AWS Data Engineering Training Curriculum

Data Engineering using AWS Analytics Services

Introduction to the course

- Introduction to Data Engineering using AWS Analytics Services

Setup Local Environment for Practice

- Setup Jupyter Lab using Docker
- Understanding Jupyter Lab Environment
- Install Python boto3
- Running Shell Commands
- Install AWS CLI

Setup Environment for Practice using Cloud9

- Introduction to Cloud9
- Setup Cloud9
- Overview of Cloud9 IDE
- Docker and AWS CLI on Cloud9
- Cloud9 and EC2
- Accessing Web Applications
- Allocate and Assign Static IP
- Changing Permissions using IAM Policies
- Increasing Size of EBS Volume
- Opening ports for Cloud9 Instance
- Setup Jupyter lab on Cloud9 Instance
- Open SSH Port for Cloud9 EC2 Instance
- Connect to Cloud9 EC2 Instance using SSH

AWS Getting Started

- Introduction - AWS Getting Started
- Create s3 Bucket
- Create IAM Group and User
- Overview of Roles
- Create and Attach Custom Policy
- Configure and Validate AWS CLI

Storage - All about AWS s3 (Simple Storage Service)

- Getting Started with S3
- Setup Data Set locally
- Adding S3 Buckets and Objects
- Version Control in S3
- Cross-Region Replication
- Overview of S3 Storage Classes
- Overview of Glacier
- Managing S3 using AWS CLI
- Managing Objects in S3 using CLI – Lab

User Level Security - Managing Users, Roles and Policies using IAM

- Creating IAM Users
- Logging into AWS Management Console using IAM User
- Validate Programmatic Access to IAM User
- IAM Identity-based Policies
- Managing IAM Groups
- Managing IAM Roles
- Overview of Custom Policies
- Managing IAM using AWS CLI

Infrastructure - AWS EC2 (Elastic Cloud Compute) Basics

- Getting Started with EC2
- Create EC2 Key Pair
- Launch EC2 Instance
- Connecting to EC2 Instance
- Security Groups Basics
- Public and Private IP Addresses
- EC2 Life Cycle
- Allocating and Assigning Elastic IP Address
- Managing EC2 Using AWS CLI
- Upgrade or Downgrade EC2 Instances

Data Ingestion using Lambda Functions

- Hello World using AWS Lambda
- Setup Project for local development
- Deploy Project to AWS Lambda console
- Develop download functionality using requests
- Using 3rd party libraries in AWS Lambda
- Validating s3 access for local development
- Develop upload functionality to s3
- Validating using AWS Lambda Console
- Run using AWS Lambda Console
- Validating files incrementally
- Reading and Writing Bookmark using s3
- Maintaining Bookmark using s3
- Review the incremental upload logic
- Deploying lambda function
- [Instructions and Source Code] - ghactivity-downloader Lambda Function
- Schedule Lambda Function using AWS Event Bridge

Development Lifecycle for Pyspark

- Setup Virtual Environment and Install Pyspark
- Getting Started with Pycharm
- Passing Run Time Arguments
- Accessing OS Environment Variables
- Getting Started with Spark
- Create Function for Spark Session
- Setup Sample Data
- Read data from files
- Process data using Spark APIs
- Write data to files
- Validating Writing Data to Files
- Productionizing the Code

Overview of Glue Components

- Introduction - Overview of Glue Components
- Create Crawler and Catalog Table
- Analyze Data using Athena
- Creating S3 Bucket and Role
- Create and Run the Glue Job
- Validate using Glue CatalogTable and Athena
- Create and Run Glue Trigger
- Create Glue Workflow

- Run Glue Workflow and Validate

Setup Spark History Server for Glue Jobs

- Introduction - Spark History Server for Glue
- Setup Spark History Server on AWS
- Clone AWS Glue Samples repository
- Build Glue Spark UI Container
- Update IAM Policy Permissions
- Start Glue Spark UI Container

Deep Dive into Glue Catalog

- Prerequisites for Glue Catalog Tables
- Steps for Creating Catalog Tables
- Download Data Set
- Upload data to s3
- Create Glue Catalog Database - itvghlandingdb
- Create Glue Catalog Table - ghactivity
- Running Queries using Athena - ghactivity
- Crawling Multiple Folders
- Managing Glue Catalog using AWS CLI
- Managing Glue Catalog using AWS CLI
- Managing Glue Catalog using Python Boto3

Exploring Glue Job APIs

- Update IAM Role for Glue Job
- Generate baseline Glue Job
- Running baseline Glue Job
- Glue Script for Partitioning Data
- Validating using Athena

Glue Job Bookmarks

- Introduction to Glue Job Boomarks
- Cleaning up the data
- Overview of AWS Glue CLI
- Run Job using Bookmark
- Validate Bookmark using AWS CLI
- Add new data to landing
- Rerun Glue Job using Bookmark
- Validate Job Bookmark and Files for Incremental run
- Recrawl the Glue Catalog Table using CLI

