KRISHNAIK ACADEMY

Big Data Bootcamp With AWS & Azure

Big Data Bootcamp with AWS & Azure

This comprehensive program is designed to equip you with the skills and knowledge needed to thrive as a Big Data Engineer in today's cloud-driven world. Starting from foundational Big Data concepts and tools, you'll progress to mastering distributed computing frameworks like Apache Spark and Kafka.
The course emphasizes practical learning through **hands-on projects** and **real-world use cases**, focusing on integrating Big Data solutions with **Azure & AWS Cloud**. By the end of the program, you'll have the ability to design, build, and deploy scalable data pipelines, process massive datasets, and implement analytics workflows.

The course is completely **beginner friendly** with only pre-requisite being understanding of Python and basic Idea of databases and SQL which too will be provided as part of this course.

Learning Objectives

- **Build a Strong Foundation**: Understand the fundamentals of Big Data, distributed systems, and cloud computing.Learn the differences between batch and stream processing and where each fits in modern data pipelines.
- **Master Industry-Standard Tools and Frameworks** :Gain hands-on experience with Hadoop, HDFS, Apache Spark, and Apache Kafka. Explore the differences between on-premise systems and cloud-based architectures.
- **Work with Azure Cloud** :Understand the basics of Azure Cloud services, including storage, virtual machines, and networking.Use Azure Data Factory to ingest and transform data. Leverage Azure Databricks for scalable data processing and machine learning.
- **Develop Scalable Data Pipelines**: Learn to process structured and unstructured data efficiently. Build real-time streaming solutions with Kafka and Spark Streaming.
- **Focus on Optimization and Deployment**: Optimize workflows for performance using Spark tuning and partitioning. Deploy Big Data solutions in Azure for enterprise-scale applications.
- **Complete Real-World Projects**: Build and deploy an end-to-end data pipeline using Hadoop, Spark, and Azure. Implement a real-time analytics dashboard using Azure Synapse and Databricks.

Module 0

Course Prerequisites: Python, SQL, and Database Basics (Recorded)

This module is designed to prepare students with the foundational skills necessary for tackling the advanced concepts covered in the course. It focuses on the basics of Python programming, SQL, and databases, which are integral tools for any data engineering professional.
Understanding these concepts will help students interact with data, manipulate it programmatically, and write efficient queries to retrieve and process information. These skills will be directly applied in subsequent modules, such as working with Hadoop, Spark, Hive, NoSQL databases, and Azure-based services.
This module ensures that even students who are beginners in programming or databases can catch up and confidently dive into the rest of the course.

Topics	
Python Basics for Data Engineering	Introduction to Python fundamentals: working with variables, data types, conditional statements, loops, functions, and file handling.
Working with Python Libraries	Overview of essential Python libraries such as Pandas (for data manipulation), OS (for file handling), and Math (for basic calculations).
Introduction to SQL	Basics of SQL queries: SELECT, INSERT, UPDATE, and DELETE statements, and understanding database schemas.
Joins and Aggregations in SQL	Learn how to use JOINs (INNER, LEFT, RIGHT, FULL) to combine data from multiple tables and apply aggregate functions (e.g., SUM, AVG, COUNT) for analysis.
Introduction to Databases	Overview of relational databases (SQL-based) and NoSQL databases, including key differences, practical use cases, and database management concepts.

Module 1

Brief Overview of Big Data Concepts & Foundations

This section introduces the fundamentals of Big Data, exploring its importance in today's data-driven world. It explains the challenges of traditional data processing methods and how Big Data technologies address these challenges. The section also covers distributed systems, key characteristics of Big Data, and its applications across various industries. Finally, it introduces the Hadoop ecosystem and distributed storage and processing concepts, setting the stage for advanced topics in subsequent modules.

