Domain: Shipping

Core / Problem Addressed:

This project focuses on cargo capacity planning for transporting cars from the Port of Loading (Mumbai) to the Port of Departure (Bangalore). It determines how many vehicles can be loaded, calculates transportation timelines, and processes forecast data for customers such as JCB, Honda, Hyundai, BMW, and Samsung SDI. The system processes customer forecast files using two input files along with master data tables.

Key Proceses & Data Flow:

1. Data Ingestion:
   * Customers send forecast files via email.
   * Daily production data is ingested from email attachments.
   * Bulk data ingestion from SharePoint.
   * Additional ingestion via FTP/SFTP through APIs.
   * Azure Logic App checks emails every 10 seconds for attachments.
   * Attachments and email bodies are stored in Blob Storage landing zone.
   * Capable of handling 40,000–120,000 rows of data.
2. Customer Identification & Data Processing (ETL):
   * Files from the landing zone are processed by ML pipelines for ETL.
   * Python scripts read data and match headers using a mapping sheet
   * Customer and region identification based on content, columns, sender email, and subject line.
   * Data cleaning: whitespace removal, null row/column removal, naming translations, data type conversions.
   * Calculation of dimensions (width, length, weight, height), CBM, BT, BBD.
   * Data Quality Index (DQI) computation.
   * Cargo Ready Date calculated using planned delivery date and lead times.
   * Processed data stored in Couchbase.
   * Processed files moved to respective Blob Storage folders.
   * Landing zone files deleted after processing.
   * Notification triggered post-processing.
   * Transaction history stored in database.

Technologies Used:

* Azure Services: Azure Data Factory, Blob Storage, ADLS Gen2, Logic Apps, Azure ML Studio, Azure Synapse Analytics.
* Data Storage: Data Lake, Couchbase, Blob Storage (multiple containers), One Bridge database.
* Programming / Tools: Python, PySpark, SQL.
* Architecture: Medallion (Bronze, Silver, Gold).
* BI Tools: Power BI Desktop, Power Query, Power BI Service, DAX.
* Other: Git, Azure DevOps.

**Summary Points (Shipping Domain)**

 Designed and implemented an **Azure-based data ingestion and ETL pipeline** for processing large-scale customer forecast data

 Automated **file ingestion** from Email via Logic Apps, (SharePoint – Optional) into **Azure Blob Storage Landing Zone**.

 Configured **Azure Logic Apps** to check the development email inbox every 10 seconds, filter attachments (CSV/Excel only), and reject non-accepted formats.

 Stored email metadata (customer ID, file name, subject, body) in Blob Storage for tracking and audit purposes.

 Used **orchestration logic** and a **metadata mapping sheet** to identify the correct customer, region, make, and model from file headers.

 Implemented **data cleaning** – trimming whitespaces, removing null-only rows/columns, correcting naming inconsistencies, and converting data types.

 Performed **dimension calculations** (width, length, weight, height) and computed **CBM, BT, BBD** values from forecast files.

 Calculated **Cargo Ready Date** using planned delivery dates and lead times for factory-to-port movements.

 Applied **Data Quality Index (DQI)** checks to validate mandatory fields and flag incomplete or inconsistent records.

 Stored processed and validated data into **Couchbase** and moved cleaned files to **customer-specific Blob Storage folders**.

 Deleted processed files from the Landing Zone and triggered **notifications** upon successful or failed processing.

 Followed the **Medallion Architecture** to enable downstream **Power BI dashboards** for operational insights.

**Project Domain: Healthcare**

Project Overview: Engineered an end-to-end data pipeline on AWS to process and analyze doctor-patient conversations from video and audio sources. The project aimed to provide data analysts with actionable insights through a data mart and train a Large Language Model (LLM) to reproduce video content and answer user queries. The solution leveraged a modern data stack, incorporating DevOps principles, Infrastructure as Code, containerization, and a data lakehouse architecture.

Responsibilities & Achievements:

* Data Ingestion and Processing: Developed an automated data pipeline using Apache Airflow as an orchestration tool. Ingested video and transcript data from a custom YouTube API and OpenAI's Whisper API. Handled both streaming and bulk data using Kafka/Kinesis and S3 buckets.
* Data Lake and Transformation: Implemented a medallion architecture with Bronze, Silver, and Gold layers for raw, transformed, and analyzed data, respectively. Utilized PySpark notebooks on Databricks for data cleaning and transformation, including handling null values, removing whitespaces, and correcting data types. Ensured data integrity by performing normalization and utilizing a Snowflake schema to organize facts and dimensions.
* Machine Learning and AI Integration: Used Amazon SageMaker to fine-tune an LLM with the processed transcription data. The LLM was trained to reproduce video content and generate synthetic data for educational purposes. Integrated the fine-tuned model into a Django application for end-user access.
* Deployment and Infrastructure: Deployed the application using Docker and Amazon ECS. Implemented Infrastructure as Code using Terraform. Set up AWS Glue and Athena for cataloging and querying the data lake.
* Reporting and Insights: Created a data mart to provide business intelligence for data analysts, enabling them to gain insights from patient conversation data. The transformed data was served to end-users to provide these insights.