- Run Athena Queries for Data Validation

Getting Started with AWS EMR

- Planning of EMR Cluster
- Create EC2 Key Pair
- Setup EMR Cluster with Spark
- Understanding Summary of AWS EMR Cluster
- Review EMR Cluster Application User Interfaces
- Review EMR Cluster Monitoring07 Review EMR Cluster Monitoring
- Review EMR Cluster Hardware and Cluster Scaling Policy
- Review EMR Cluster Configurations
- Review EMR Cluster Events
- Review EMR Cluster Steps
- Review EMR Cluster Bootstrap Actions
- Connecting to EMR Master Node using SSH
- Disabling Termination Protection and Terminating the Cluster
- Clone and Create New Cluster
- Listing AWS S3 Buckets and Objects using AWS CLI on EMR Cluster
- Listing AWS S3 Buckets and Objects using HDFS CLI on EMR Cluster
- Managing Files in AWS s3 using HDFS CLI on EMR Cluster

Deploying Spark Applications using AWS EMR

- Deploying Applications using AWS EMR - Introduction
- Setup EMR Cluster to deploy applications
- Validate SSH Connectivity to Master node of AWS EMR Cluster
- Setup Jupyter Notebook Environment on EMR Cluster
- Create required AWS s3 Bucket
- Upload GHActivity Data to s3
- Validate Application using AWS EMR Compatible Versions
- Deploy Application to AWS EMR Master Node
- Create user space for ec2-user on AWS EMR Cluster
- Run Spark Application using spark-submit on AWS EMR Master Node
- Validate Data using Jupyter Notebooks on AWS EMR Cluster
- Clone and Start Auto Terminated AWS EMR Cluster
- Delete Data Populated by GHAcitivity Application using AWS EMR Cluster
- Differences between Spark Client and Cluster Deployment Modes
- Running Spark Application using Cluster Mode on AWS EMR Cluster
- Overview of Adding Pyspark Application as Step to AWS EMR Cluster
- Deploy Spark Application to AWS S3
- Running Spark Applications as AWS EMR Steps in client mode
- Running Spark Applications as AWS EMR Steps in cluster mode
- Validate AWS EMR Step Execution of Spark Application

Streaming Pipeline using Kinesis

- Building Streaming Pipeline using Kinesis
- Rotating Logs
- Setup Kinesis Firehose Agent
- Create Kinesis Firehose Delivery Stream
- Planning the Pipeline
- Create IAM Group and User
- Granting Permissions to IAM User using Policy
- Configure Kinesis Firehose Agent
- Start and Validate Agent
- Conclusion - Building Simple Steaming Pipeline

Populating GitHub Data to Dynamodb

- Install required libraries
- Understanding GitHub APIs
- Setting up GitHub API Token
- Understanding GitHub Rate Limit
- Create New Repository for since
- Extracting Required Information
- Processing Data
- Grant Permissions to create dynamodb tables using boto3
- Create Dynamodb Tables
- Dynamodb CRUD Operations
- Populate Dynamodb Table
- Dynamodb Batch Operations

Overview of Amazon Athena

- Getting Started with Amazon Athena
- Quick Recap of Glue Catalog Databases and Tables

- Access Glue Catalog Databases and Tables using Athena Query Editor
- Create Database and Table using Athena
- Populate Data into Table using Athena
- Using CTAS to create tables using Athena
- Overview of Amazon Athena Architecture
- Amazon Athena Resources and relationship with Hive
- Create Partitioned Table using Athena
- Develop Query for Partitioned Column
- Insert into Partitioned Tables using Athena
- Validate Data Partitioning using Athena
- Drop Athena Tables and Delete Data Files
- Drop Partitioned Table using Athena
- Data Partitioning in Athena using CTAS

Amazon Athena using AWS CLI

- Amazon Athena using AWS CLI - Introduction
- Get help and list Athena databases using AWS CLI
- [Commands] Get help and list Athena databases using AWS CLI
- Managing Athena Workgroups using AWS CLI
- [Commands] Managing Athena Workgroups using AWS CLI
- Run Athena Queries using AWS CLI
- [Commands] Run Athena Queries using AWS CLI
- Get Athena Table Metadata using AWS CLI
- [Commands] Get Athena Table Metadata using AWS CLI
- Run Athena Queries with custom location using AWS CLI
- [Commands] Run Athena Queries with custom location
- Drop Athena table using AWS CLI
- [Commands] Drop Athena table using AWS CLI
- Run CTAS under Athena using AWS CLI
- [Commands] Run CTAS under Athena using AWS CLI