Topics	
Big Data Overview	Understand what Big Data is, its importance, and its defining characteristics (Volume, Velocity, Variety, Veracity, and Value
Challenges in Traditional Systems	Learn why traditional systems fail to handle massive datasets efficiently.
Distributed Systems Basics	Explore the fundamentals of distributed systems and how they solve scalability and fault-tolerance issues.
Batch vs. Stream Processing	Differentiate between batch and real-time data processing approaches and their use cases.
Applications of Big Data	Learn about real-world use cases of Big Data in industries like finance, healthcare, retail, and cloud computing.

Module 2

Deep Dive into Hadoop Architecture and Ecosystem

This module focuses on the Hadoop ecosystem, covering its architecture, core components, and the role of distributed storage and processing. Students will learn about the Hadoop Distributed File System (HDFS) and YARN, and how they enable the handling of massive datasets. The module also introduces essential tools in the Hadoop ecosystem that support data storage, processing, and management.

Topics	
Hadoop Architecture Overview	Learn the overall architecture of Hadoop, including its core components and how they work together.
HDFS (Hadoop Distributed File System)	Understand the design principles of HDFS, its block storage mechanism, and replication strategies for fault tolerance.
YARN (Yet Another Resource Negotiator)	Explore the role of YARN in resource management and scheduling across distributed systems.
MapReduce Framework	Learn how the MapReduce paradigm processes data using the map and reduce functions.
Hadoop Ecosystem Tools Overview	Get an introduction to ecosystem tools like Hive (data querying), Pig (data scripting), and HBase (NoSQL database).
Hadoop Use Cases	Explore real-world applications of Hadoop in industries such as finance, healthcare, and e-commerce.

Module 3

Foundations of Apache Spark: Architecture and Core Concepts

This module introduces Apache Spark as a unified data analytics engine for large-scale distributed data processing. It focuses on Spark's foundational architecture, core components, and its advantages over traditional systems like MapReduce. Students will gain hands-on experience with Spark's Resilient Distributed Datasets (RDDs) and will learn how to perform basic transformations and actions for batch data processing.

Topics	
Introduction to Apache Spark	Understand Spark's evolution, its significance in Big Data processing, and its advantages over traditional systems.
Spark Architecture Overview	Learn about Spark's architecture, including the Driver Program, Executors, and Cluster Managers (e.g., YARN).
Spark Execution Model	Explore Spark's Directed Acyclic Graph (DAG) execution model and its fault tolerance mechanisms.
Resilient Distributed Datasets (RDDs)	Understand the concept of RDDs, their creation, and their role as the core abstraction in Spark.
Transformations and Actions in Spark	Learn the difference between transformations (lazy evaluation) and actions (triggering computations).
Parallelism and Partitioning	Explore how Spark handles parallelism and data partitioning for efficient processing across clusters.
Introduction to Spark Deployments	Understand how Spark is deployed on clusters using standalone mode, YARN, and Kubernetes.

Module 4

DataFrames and Structured Data Processing with Spark

This module focuses on Spark's high-level APIs for structured data processing, emphasizing the use of DataFrames and Spark SQL. Students will learn how to work with structured and semi-structured data, perform SQL-like queries, and optimize data processing tasks. This module also highlights Spark's Catalyst optimizer and Tungsten execution engine, providing insights into Spark's efficiency and performance.

Topics	
Introduction to DataFrames	Understand DataFrames as a distributed collection of data organized into named columns, and how they differ from RDDs.
Spark SQL Basics	Learn to query structured data using Spark SQL and integrate it with DataFrames for seamless processing.
Schema Management	Explore how to define, infer, and manage schemas for structured and semi-structured data.
Optimization with Catalyst	Dive into Spark's Catalyst optimizer for query optimization and understand its role in performance tuning.
Hands-on with DataFrames	Practice operations like filtering, grouping, joining, and aggregating data using DataFrames in PySpark.