**Summary Points (Healtcare Domain)**

* Designed and developed an **AWS-based ELT pipeline** to process 100GB of video/audio data from multiple hospitals (10–12GB per hospital).
* Collected **doctor–patient conversation recordings** and converted them into text transcripts using transcription models.
* Ingested raw video, audio, and transcript data into **Amazon Kinesis** (real-time streaming) and stored it in **AWS S3** staging (Bronze layer).
* Orchestrated ETL workflows using **Apache Airflow** integrated with **Terraform** and **Docker** for reproducible deployments.
* Created a **Glue Data Catalog** for schema management and used **Athena** for querying raw and transformed datasets.
* Performed data cleaning in **Databricks (PySpark)** – datatype conversion, timestamp formatting, whitespace removal, and regex-based special character removal.
* Implemented both **incremental loads** (history + current) and **full loads** (current-only refresh) to optimize processing.
* Structured datasets in **Medallion Architecture** – Staging/Bronze → Intermediate/Silver → Data Mart/Gold for analytics.
* Fine-tuned **LLM models** on Amazon SageMaker using Hugging Face and **LangChain** for Q&A on transcript content.
* Deployed a **Django web application** on Amazon ECS (Docker containers) to provide real-time transcript search and content summarization.
* Built automated pipelines for **ML model training, deployment, and inference** within the AWS environment.
* Delivered a complete **video content analytics and AI-powered Q&A system**, enabling healthcare teams to derive insights and enhance patient engagement.

**Domain: Insurance**

Data Size: Monthly datasets for 28,000+ brokers/partners/agents, 40,000+ companies, and ~289,000 employees.

Introduction & Goal: This project implemented an AWS-based data processing, validation, and master data management (MDM) system for ACA (Affordable Care Act) compliance reporting. It automated broker–company–user mapping, ensured data quality, and streamlined the ACA form submission process for UHC through HRLogics and Mineral.

Key Processes & Data Flow:

1. Data Ingestion & Storage:
   * Monthly ACA datasets received from UHC between the 25th and 30th.
   * Ingested into AWS S3 Data Lake and cataloged with AWS Glue Crawlers.
   * Stored as Managed, External, and Delta tables for querying.
2. Data Matching & Validation:
   * FuzzyWuzzy library with cosine similarity used to handle typos and near matches (e.g., “Pune”, “Prue” → “Pune”).
   * Matching criteria: Company Name, Address, Email, Phone, URL, Zip Code, User Email.
   * Missing brokers or companies automatically created in Mineral with reference to state lists.
   * Existing entities updated; new companies created under matching brokers.
3. Data Quality Checks:
   * Validations for:
     + Invalid employee counts (null, zero, negative).
     + Incomplete client data.
     + Missing or multiple primary locations.
     + Active clients without users.
     + Multi-sponsor clients (more than one active partner in Salesforce).
     + Incorrect client names or contact details.
   * Generated Data Quality Index (DQI) for each dataset.
4. Master Data Management (MDM):
   * Linked identical client records across Salesforce and Mineral.
   * Partner–Client hierarchy accurately maintained.
   * Enabled single version of truth for billing and reporting.
5. Reporting & Compliance:
   * ACA forms (1094A, 1094B, 1094C) prepared and submitted via HRLogics.
   * Monthly analytics for broker/company/user onboarding and retirements.
   * Used Athena Serverless SQL Editor for ad-hoc queries and data quality tracking.

Technologies Used:

* AWS Services: S3, Glue, Athena, Redshift.
* Data Processing: PySpark, Python (FuzzyWuzzy, Cosine Similarity), SQL.
* Integration: Salesforce, HRLogics, Mineral.
* Other Tools: Databricks, Git, Figma (data flow design).

**Project Summary (Insurance Domain)**

* Built an AWS-based ACA compliance data pipeline for processing monthly datasets from 28K+ brokers/partners, 40K+ companies, and ~289K employees.
* Automated data ingestion from UHC into AWS S3 Data Lake between the 25th and 30th of every month.
* Used AWS Glue Crawlers to catalog ingested data and create queryable schemas for processing.
* Implemented FuzzyWuzzy with cosine similarity for broker/company name matching and typo correction (e.g., “Pune” → “Pune”).
* Applied matching criteria such as company name, address, email, phone, URL, zipcode, and user email to ensure accurate mapping.
* Designed conditional workflows to:
* Create new brokers if missing in Mineral.
* Create new companies under existing brokers.
* Update company data if already present.
* Developed Data Quality Index (DQI) checks to validate employee counts, locations, client names, contact details, and active client/user status.
* Implemented multi-sponsor detection for clients linked to multiple active partners in Salesforce.
* Maintained Master Data Management (MDM) to unify duplicate client records across Salesforce and Mineral systems.
* Processed and stored validated datasets in AWS Athena and Redshift for reporting and compliance.
* Automated preparation and submission of ACA forms (1094A, 1094B, 1094C) through integration with HRLogics.
* Delivered monthly analytical reports on broker/company/user additions, retirements, and activity trends for executive review.

