**SwiftSupply Logistics – Azure Data Platform**

# 1. Introduction

SwiftSupply Logistics Pvt Ltd (Fictional Company) operates a retail & e-commerce supply chain with 5 warehouses across India (Mumbai, Delhi, Bangalore, Chennai, Hyderabad). The company faced challenges such as stock shortages due to poor demand forecasting, late shipments causing customer complaints, and fragmented data across Excel, SQL Server, and IoT devices.  
  
This project builds a modern Azure Data Platform that ingests, processes, and curates both batch and streaming data, enabling real-time supply chain insights through Power BI and Synapse Analytics.

# 2. Architecture

The architecture includes data ingestion from AWS S3 (batch) and IoT devices (streaming), storage in ADLS, processing using Databricks, orchestration with ADF, Data Ingestion using Synapse Pipeline, Data warehousing in Synapse Dedicated SQL Pool and visualization in Power BI.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

# 3. Storage Design (ADLS Containers)

- **archive/**

  - batch/{yyyyMMdd}/ (csv files)

  - streaming/{yyyyMMdd}/ (json files)

- **bronze/**

  - batch/

    - filemapping/fileMapping.json  (used by lookup task)

    - inventory/ (csv file)

    - salesorder/ (csv file)

- streaming/

    - landing\_shipments/ (json files)

    - shipments/ (delta files)

    - shipments\_autoloader\_schema/

    - shipments\_checkpoints/

- **silver/**

  - batch/

    - inventory/ (delta files)

    - salesorder/ (delts files)

  - streaming/

    - shipments/ (delta files)

    - shipments/checkpoints

- **gold/**

  - tables/ (7 delta tables managed by Databricks SQL)

- dim\_product\_parquet/ (parquet file from delta table)

- dim\_warehouse\_parquet/ (parquet file from delta table)

- fact\_inventory\_daily\_parquet / (parquet file from delta table)

- fact\_sales\_daily\_parquet / (parquet file from delta table)

- fact\_shipments\_parquet/ (parquet file from delta table)

- supplychain\_snapshot\_parquet/ (parquet file from delta table)

- alerts\_shipments\_parquet/ (parquet file from delta table)

- \_success/ (acknowledgement txt file for 7 parquet files)

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# 4. Data Sources

## Batch (Historical CSVs – from S3)

**1. Inventory**  
Columns: warehouse\_id, product\_id, product\_name, category, quantity\_in\_stock, last\_restock\_date  
Example:  
WH1, PRD001, Laptop Bag, Accessories, 120, 2025-07-15

**2. Sales Orders**  
Columns: order\_id, product\_id, quantity\_sold, sale\_date, warehouse\_id  
Example:  
ORD00001, PRD001, 5, 7/10/2025, WH1

## Streaming (IoT JSON – from Simulator)

{  
 "shipment\_id": "SH92384",  
 "warehouse\_id": "WH1",  
 "status": "In Transit",  
 "location": "Mumbai",  
 "timestamp": "2025-08-12T10:25:00Z"  
}

# 5. Pipelines & Processing

## A. Batch Pipeline (ADF)

- Triggered daily.  
- Monitors S3 bucket for inventory\_<date>.csv and salesorder\_<date>.csv.  
- Waits up to 2 hours for both files.  
- If files exist → copied to bronze/batch/.  
- If missing/invalid → pipeline times out & sends alert.  
- After ingestion → runs Databricks notebooks for processing.  
- Archives batch files to archive/batch/{date}/ and streaming files to archive/streaming/{date}

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## B. Streaming Pipeline (Databricks)

- Python script generates shipment JSON events → stored in bronze/streaming/landing\_shipments/.  
- Autoloader Notebook 1: Reads JSONs, cleans, writes Delta to bronze/streaming/shipments.  
- Autoloader Notebook 2: Reads Bronze shipments, transforms, writes Delta to silver/streaming/shipments.

Scheduled as Databricks Jobs and runs 24/7.

## C. Transformation & Gold Tables

Notebook 1 – Batch Transform:  
- Read CSVs (inventory and salesorder), cleans: data cleaning, schema enforcement, deduplication.  
- Saves as Silver Delta to silver/streaming/inventory and silver/streaming/salesorder.  
  
Notebook 2 – Gold Tables Creation:  
- Joins batch & streaming Silver data.  
- Creates 7 curated Delta tables.

* dim\_product
* dim\_warehouse
* fact\_inventory\_daily
* fact\_sales\_daily
* fact\_shipments
* alert\_shipments
* supply\_chain\_snapshorts

- Additionally, writes each gold table as a Parquet file into gold/parquet/ in ADLS.

- After successful write of all 7 Parquet files, a \_success.txt marker is dropped in gold/\_success/.

**D. Synapse Analytics Integration**

- A Dedicated SQL Pool is created in Synapse Analytics.

- Stored procedure implemented to use **COPY INTO** command to load the 7 Parquet files from ADLS (gold/parquet/) into SQL Pool tables under **gold schema**.

- This stored procedure is triggered via a Synapse pipeline activity whenever the \_success.txt marker arrives in ADLS.

- Event trigger: Storage Event Trigger configured in Synapse pipeline monitors gold/\_success/ path.

- Once triggered, Synapse loads the curated Parquet files into the data warehouse, ensuring BI reporting tables are always updated.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# 6. Business Rules

1. Low Stock Rule: If quantity\_in\_stock < 50 → mark as Low Stock.  
2. Shipment Delay Alerts: If status = 'Delayed' → insert into alerts\_shipments.  
3. Retention Rule: Keep only last 90 days of shipment data in curated zone.