Module 5

Advanced Data Processing and Optimization with Spark

This module covers the advanced topics of Spark that focus on optimizing large-scale data processing tasks. Students will dive into Spark internals, understand its underlying architecture, and learn advanced techniques such as caching, partitioning, and performance tuning for optimal processing. The goal of this module is to equip students with the knowledge to handle complex big data pipelines with Spark efficiently and at scale.

Topics	
Understanding Spark Internals	Learn the internal workings of Spark, including the role of the Spark Scheduler, DAG scheduler, and task execution.
Caching and Persistence in Spark	Explore Spark's caching and persistence mechanisms to store intermediate RDDs and DataFrames in memory for faster access.
Data Partitioning and Shuffling	Understand how data is partitioned in Spark and how shuffling occurs during transformations like joins and groupBy.
Performance Tuning in Spark	Learn strategies for optimizing Spark performance, including managing resources, optimizing query plans, and fine-tuning execution.
Optimizing Spark Jobs with Configurations	Dive into tuning Spark configurations to maximize efficiency, focusing on driver and executor memory, number of partitions, and parallelism.

Module 6

Spark Performance Optimization and Advanced Tuning

This module will focus entirely on advanced performance optimization techniques for Spark. After covering the basics and intermediate performance tuning in earlier modules, this section will dive into the finer details of optimizing Spark jobs, tuning Spark for large datasets, and improving the overall efficiency of Spark workloads. Students will also explore best practices for performance troubleshooting and debugging Spark jobs.

Topics	
Advanced Spark SQL Optimizations	Learn how to optimize Spark SQL queries, including query plan optimization, partition pruning, and predicate pushdown.
Tuning Spark for Large-Scale Data	Understand the strategies for tuning Spark when working with very large datasets, such as handling skew and data repartitioning.
Executor and Memory Management	Dive deeper into managing executor memory, fine-tuning garbage collection, and adjusting Spark configurations for better performance.
Spark Shuffle Optimization	Learn how to optimize shuffle operations in Spark, focusing on reducing shuffle size, controlling shuffle partitions, and preventing shuffle spill.
Performance Troubleshooting and Debugging	Explore techniques for troubleshooting slow Spark jobs, identifying bottlenecks, and debugging issues related to memory usage and task execution.

Module 7

Introduction to NoSQL Databases and Comparison with SQL Databases

This module provides an introduction to NoSQL databases, discussing their key characteristics and advantages over traditional SQL databases. Students will learn about different types of NoSQL databases, including document stores, key-value stores, column-family stores, and graph databases. The module also includes a detailed comparison between SQL and NoSQL databases, helping students understand when to use each type based on the application needs.

Topics	
Introduction to NoSQL Databases	Understand the core principles of NoSQL, including scalability, flexibility, and schema-less designs.
Types of NoSQL Databases	Learn about the four major types of NoSQL databases: document stores, key-value stores, column-family stores, and graph databases.
NoSQL vs SQL	Compare NoSQL and SQL databases, focusing on their differences in structure, scalability, and consistency models.
Advantages of NoSQL Databases	Explore the benefits of using NoSQL, including horizontal scalability, flexibility in handling semi-structured data, and speed in handling large volumes of data.
Use Cases for NoSQL	Learn when to use NoSQL databases, with real-world examples such as social media platforms, IoT applications, and big data analytics.

Module 8

MongoDB: Document-Based NoSQL Database

This module focuses on MongoDB, a popular document-based NoSQL database. Students will learn about MongoDB's architecture, data model, and query language. Hands-on practice will include creating collections, inserting documents, querying data, and performing aggregations using MongoDB's powerful features.

Topics	
Introduction to MongoDB	Understand the core concepts of MongoDB, including its document-based storage model and JSON-like format (BSON).
MongoDB Architecture	Learn how MongoDB stores data, with an emphasis on collections, documents, and indexes.
CRUD Operations in MongoDB	Learn how to perform basic CRUD operations (Create, Read, Update, Delete) in MongoDB.
Aggregation Framework	Explore MongoDB's aggregation framework for complex queries and data transformations.
MongoDB Indexing and Performance	Learn how to optimize MongoDB queries using indexing and other performance optimization techniques.

