MASTERING MLOPS WITH CLOUD PLATFORMS: AWS, GCP, AND AZURE

Module 1. Introduction to MLOps

- What is MLOps?
- Definition and importance
- MLOps lifecycle
- Key Components of MLOps
- Data Management
- Model Development
- Model Deployment
- Monitoring and Maintenance

Module 2. Data Management

- Data Collection and Storage
- Data sources and types
- Data storage solutions (AWS S3, GCP Cloud Storage, Azure Blob Storage)
- Data Preprocessing
- Data cleaning and transformation
- Feature engineering

Module 3. Model Development

- Machine Learning Frameworks
- Introduction to TensorFlow, PyTorch, and Scikit-learn
- Model Training and Evaluation
- Training models on cloud platforms (AWS SageMaker, GCP AI Platform, Azure ML)

- Model evaluation metrics

Module 4. Model Deployment

- Containerization
- Introduction to Docker
- Creating Docker containers for ML models
- Deployment Strategies
- Deploying models on AWS, GCP, and Azure
- Using Kubernetes for scalable deployments

Module 5. Monitoring and Maintenance

- Model Monitoring
- Tools for monitoring model performance (AWS CloudWatch, GCP Stackdriver, Azure Monitor)
- Model Retraining
- Automating model retraining pipelines
- Continuous Integration/Continuous Deployment (CI/CD) for ML models

Projects: Let's Perform Projects

Project 1: MLOps Pipeline on AWS – Predicting Call Drops in Telecom Networks

Problem Definition: In the telecommunications industry, call drops are a significant issue affecting customer satisfaction and regulatory compliance. A call drop occurs when a call is unexpectedly disconnected due to network issues, poor signal strength, or system overloads.

Why This Project Is Needed:

1. Customer Satisfaction: High call drop rates lead to customer frustration and potential loss of subscribers.
2. Regulatory Compliance: Telecom operators face penalties if call drop rates exceed regulatory thresholds.
3. Operational Efficiency: Predicting call drops helps in proactive maintenance and optimal resource allocation.
4. Competitive Advantage: Improving call quality can differentiate a provider in a saturated market.
5. MLOps Significance: Demonstrates the application of machine learning and MLOps to solve industry-specific challenges in real-time environments.

How This Project Will Solve the Problem:

By developing a predictive model that analyzes network data to forecast potential call drops, telecom operators can take preemptive measures to optimize network performance, allocate resources effectively, and enhance overall service quality.

Tools and Services Needed:

- AWS S3: For scalable data storage of large volumes of network logs.
- AWS EMR (Elastic MapReduce): For processing big data using Hadoop/Spark clusters.
- AWS Glue: For ETL (Extract, Transform, Load) processes and data cataloging.
- AWS SageMaker: For building, training, and deploying machine learning models.
- AWS Lambda: For serverless computing tasks.
- AWS Kinesis: For real-time data streaming and processing.
- AWS Step Functions: For orchestrating complex workflows.

- AWS CloudWatch: For logging, monitoring, and alerting.
- AWS CodeCommit & CodePipeline: For source control and CI/CD pipelines.
- MLflow: For experiment tracking and model management.
- Python Libraries: Pandas, NumPy, Scikit-learn, PySpark, etc.

Project 2: MLOps Pipeline on GCP – E-commerce Personalized Recommendation System

Problem Definition: In the competitive e-commerce landscape, providing personalized product recommendations is essential for enhancing user experience, increasing sales, and fostering customer loyalty.

Why This Project Is Needed:

1. Customer Engagement: Personalised recommendations keep users engaged and encourage exploration.
2. Revenue Growth: Cross-selling and up-selling through recommendations boost average order value.
3. Customer Retention: Tailored experiences increase customer satisfaction and loyalty.
4. MLOps Significance: Showcases deploying a recommendation engine in a real-world, scalable environment using MLOps practices.

How This Project Will Solve the Problem:

- By developing a recommendation engine that suggests products based on user behaviour and preferences, the e-commerce platform can enhance user experience, leading to increased sales and customer retention.

Tools and Services Needed:

- Google Cloud Storage (GCS): For storing datasets and artifacts.
- BigQuery: For large-scale data querying and analysis.
- Cloud Dataflow: For data processing pipelines.
- Vertex AI: For model development, training, and deployment.
- TensorFlow Extended (TFX): For building production-grade ML pipelines.
- Google Kubernetes Engine (GKE): For containerized deployment of applications.
- Cloud Pub/Sub: For messaging and streaming data.

- Cloud Functions: For serverless computing tasks.
- Cloud Build & Cloud Source Repositories: For CI/CD pipelines.
- Cloud Monitoring & Logging: For system observability.
- Python Libraries: TensorFlow, Scikit-learn, Pandas, etc.

Project 3: MLOps Pipeline on Azure – Real-Time Network Intrusion Detection System

Problem Definition: With the increasing sophistication of cyber-attacks, organizations need robust Network Intrusion Detection Systems (NIDS) to protect their infrastructure. Traditional signature-based systems are insufficient against novel threats.

Why This Project Is Needed:

- Cybersecurity Threats: Protecting networks from unauthorized access is critical.
- Data Protection: Safeguarding sensitive information from breaches.
- Regulatory Compliance: Meeting standards like GDPR, HIPAA, and others.
- MLOps Significance: Demonstrates deploying real-time machine learning models in a security context.

How This Project Will Solve the Problem:

By developing an anomaly detection model that monitors network traffic in real-time to identify suspicious activities, organizations can respond promptly to potential threats.

Tools and Services Needed:

- Azure Event Hubs: For ingesting streaming network data.
- Azure Stream Analytics: For real-time data processing and analysis.
- Azure Data Lake Storage: For storing large volumes of network logs.
- Azure Databricks: For data processing and machine learning tasks.
- Azure Machine Learning Service: For managing the ML lifecycle.
- Azure Kubernetes Service (AKS): For deploying the model.
- Azure Functions: For serverless event-driven execution.
- Azure DevOps: For CI/CD pipelines.

- Azure Monitor and Log Analytics: For monitoring and logging.
- Python Libraries: PySpark, Scikit-learn, TensorFlow, etc.