# 7. Security

- Azure Key Vault stores ADLS credentials.  
- ADLS RBAC: Bronze → ingestion team, Silver → data engineers, Gold → BI team.

- Synapse access controlled via Azure AD groups for analysts and BI users.

A screenshot of a computer

AI-generated content may be incorrect.

# 8. Monitoring

- ADF: Failure or timeout → alert notification.  
- Databricks: Streaming jobs monitored via UI & logs.

- Synapse: Pipeline execution logs and SQL monitoring track COPY INTO jobs.

# 9. Power BI Dashboard

The Gold tables are consumed in Power BI for real-time supply chain reporting.

# 10. Repository Structure

swift-supply-azure-data-platform/  
│── README.md  
│── docs/  
│ ├── SwiftSupply\_Azure\_Data\_Platform.docx  
│ ├── architecture\_diagram.png  
│ └── PowerBI\_dashboard.png  
│── data/  
│ ├── inventory\_sample.csv  
│ ├── salesorder\_sample.csv  
│ └── shipments\_sample.json  
│── notebooks/  
│ ├── SSL-batch-bronze-to-silver.py  
│ ├── SSL-silver-gold.py  
│ ├── SSL-streaming-bronze-silver.py  
│ └── SSL-streaming-source-bronze.py  
│── pipelines/  
│ ├── PL\_SwiftSupply\_DataIngestion.json   
│── scripts/  
│ └── Source\_files\_generator.py  
│── configs/  
│ ├── keyvault\_config.json  
│ └── adls\_paths.md  
└── powerbi/  
 └── supplychain\_dashboard.pbix

# 11. Setup Guide

1. Deploy ADLS Gen2 and create containers.  
2. Configure Key Vault with credentials.  
3. Deploy ADF pipeline from JSON and create linkedservices.  
4. Upload sample files to S3.  
5. Run shipment simulator.  
6. Trigger ADF pipeline.  
7. Verify Gold tables in Databricks.

8. Confirm Parquet files and \_success marker in ADLS Gold.

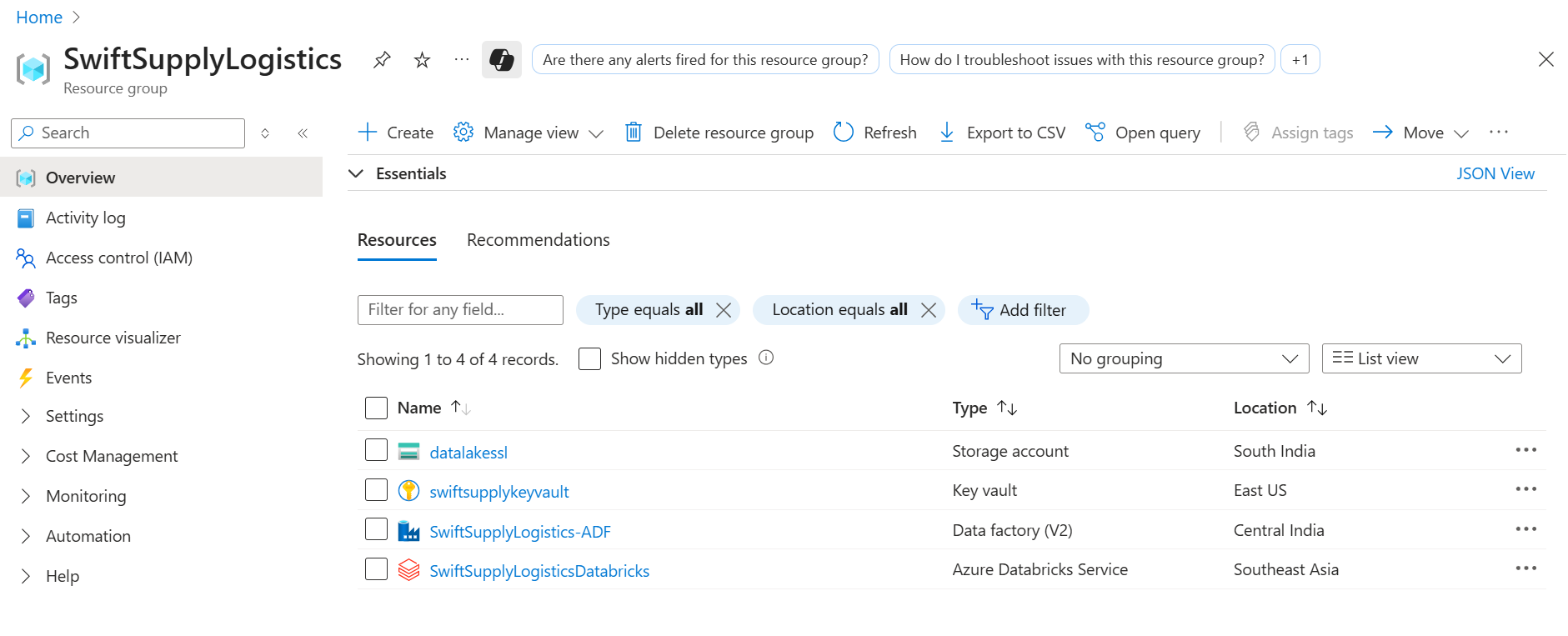
9. Verify Synapse stored procedure loads data into SQL Pool.

10. Connect Power BI to synapse.

# 12. Next Enhancements

- ML demand forecasting with Azure ML.  
- CI/CD pipelines for ADF, Databricks and Synapse.  
- Data Quality Monitoring with Delta Live Tables.

# 13. ScreenShots



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.