Module 9

Cassandra: Column-Family Based NoSQL Database

In this module, students will explore Cassandra, a column-family-based NoSQL database designed for handling large-scale, high-velocity data. Cassandra is widely used for applications that require high availability and fault tolerance. This module will cover its architecture, query language (CQL), and how to scale Cassandra for massive datasets.

Topics	
Introduction to Cassandra	Understand the key features of Cassandra, including its distributed architecture and horizontal scalability.
Cassandra Architecture	Learn about the Cassandra architecture, including nodes, clusters, and the concept of eventual consistency.
Cassandra Query Language (CQL)	Learn how to use CQL, Cassandra's query language, to interact with the database.
Data Modeling in Cassandra	Understand how to model data in Cassandra, including partition keys, clustering keys, and table design principles.
Cassandra Performance and Scaling	Learn strategies for optimizing Cassandra, including replication, tuning, and data distribution.

Module 10

Hive Architecture, Setup, and Basic Operations

In this module, students will be introduced to Apache Hive—a data warehouse system built on top of Hadoop that provides SQL-like querying capabilities for Big Data. The focus will be on understanding Hive's architecture, setting up a Hive environment, and performing basic operations like creating databases and tables. The module will also cover loading data from both local and HDFS sources into Hive.

Topics	
Hive Architecture Overview	Understand the core components of Hive, such as the Hive Metastore, Driver, Compiler, and Execution Engine.
Setting Up Hive on Hadoop Cluster	Learn the process of setting up Hive locally on a Hadoop cluster, including installation and configuration steps.
Hive Data Types	Explore Hive's data types, including primitive types (int, string) and complex types (array, map).
Creating Databases and Tables	Learn how to create databases and tables in Hive, including specifying column types and table partitions.
Loading Data into Hive	Cover the process of loading data into Hive from local storage and HDFS (both internal and external tables).

Module 11

Advanced Hive Features: Partitioning, Optimization, and Performance Tuning

This module dives deeper into advanced Hive features, such as partitioning, bucketing, and optimization techniques that enhance query performance. Students will also explore various SerDe (Serialization and Deserialization) formats like CSV, JSON, Parquet, and ORC, and their impact on data handling. The module also covers advanced join optimizations like Map-Side Join, Sorted Merge Join, and Skew Join.

Topics	
Internal vs. External Tables	Understand the difference between internal and external tables in Hive and their practical uses.
Complex Data Types in Hive	Learn how to work with complex data types like Arrays, Maps, and Structs for flexible data storage.
Hive SerDe (Serialization/Deserialization)	Explore various SerDe options in Hive, including CSV, JSON, Parquet, and ORC formats for handling diverse data.
Partitioning in Hive	Understand both static and dynamic partitioning in Hive and how they help optimize data queries.
Bucketing and Performance Tuning	Learn how bucketing helps distribute data across files and optimizes query performance for large datasets.
Join Optimizations in Hive	Learn about advanced join techniques like Map-Side Join, Sorted Merge Join, and Skew Join for optimizing large queries.

Module 12

Introduction to Kafka: Architecture and Core Concepts

In this module, students will be introduced to Apache Kafka, understanding its distributed architecture and core components, including brokers, topics, partitions, and producers/consumers. The focus will be on learning how Kafka ensures fault tolerance, high availability, and scalability.

Topics	
Kafka Cluster Architecture	Understand the architecture of a Kafka cluster, including brokers, topics, and partitions.
Producer-Consumer Model	Learn how Kafka handles data flow through producers and consumers, and how consumer groups operate.
Offset Management	Explore how Kafka manages offsets for consumers to track message processing.
Replication and Fault Tolerance	Understand how replication ensures data availability and fault tolerance in Kafka.
Synchronous and Asynchronous Commits	Learn the differences between sync and async commits and their implications on performance.

