Processing