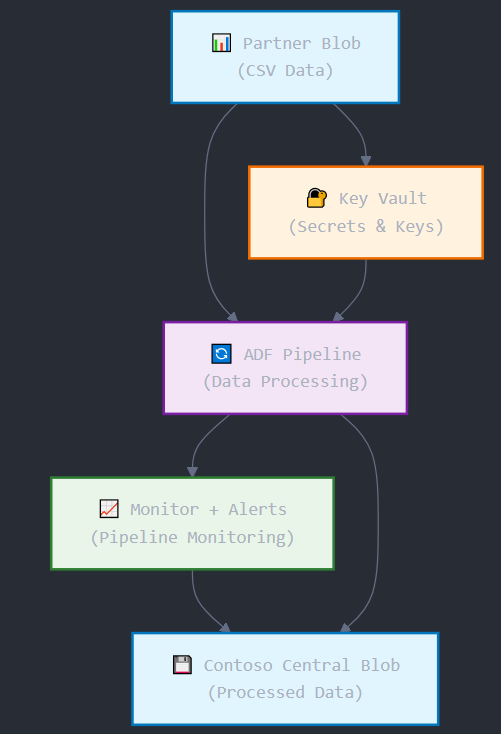
**Case Study Title: Secure and Scalable CSV-to-Blob Copy Pipeline with Key Vault Integration and SLA Monitoring**

**1. Business Scenario**

**Contoso Ltd.** is a retail company that processes daily sales data in CSV format from its partners. This data needs to be copied securely from a partner’s storage account to the company’s centralized **Blob Storage** for further analysis. To ensure **scalability, security, and operational reliability**, Contoso wants to:

* Use **parameterized pipelines** to dynamically process different source files and destinations
* Secure credentials using **Azure Key Vault**
* Set **SLA-based alerting** in case of failure or slow performance

**2. Architecture Overview**



**3. Key Features Implemented**

| **Feature** | **Description** |
| --- | --- |
| Parameterized Pipeline | Allows flexibility in file name, folder path, and destination |
| Azure Key Vault Integration | Credentials (connection strings or keys) stored securely |
| SLA Monitoring | Alert if pipeline exceeds 5 minutes or fails |
| Retry and Timeout Configuration | Robust fault-tolerant execution |
| Reusability | Datasets and Linked Services designed for reuse |

**4. Implementation Steps**

**Step 1: Setup Azure Resources**

* **Create Resource Group**: contoso-adf-rg
* **Create Storage Accounts**:
  + contosopartnersa (source)
  + contosocentralsa (sink)
* **Create Key Vault**:
  + Store secrets:
    - source-sa-key
    - sink-sa-key

**Step 2: Create Linked Services with Key Vault Authentication**

**a. Add Key Vault Linked Service**

1. ADF Studio → Manage → **Linked Services** → New
2. Select **Azure Key Vault**
3. Use system-assigned managed identity
4. Test connection and save as kv-contoso

**b. Source Linked Service (Blob via KeyVault)**

1. Add new linked service → Azure Blob Storage
2. Authentication: Account key (via Key Vault)
3. Enter:
   * Key Vault: kv-contoso
   * Secret name: source-sa-key
4. Save as ls-source-blob-kv

**c. Sink Linked Service (Blob via KeyVault)**

Repeat the same process using sink-sa-key  
Save as ls-sink-blob-kv

**Step 3: Create Parameterized Datasets**

**a. Source Dataset (CSV)**

1. Author → Dataset → Azure Blob Storage → CSV
2. Linked Service: ls-source-blob-kv
3. Set parameters:
   * folderPath (String)
   * fileName (String)
4. In Connection tab:
   * Container: select dynamically
   * Directory: @dataset().folderPath
   * Filename: @dataset().fileName
5. Save as ds-source-csv

**b. Sink Dataset (CSV)**

Same as source but using ls-sink-blob-kv  
Save as ds-sink-csv

**Step 4: Create Parameterized Pipeline**

1. Author → Pipeline → New
2. Name: csv-parameterized-copy-pipeline

**a. Add Parameters to Pipeline:**

* param\_source\_container
* param\_source\_path
* param\_source\_file
* param\_sink\_container
* param\_sink\_path
* param\_sink\_file

**b. Add Copy Activity:**

* **Source**:
  + Dataset: ds-source-csv
  + Set dataset parameters:
    - folderPath: @pipeline().parameters.param\_source\_path
    - fileName: @pipeline().parameters.param\_source\_file
* **Sink**:
  + Dataset: ds-sink-csv
  + Set parameters similarly

**c. Configure Copy Settings:**

* Enable fault-tolerance → skip incompatible rows
* Retry: 3
* Timeout: 10 minutes

**Step 5: Add SLA Monitoring with Alert**

**a. Enable Diagnostics Logs**

1. Go to ADF in Azure Portal
2. Enable **Diagnostics Settings** → send to **Log Analytics**

**b. Create Azure Monitor Alert Rule**

1. Azure Portal → Monitor → Alerts → New Alert
2. Resource: ADF
3. Condition:
   * Metric: **Activity duration**
   * Aggregation: > 5 minutes
4. Action Group: Email or Teams
5. Alert name: ADF-SLA-Breach

**c. (Optional) Alert on Failure:**

* Condition: Activity Failed > 0
* Action: Same as above

**Step 6: Test Execution**

1. Trigger pipeline manually
2. Pass following parameters:

{

"param\_source\_container": "partner-data",

"param\_source\_path": "sales/2025/August/",

"param\_source\_file": "sales\_2025\_08\_01.csv",

"param\_sink\_container": "processed-data",

"param\_sink\_path": "sales/archive/",

"param\_sink\_file": "sales\_2025\_08\_01.csv"

}

1. Observe run in **Monitor Hub**

**Step 7: Cost View and Optimization Tips**

1. Go to Azure → **Cost Management**
2. Filter by ADF resource
3. Observe costs based on:
   * Pipeline run frequency
   * Data movement (GB)
4. Cost Optimization:
   * Schedule during off-peak
   * Use staging copy only if required
   * Tune parallelism for large files

**Outcome**

* Flexible data ingestion across various partners and file structures
* Secrets fully secured using Key Vault
* SLA-driven alerting ensures proactive monitoring
* Cost visibility allows tuning pipeline frequency and performance

**Extensions / Next Steps**

* Use **Metadata-driven pipelines** with control tables
* Log pipeline metadata (file name, row count) to Azure SQL
* Integrate with **Power BI dashboards** for operational observability
* Add **data validation** with Great Expectations before copying