Module 13

Working with Kafka Producers, Consumers, and Message Formats

This module focuses on practical aspects of working with Kafka, covering producer-consumer implementation, message formats (JSON, CSV), and Kafka's Schema Registry for schema management.

Topics	
Kafka Producer and Consumer Code	Learn to write Kafka producer-consumer code with serialization and deserialization.
Schema Registry	Understand how to use Schema Registry for managing message schemas and ensuring consistency in Kafka.
Message Key-Value Pairs in Kafka	Explore working with key-value pairs for Kafka messages.
Working with JSON, CSV Data	Learn to send and consume JSON and CSV formatted data using Kafka.
Producer and Consumer in Consumer Groups	Learn the concept of consumer groups and how they manage parallel processing of Kafka messages.

Module 14

Spark Structured Streaming: Real-Time Data Processing with Kafka

In this module, students will learn to integrate Kafka with Spark Structured Streaming for real-time data processing. The module will cover the fundamentals of Spark Structured Streaming, including how to consume data from Kafka topics and process it in real-time.

Topics	
Introduction to Spark Structured Streaming	Understand the fundamentals of Spark Structured Streaming, a scalable real-time data processing engine.
Kafka Integration with Spark	Learn how to consume data from Kafka topics in Spark Structured Streaming for real-time processing.
Stream Processing with Spark	Explore streaming DataFrames, stream transformations, and handling window operations.
Stateful vs Stateless Transformations	Learn about stateful and stateless transformations in streaming applications.
Fault Tolerance and Checkpointing	Explore how checkpointing ensures fault tolerance in Spark Structured Streaming jobs.

Module 15

Introduction to Apache Airflow: Orchestration and Dependency Management in Data Pipelines

In this module, students will learn about data pipeline orchestration using Apache Airflow, a platform designed to programmatically author, schedule, and monitor workflows. The module will cover the basics of orchestration in Big Data, the need for dependency management in data pipelines, and an in-depth understanding of Airflow's architecture, including its components and operators. Students will also get hands-on experience with creating and scheduling DAGs (Directed Acyclic Graphs), managing task dependencies, and running parallel tasks.

Topics	
What is Orchestration in Big Data?	Understand the concept of orchestration and its role in automating the execution of data pipelines in Big Data environments.
Need for Dependency Management in Data Pipeline Design	Learn why dependency management is crucial in ensuring tasks are executed in the correct order and how it prevents issues in complex data workflows.
What is Apache Airflow?	Get an introduction to Apache Airflow, its purpose in data pipeline orchestration, and its role in the Big Data ecosystem.
Architecture and Components of Airflow	Explore the key components of Airflow, including Scheduler, Executor, Web UI, and Metastore, and understand how they work together to execute workflows.
Airflow Operators	Learn about the operators in Airflow, such as BashOperator and PythonOperator, and their role in task execution.
Writing Airflow DAG Scripts	Understand how to write DAG scripts in Airflow, including the basic structure, task dependencies, and scheduling.
Executing Parallel Tasks in Airflow	Learn how to configure parallel task execution in Airflow to run multiple tasks concurrently.

Module 16

Introduction to Cloud Computing and Overview of Azure

This module introduces cloud computing, its key components, and how it relates to Big Data. The focus will be on Azure Cloud as a platform for scalable, secure, and cost-effective Big Data engineering. We will provide an overview of Azure's services, explaining the different cloud models (IaaS, PaaS, SaaS) and how these apply to Big Data workflows.

Topics	
What is Cloud Computing?	Introduction to cloud computing: definition, importance, and how it has transformed modern IT systems.
Cloud Service Models	Overview of the three primary cloud models: IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service).
Benefits of Cloud for Big Data	Explore the benefits of using the cloud for Big Data engineering: scalability, flexibility, cost efficiency, and on-demand computing.
Overview of Azure	Introduction to Azure and its role in the cloud ecosystem, highlighting key services for data engineering.
Azure Global Infrastructure	Learn about Azure's data centers, regions, and availability zones and their importance for high-availability systems.