**OTT Domain**

The dashboard's primary purpose is to provide a holistic, real-time view of the OTT platform's performance, focusing on key areas: user engagement, content popularity, subscription health, and partner performance. It's designed for stakeholders such as product managers, marketing teams, and executives who need to make data-driven decisions.

Purpose: To provide a quick, high-level overview of the platform's health. This section is for executives or anyone who needs to grasp the core metrics in a few seconds.

* KPI Card 1: Total Users (Users Count)
  + Purpose: The most fundamental metric. Shows the total number of registered users.
  + Value: 1.2M (with a trend indicator showing a 5% increase over the last 30 days).
* KPI Card 2: Total Content (Total Content)
  + Purpose: Measures the size of our content library.
  + Value: 50,000+ videos, movies, and series.
* KPI Card 3: Active Subscriptions (Subscription)
  + Purpose: The core revenue metric. Differentiates between total users and paying users.
  + Value: 450,000 (with a gauge showing 75% of our monthly target).
* KPI Card 4: Subscriptions Expiring This Month (Subscription Expire)
  + Purpose: Proactive churn management. Highlights the immediate risk to revenue.
  + Value: 25,000 (with a sparkline chart showing the trend over the last 90 days).
* Pie Chart: User Login Type
  + Purpose: Visualizes how users are signing up. This helps the marketing team decide where to focus their acquisition efforts (e.g., if Google logins are dominant, we might invest more in Google ads).
  + Visuals: A pie chart showing the percentage breakdown: Google (45%), Facebook (30%), Mail (20%), Mobile Number (5%).
* Pie Chart: User Subscription Plan
  + Purpose: Breaks down our revenue streams by plan type. Helps the product team understand which pricing tiers are most popular.
  + Visuals: A pie chart showing the distribution: Monthly Plan (60%), Yearly Plan (35%), Partner Plan (5%). This highlights the strong preference for monthly plans but also the significant value of yearly subscribers.
* Pie Chart: Content Type Distribution
  + Purpose: Shows the composition of our content library. This can be compared against user consumption to identify gaps (e.g., if we have lots of video content but most views are on movies, we should license more movies).
  + Visuals: A pie chart showing the mix: Video (25%), Audio (5%), Movie (35%), Show (30%), Playlist (5%).
* Numbered List/Table: Top 5 Most Requested Content
  + Purpose: Identifies content that users are actively searching for but may not be available. This is a powerful signal for content acquisition teams.
  + Visuals: A simple table showing the top 5 requested titles, along with the number of requests.
* Numbered List/Table: Top 5 Content in Watchlist
  + Purpose: A strong indicator of future engagement. This shows what users are intending to watch and can inform a/b tests for recommendation engines.
  + Visuals: A list of the top 5 titles in watchlists.
* Bar Chart: Most Favorite Promocode
  + Purpose: Directly measures the effectiveness of our promotional campaigns. We can see which codes are driving the most conversions.
  + Visuals: A bar chart comparing the top 20 vs. top 10 most used promo codes. This could be a drill-down feature.
* Table: User Details (Detailed Report)
  + Purpose: A detailed view for deeper analysis and customer support. It contains specific user data.
  + Visuals: A table with searchable columns: Apple ID, Subscription Type, Device, Email.
* Cards & Tables: Partner Performance
  + Purpose: To monitor the value our partners are bringing to the platform, either through content or user acquisition.
  + Visuals:
    - Card: Playback Request: Total playbacks.
    - Card: Playback API by Partner: Number of API calls from specific partners, indicating their content is being consumed.
    - Card: Partners: Number of active partners.
    - Table: Partners: A table with a list of partners and key metrics (e.g., number of users acquired, content views).
* **Project Summary (OTT Domain)**

 Designed and developed a comprehensive Power BI dashboard for an OTT platform, integrating data from PostgreSQL and Firebase to provide a unified view of business performance.

 Authored and implemented key DAX measures to track critical metrics, including active subscriptions, monthly subscription churn, and content playback requests.

 Created a dynamic visual data story for stakeholders, featuring an executive summary with KPI cards and detailed sections on user behavior and content trends.

 Delivered actionable insights by visualizing user acquisition channels, subscription plan adoption, and the most popular content types through intuitive pie charts.

 Enabled proactive business decisions by building a metric to identify subscriptions expiring in the current month, allowing the marketing team to target at-risk users.

 Analyzed content performance by tracking the most requested content and top watchlisted titles, providing data-backed recommendations for content acquisition strategy.

 Evaluated marketing campaign effectiveness by developing a bar chart to compare promocode usage, directly linking promotional efforts to user conversion.

 Monitored partner performance by creating dedicated metrics to track playback API requests originating from partners, assessing their value to the platform.

 Designed a user-centric dashboard layout that presented complex data in a clear, digestible format, empowering a wide range of users from executives to product managers.

 Translated raw data into strategic business intelligence, connecting platform activity with revenue health and informing future product development priorities.