ADF for ETL/ELT

As a Data Engineer working within a data team, making data available to various business units or internal users involves more than simply moving data — it requires a solid understanding of the tools, processes, and requirements involved in ETL (Extract, Transform, Load) and ELT (Extract, Load, Transform).

Reading documentation and watching tutorials can help, but the real understanding often comes through practical, hands-on experience with tools like Azure Data Factory (ADF). This guide is based on real-life experience and is intended to help data professionals, especially beginners, understand the step-by-step process of building ETL/ELT pipelines using ADF.

Understanding ETL vs. ELT

Both ETL and ELT involve moving and transforming data between systems, but the order of operations differs:

- **ETL (Extract → Transform → Load)**: Data is first extracted from the source, then transformed (cleaned, aggregated, etc.), and finally loaded into the destination.
- **ELT (Extract → Load → Transform)**: Data is extracted and immediately loaded into the destination system (often a powerful data warehouse), and the transformation happens after loading.

The choice between ETL and ELT depends on factors such as data volume, processing power, transformation complexity, and destination system capabilities.

Why Do We Use a Data Pipeline?

A data pipeline is like a transportation system. Think of it as a structured series of actions that move data from its source to its destination with the necessary transformations along the way.

Analogy: Imagine traveling from Lagos to Abuja. Lagos is your source, Abuja is your destination. The road or flight is your pipeline, the bus or plane is your ETL tool (ADF), and your travel experience tickets, checkpoints, seating — is the transformation process.

Building a pipeline helps automate, schedule, monitor, and secure the data movement process efficiently.

Why Use Azure Data Factory (ADF)?

ADF is a **cloud-based ETL service** that enables you to build data pipelines without writing extensive code. Some key benefits of ADF include:

- Drag-and-drop pipeline creation (low code/no code)
- Support for on-premises and cloud data sources
- Built-in data transformation tools (Data Flows)
- Integration with Azure Monitoring, Triggers, and Alerts
- Supports incremental loads, parameterization, and scheduling
- Scalable, cost-effective compute options (Pay-as-you-go)

Prerequisites for Using ADF

To get started with ADF, ensure the following:

1. **Azure Subscription** — A valid Azure account with billing enabled.
2. **Resource Group & Data Factory Instance** — For managing and organizing services.
3. **Integration Runtime** — For connecting to on-premise databases (install **Self-hosted Integration Runtime**).
4. **Source System** — On-premises SQL Server with sales or operational data.
5. **Sink/Destination** — PostgreSQL database (on-prem or cloud) as your data warehouse

   .

Step-by-Step: Building an ETL Pipeline with ADF

Let's assume you want to move daily sales data from an on-premises SQL Server to a PostgreSQL data warehouse. Here's how to do it:

Step 1: Create a Data Factory Instance

1. Go to the Azure Portal
2. Search for "Data Factory" and click "Create".
3. Choose your subscription, resource group, region, and give it a name.
4. Click "Review + Create", then "Create".

Step 2: Set Up Linked Services

Linked Services store connection details to your source and destination.

- For Source (SQL Server):
- Install and configure Self-hosted Integration Runtime (SHIR).
- Create a new Linked Service to connect to your on-prem SQL Server (enter server name, authentication method, database name).
- For Sink (PostgreSQL):
- Create another Linked Service to connect to your PostgreSQL database.
- Ensure the necessary network/firewall settings allow ADF to reach the database

Step 3: Create Datasets

Datasets represent your source and destination data structures.

- Source Dataset: Configure it to reference your SQL Server table or view.
- Sink Dataset: Configure it to point to your PostgreSQL table.

Step 4: Create a Pipeline

1. Navigate to Author in ADF Studio.
2. Create a new pipeline.
3. Add a Copy Activity.
4. Source: Select the source dataset.
5. Sink: Select the destination dataset.
6. Optionally, apply column mapping and transformation rules.
7. Add parameters, variables and any other activity if needed for dynamic behavior.

Step 5: Configure Triggers

1. Set up a Schedule Trigger to run the pipeline daily.
2. Configure the time and recurrence (e.g., every day at 2:00 AM).
3. Publish the trigger and activate it.

Step 6: Monitor Pipeline Runs

Go to the Monitor tab in ADF Studio to:

- Track pipeline execution status
- Check for errors
- View run history
- Debug failed runs