Amazon Athena using Python boto3

- Amazon Athena using Python boto3 - Introduction
- Getting Started with Managing Athena using Python boto3
- [Code] Getting Started with Managing Athena using Python boto3
- List Amazon Athena Databases using Python boto3
- [Code] List Amazon Athena Databases using Python boto3
- List Amazon Athena Tables using Python boto3
- [Code] List Amazon Athena Tables using Python boto3
- Run Amazon Athena Queries using Python boto3
- [Code] Run Amazon Athena Queries using Python boto3
- Review Athena Query Results using boto3

- [Code] Review Athena Query Results using Python boto3

Getting Started with Amazon Redshift

- Getting Started with Amazon Redshift - Introduction
- Create Redshift Cluster using Free Trial
- Connecting to Database using Redshift Query Editor
- Get list of tables querying information schema
- [Queries] - Get list of tables querying information schema
- Run Queries against Redshift Tables using Query Editor
- [Queries] - Validate users data using Query Editor
- Create Redshift Table using Primary Key
- [Queries] - Create Redshift Table
- [Consolidated Queries] - CRUD Operations
- Insert Data into Redshift Tables
- Update Data in Redshift Tables
- Delete data from Redshift tables
- Redshift Saved Queries using Query Editor
- Deleting Redshift Cluster
- Restore Redshift Cluster from Snapshot

Copy Data from s3 into Redshift Tables

- Copy Data from s3 to Redshift - Introduction
- Setup Data in s3 for Redshift Copy
- Copy Database and Table for Redshift Copy Command
- Create IAM User with full access on s3 for Redshift Copy
- Run Copy Command to copy data from s3 to Reshift Table
- Troubleshoot Errors related to Redshift Copy Command
- Run Copy Command to copy from s3 to Redshift table
- Validate using queries against Redshift Table
- Overview of Redshift Copy Command
- Create IAM Role for Redshift to access s3
- Copy Data from s3 to Redshift table using IAM Role
- Setup JSON Dataset in s3 for Redshift Copy Command
- Copy JSON Data from s3 to Redshift table using IAM Role

Develop Applications using Redshift Cluster

- Develop application using Redshift Cluster - Introduction
- Allocate Elastic Ip for Redshift Cluster
- Enable Public Accessibility for Redshift Cluster
- Update Inbound Rules in Security Group to access Redshift Cluster
- Create Database and User in Redshift Cluster

- Connect to database in Redshift using psql
- Change Owner on Redshift Tables
- Download Redshift JDBC Jar file
- Connect to Redshift Databases using IDEs such as SQL Workbench
- Setup Python Virtual Environment for Redshift
- Run Simple Query against Redshift Database Table using Python
- Truncate Redshift Table using Python
- Create IAM User to copy from s3 to Redshift Tables
- Validate Access of IAM User using Boto3
- Run Redshift Copy Command using Python

Redshift Tables with Distkeys and Sortkeys

- Redshift Tables with Distkeys and Sortkeys - Introduction
- Quick Review of Redshift Architecture
- Create multi-node Redshift Cluster
- Connect to Redshift Cluster using Query Editor
- Create Redshift Database
- Create Redshift Database User
- Create Redshift Database Schema
- Default Distribution Style of Redshift Table
- Grant Select Permissions on Catalog to Redshift Database User
- Update Search Path to query Redshift system tables
- Validate table with DISTSTYLE AUTO
- Create Cluster from Snapshot to the original state
- Overview of Node Slices in Redshift Cluster
- Overview of Distribution Styles
- Distribution Strategies for retail tables in Redshift
- Create Redshift tables with distribution style all
- Troubleshoot and Fix Load or Copy Errors
- Create Redshift Table with Distribution Style Auto
- 19 Create Redshift Tables using Distribution Style Key
- Delete Cluster with manual snapshot

Redshift Federated Queries and Spectrum

- Redshift Federated Queries and Spectrum - Introduction
- Overview of integrating RDS and Redshift for Federated Queries
- Create IAM Role for Redshift Cluster
- Setup Postgres Database Server for Redshift Federated Queries
- Create tables in Postgres Database for Redshift Federated Queries
- Reading Json Data to Dataframe using Pandas

- Write JSON Data to Database Tables using Pandas
- Create IAM Policy for Secret and associate with Redshift Role
- Create Redshift Cluster using IAM Role with permissions on secret
- Create Redshift External Schema to Postgres Database
- Update Redshift Cluster Network Settings for Federated Queries
- Performing ETL using Redshift Federated Queries
- Clean up resources added for Redshift Federated Queries
- Grant Access on Glue Data Catalog to Redshift Cluster for Spectrum
- Setup Redshift Clusters to run queries using Spectrum
- Quick Recap of Glue Catalog Database and Tables for Redshift Spectrum
- Create External Schema using Redshift Spectrum
- Run Queries using Redshift Spectrum
- Cleanup the Redshift Cluster

Projects

- **AWS Data Engineering Project1**
- **AWS Data Engineering Project2**