Module 17

Azure Storage Services for Big Data

This module focuses on the various Azure Storage Services, including Blob Storage and Azure Data Lake Storage Gen2, which are critical for storing and managing large datasets for Big Data applications.

Topics	
Azure Storage Overview	Learn about Azure's storage solutions and their roles in storing data for Big Data engineering.
Azure Blob Storage	Introduction to Azure Blob Storage, its use cases, and data management techniques for unstructured data.
Azure Data Lake Storage Gen2	Explore ADLS Gen2, its integration with HDFS, and its hierarchical namespace for managing large-scale data.
Storage Tiers in Azure	Understand the Hot, Cool, and Archive tiers in Blob Storage for cost-effective data management.
Setting up Blob Storage and ADLS Gen2	Hands-on setup and configuration of Blob Storage and Azure Data Lake Storage Gen2.

Module 18

Introduction to Azure Databricks

This module covers Azure Databricks, a powerful platform for Apache Spark. Students will learn how to create and manage Databricks clusters, use notebooks for data processing, and perform data transformations with Spark.

Topics	
What is Azure Databricks?	Learn about Azure Databricks, a unified analytics platform for big data processing and machine learning.
Setting up Databricks Workspace	Understand how to create a Databricks workspace in Azure and configure clusters for distributed data processing.
Databricks Notebooks	Explore the use of Databricks notebooks for data analysis, using Spark and SQL.
Integrating Apache Spark with Databricks	Understand how Apache Spark integrates with Databricks for scalable data engineering and analytics.
Azure Databricks Pricing	Learn about pricing models for Azure Databricks and how to optimize cluster usage for cost efficiency.

Module 19

Azure Data Factory - Data Orchestration

This module introduces Azure Data Factory (ADF), a cloud service for orchestrating data workflows. Students will learn how to create data pipelines, schedule data movements, and monitor the performance of their pipelines.

Topics	
Introduction to Azure Data Factory	Learn about Azure Data Factory (ADF) and its role in creating and managing ETL and ELT pipelines.
Creating Data Pipelines in ADF	Understand how to create data pipelines for automating data ingestion, transformation, and loading tasks.
Working with Datasets and Linked Services	Learn about datasets and linked services in ADF to define source and destination data locations.
Scheduling Pipelines in ADF	Understand how to schedule pipelines in ADF and automate data movement between various sources and sinks.
Monitoring and Troubleshooting in ADF	Learn how to monitor pipeline executions and troubleshoot common issues in ADF pipelines.

Module 20

Advanced Data Factory – Transformations, Monitoring, and Error Handling

Building on ADF basics, this module dives deeper into more advanced data transformation capabilities and introduces monitoring, error handling, and logging in ADF pipelines.

Topics	
Advanced Data Transformations in ADF	Learn how to apply data transformations using Mapping Data Flows and other ADF transformation tools.
Error Handling and Logging	Understand error handling and logging best practices in ADF to ensure pipeline robustness.
Data Flow Debugging and Optimization	Learn how to debug and optimize data flows in ADF, improving performance in large-scale workflows.
Monitoring Pipelines	Explore advanced monitoring techniques in ADF for optimizing pipeline performance and identifying bottlenecks.
ADF Integration with Other Azure Services	Understand how ADF integrates with other Azure services such as Azure Databricks, Azure Functions, and Azure Synapse.

Module 21

AWS EMR: Scalable Big Data Processing with Elastic MapReduce

This module introduces AWS EMR (Elastic MapReduce), a managed Big Data processing service that simplifies running Hadoop, Spark, and other distributed frameworks on AWS. Students will learn to set up and configure EMR clusters for scalable data processing, use Hadoop MapReduce and Spark for distributed jobs, and integrate with S3 for input and output data. The module also covers tracking, debugging, and optimizing jobs.

