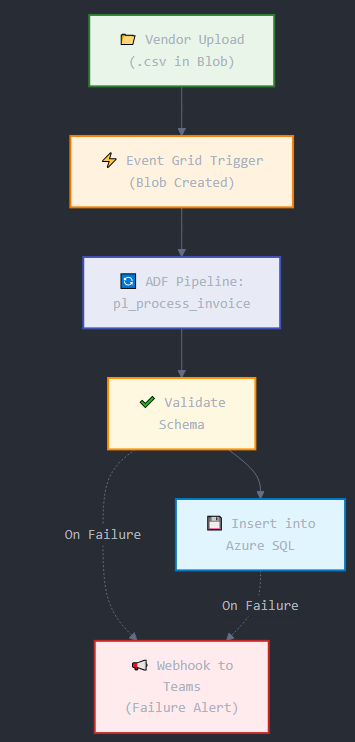
**Case Study: Real-Time Invoice Processing from Azure Blob with Retry and Teams Notification**

**Background**

A company receives invoices from vendors in real-time via Azure Blob Storage. Each file needs to be validated and uploaded to a SQL database. The system should be:

* **Event-driven**: Triggered by blob upload.
* **Resilient**: Retry on transient failures (e.g., network, database).
* **Transparent**: Notify IT support via Teams if a failure persists after retries.
* **Compliant**: All pipeline executions must be traceable.

**Architecture Overview**



**Objectives**

1. **Trigger pipeline using Event Grid** on blob upload.
2. **Validate file schema and structure**.
3. **Ingest data into SQL** only if validation succeeds.
4. **Retry with exponential back-off** on transient failure.
5. **Notify Teams** if failure persists.

**Step-by-Step Implementation**

**1. Set up Event Grid Trigger in ADF**

**a. Create an ADF pipeline pl\_process\_invoice.**

**b. Create an Event Grid subscription:**

* Go to the Storage Account > Events > + Event Subscription.
* Event Type: **Blob Created**
* Endpoint Type: **Azure Data Factory**
* Select pl\_process\_invoice.

This will make the pipeline **event-driven**, reacting to new blob uploads.

**2. Pipeline Activities**

**a. Get Metadata Activity (GetFileMetadata)**

* Input: Blob Path (from trigger parameters)
* Purpose: Ensure file exists and retrieve metadata.

**b. Validation Script (ValidateSchema)**

* Use **Azure Function** or **Stored Procedure** for checking:
  + Columns
  + Headers
  + Data types

**c. Copy Activity (CopyToSQL)**

* Source: Azure Blob
* Sink: Azure SQL Table Invoices

**3. Configure Retry and Back-off on Copy Activity**

In CopyToSQL activity:

* Retry: 3
* Retry Interval: 30 seconds
* Auto Back-off: Enabled by default

This ensures:

* Retry on transient failures like connection loss, timeout, throttling.
* Each retry increases wait time exponentially.

**4. Add Web Activity for Teams Notification (on failure)**

**a. Go to Microsoft Teams:**

* Add connector to channel > **Incoming Webhook**
* Name: ADF Alerts
* Copy the generated **Webhook URL**

**b. In ADF Pipeline:**

* After CopyToSQL, add **Failure Path**
* Insert **Web Activity** called SendTeamsAlert
* Configuration:
  + URL: Teams Webhook
  + Method: POST
  + Body:

json

{

"text": "🚨 \*\*ADF Pipeline Failure\*\*\nPipeline: @pipeline().Pipeline\nStatus: Failed\nFile: @triggerBody().fileName\nTime: @utcNow()"

}

**5. Monitor and Validate**

* Upload an invalid invoice file and observe:
  + Triggered automatically via Event Grid
  + Retry 3 times on CopyToSQL
  + Failure triggers Teams alert
* Upload a valid file:
  + Pipeline succeeds
  + No Teams alert

**Error Handling Strategy**

| **Component** | **Strategy** |
| --- | --- |
| Validation | Custom error messages, fail-fast |
| Copy Activity | 3 retries with back-off |
| Alerting | Triggered only on final failure |
| Logging | Diagnostic logs + Log Analytics |

**KQL Query for SLA / Alert Tracking (Optional)**

kusto

ADFPipelineRun

| where PipelineName == "pl\_process\_invoice"

| project PipelineName, Status, RunStart, RunEnd, DurationMs, FailureMessage

**Benefits of This Design**

| **Benefit** | **Explanation** |
| --- | --- |
| Event-Driven | Eliminates polling, reacts to data instantly |
| Resilient | Recovers from transient errors automatically |
| Alerting | Teams notifies humans only when needed |
| Scalable | Blob and SQL can auto-scale with demand |
| Auditable | Every run traceable through logs/KQL |

**Possible Enhancements**

* Store failed files in a **quarantine container** for reprocessing.
* Add **approval workflow** in Teams using Power Automate.
* Maintain a **Run History dashboard** in Power BI.
* Use **Databricks notebook** for advanced validation logic.

**Conclusion**

This event-driven pipeline with back-off retries and Teams notification provides a reliable and transparent architecture for real-time data ingestion. It ensures that transient errors are gracefully handled while maintaining operational visibility through alerts and monitoring.