Topics	
What is AWS EMR?	Introduction to Elastic MapReduce, its architecture, and how it simplifies Big Data workflows using managed clusters.
EMR Cluster Setup and Configuration	Learn to create and configure an EMR cluster, including selecting appropriate instance types, node types (Master, Core, and Task Nodes), and scaling options.
Hadoop and Spark on EMR	Understand how to run distributed Hadoop MapReduce and Apache Spark jobs on EMR clusters.
EMR and S3 Integration	Learn to store input data in S3, process it using EMR, and save the output back to S3 for scalability and cost-efficiency.
Monitoring and Optimizing EMR Jobs	Explore tools for tracking job progress, debugging issues, and tuning cluster performance for faster execution.

Module 22

AWS S3: Scalable and Cost-Effective Data Storage for Big Data

This module covers Amazon S3 (Simple Storage Service), a highly durable, scalable, and cost-effective storage solution that is central to AWS Big Data workflows. Students will learn how to create and manage S3 buckets, work with data tiers, secure data using IAM roles, and transfer data programmatically.

Topics	
Introduction to Amazon S3	Overview of Amazon S3, its role as a storage solution for Big Data, and its ability to store massive datasets efficiently.
S3 Bucket Management	Learn to create, configure, and manage S3 buckets for organizing data, including applying access policies.
S3 Storage Classes	Understand S3's storage tiers (Standard, Intelligent-Tiering, Glacier) and their cost-efficiency for different data usage scenarios.
Versioning and Lifecycle Policies	Learn to use versioning to track file changes and set lifecycle policies for archiving or deleting unused data.
Securing Data in S3	Explore how to secure S3 buckets using IAM roles, encryption mechanisms, and access control lists (ACLs).

Module 23

AWS Athena and Glue: Serverless Querying and ETL for Big Data

This module focuses on AWS Athena for serverless querying of data stored in S3 using SQL and AWS Glue for managing ETL (Extract, Transform, Load) workflows. Students will learn to catalog data, perform schema discovery, and query large datasets efficiently.

Topics	
Introduction to AWS Athena	Overview of Athena, its serverless architecture, and how it simplifies querying structured and semi-structured data stored in S3.
Setting Up Athena for Querying Data	Learn to configure Athena, define external tables, and run SQL queries on data stored in S3.
Optimizing Athena Queries	Techniques to optimize query performance by using partitioning, compression, and file formats like Parquet.
Introduction to AWS Glue	Understand the role of AWS Glue in creating data catalogs, schema discovery, and automating ETL processes.
Using Glue Crawlers	Learn how to set up Glue crawlers to infer data schemas and create metadata tables for use in Athena.

Capstone Project - 1

Capstone Project 1: Big Data ETL Pipeline with Hadoop and Hive

This project will focus on designing an ETL (Extract, Transform, Load) pipeline to process raw data stored in HDFS, transform it using Spark, and load it into Hive for querying and analysis. This project simulates real-world scenarios where businesses need to ingest large datasets, process them to extract meaningful information, and store them in queryable formats.

Outline :

- **Data Source**: Start with raw data stored locally or ingested from a file source such as CSV or logs.
- **ETL Process**:
  - Extract: Use Hadoop to ingest raw data into HDFS.
  - Transform: Clean and preprocess the data using a MapReduce job, performing tasks like filtering, deduplication, and aggregations.
  - Load: Save the processed data into Hive tables for querying and reporting.
- **Querying Data**: Use HiveQL to perform operations such as grouping, filtering, and aggregating data for business insights.

Key Technologies:

- Hadoop (HDFS, Spark): Modules 2–5.
- Hive: Modules 10–11.

Outcome:

By completing this project, students will:

- Understand how to design an end-to-end batch processing pipeline using Hadoop and Hive.
- Gain experience with HDFS storage, MapReduce, and HiveQL for Big Data analytics.

Capstone Project - 2

Capstone Project 2: Real-Time Data Processing with Kafka and Spark Streaming

This project aims to build a real-time streaming pipeline for processing data on the fly using Apache Kafka and Spark Streaming. Students will process a continuous stream of events (e.g., log data, IoT sensor readings, or clickstream data) and generate meaningful insights in real time.

Outline :

- **Data Source**: Simulate real-time data streams using a Kafka producer (e.g., sending IoT sensor readings or stock prices).
- **Pipeline Components**:
  - Apache Kafka: Use Kafka to manage the stream of incoming data with appropriate topics and partitions.
  - Spark Streaming: Consume the Kafka stream, process data in real-time (e.g., compute rolling averages or identify anomalies), and write results to HDFS or S3 for further analysis.
- **Output**: Store processed data in a NoSQL database like Cassandra or MongoDB (covered in earlier modules) for querying and visualization.

Key Technologies:

- Apache Kafka: Modules 12–13.
- Spark Streaming: Module 14.
- NoSQL Database Integration: Modules 7–9.

Outcome:

By completing this project, students will:

- Learn to create and manage real-time data pipelines.
- Apply streaming analytics for fast, event-driven insights.
- Showcase the ability to work with Kafka, Spark Streaming, and NoSQL databases in a single workflow.

Capstone Project - 3

Capstone Project 3: Data Lakehouse with Azure Databricks and Data Factory

Design a scalable data lakehouse architecture using Azure Databricks for data processing and Azure Data Factory for orchestrating pipelines. This project mimics modern cloud-based Big Data architectures used in data engineering.

Outline :

- **Data Source**: Use structured and semi-structured datasets, such as sales transactions or JSON logs, stored in Azure Data Lake Storage Gen2 (ADLS).
- **Pipeline Components**:
  - Azure Data Factory (ADF): Create pipelines to ingest data into ADLS from external sources (e.g., APIs or on-prem databases).
  - Azure Databricks: Process data using PySpark in Databricks notebooks, performing tasks like cleaning, joining, and aggregating data.

- **Output**: Save processed data in an optimized format (e.g., Parquet) for downstream BI tools or analytics..

Key Technologies:

- Azure Data Factory (ADF): Modules 19–21.
- Azure Databricks: Modules 18–21.
- Azure Data Lake Storage Gen2 (ADLS): Module 20.

Outcome:

By completing this project, students will:

- Gain experience with modern cloud-based data lakehouse architectures.
- Learn to integrate Databricks, ADF, and ADLS for scalable workflows.
- Prepare for real-world cloud-based data engineering challenges.

Capstone Project - 4

Capstone Project 4: Serverless Data Analytics with AWS Glue and Athena

This project focuses on building a serverless data analytics solution using AWS Glue for ETL workflows and AWS Athena for SQL-based querying on large datasets stored in S3. This setup represents a lightweight and cost-effective Big Data solution.

Outline :

- **Data Source**: Store a raw dataset in S3 (e.g., customer logs or product data).
- **Pipeline Components**:
  - AWS Glue Crawlers: Use Glue crawlers to automatically discover schemas and create a data catalog.
  - AWS Glue Jobs: Write transformation scripts to clean and preprocess the data, converting it into an optimized format like Parquet or ORC.
  - AWS Athena: Query the cataloged data in S3 using SQL to perform analysis, such as generating reports or KPIs.
- **Output**: Visualize the results using AWS QuickSight or export them to BI tools like Tableau.

Key Technologies:

- AWS S3: Module 23.
- AWS Glue: Module 24.
- AWS Athena: Module 24.

Outcome:

By completing this project, students will:

- Master serverless tools like Glue and Athena for Big Data analytics.
- Learn to catalog, transform, and query large datasets efficiently.
- Build cost-effective, serverless data engineering workflows.