DATA ENGINEERING AND ANALYTICS

Building ETL Pipelines with Azure Data Factory

Level: Practitioner • 2 days (expandable to 3) • Virtual, In-person

Overview

Moving data reliably is the unglamorous heart of data engineering. Anyone can copy a table once; the hard problems are the ones that show up in production: sources that change shape, loads that must be incremental, failures at 2 a.m. that need to rerun safely, and pipelines that nobody can maintain because they were built by clicking until it worked. Azure Data Factory is Microsoft's answer for orchestrating this work, and using it well means understanding its moving parts, not just its designer canvas.

This is a hands-on, practitioner course. It builds one real pipeline pattern at a time: first the core building blocks of Data Factory, then reliable ingestion, then transformation, then the orchestration, parameterization, and monitoring that turn individual pipelines into a dependable system. Following a less-but-deeper philosophy, we focus on the patterns that cover most production work rather than touring every connector and activity type. Every module ends with a lab, and each module builds on the one before.

Who Should Attend

- Data engineers building ingestion and transformation pipelines on Azure
- Database and BI developers moving from SSIS or hand-rolled scripts to cloud-native orchestration
- Developers and architects who need to integrate data movement into a broader Azure solution

Learners new to data concepts and the Azure data landscape should take *Microsoft Azure Data Fundamentals (DP-900)* first.

Prerequisites

- Working SQL knowledge: queries, joins, and basic DDL
- Familiarity with core data concepts (tables, files, batch loads)
- Basic Azure experience: navigating the portal and creating resources

What You Will Learn

- Explain Data Factory's core components: pipelines, activities, linked services, datasets, and integration runtimes
- Build reliable ingestion with the Copy activity across files, databases, and APIs
- Transform data with mapping data flows, and judge when to push transformation to external compute instead
- Design parameterized, metadata-driven pipelines instead of one pipeline per table
- Implement incremental loads, scheduling, and dependency-aware orchestration
- Monitor, troubleshoot, and rerun pipelines safely, and deploy them through source control

Course Outline

Day one: the building blocks and moving data

- How Data Factory Works
  - Pipelines, activities, linked services, datasets, and triggers: what each is for
  - Integration runtimes: where the work actually runs, including on-premises sources
  - ETL versus ELT, and where Data Factory sits in an Azure data architecture
  - Lab: provision a Data Factory and build your first working pipeline
- Ingestion with the Copy Activity
  - Connecting to common sources: SQL databases, file storage, and REST APIs
  - File formats, schema mapping, and handling messy source data
  - Fault tolerance: retries, skipping bad rows, and logging what was skipped
  - Lab: ingest data from a database and a file source into Azure Data Lake Storage
- Transforming Data
  - Mapping data flows: joins, derived columns, aggregations, and sinks
  - When to transform in Data Factory and when to hand off to SQL or Databricks
  - Debugging data flows and reasoning about their compute cost
  - Lab: build a data flow that cleans, joins, and reshapes the ingested data

Day two: orchestration, reliability, and production

- Control Flow and Parameterization
  - Control flow activities: conditions, loops, and executing pipelines from pipelines
  - Parameters, variables, and expressions
  - Metadata-driven pipelines: one pattern that loads many tables
  - Lab: convert a single-table pipeline into a parameterized, metadata-driven one
- Incremental Loads and Scheduling
  - Watermarks and change detection: loading only what is new
  - Schedule, tumbling window, and event-based triggers, and how to choose
  - Designing for reruns: idempotent loads and safe failure recovery
  - Lab: implement an incremental load with a watermark and a scheduled trigger
- Running Pipelines in Production
  - Monitoring runs, diagnosing failures, and alerting
  - Source control integration and promoting pipelines between environments
  - Security: managed identities, Key Vault, and least-privilege access to data
  - Lab: break a pipeline on purpose, then diagnose, fix, and rerun it cleanly

Extended Version

The three-day version keeps the same gradient and adds depth where production teams need it most:

- Deeper mapping data flow patterns, including slowly changing dimensions
- Orchestrating external compute: Databricks notebooks and stored procedures within pipelines
- CI/CD for Data Factory: automated deployment across dev, test, and production

- A capstone that builds a complete, parameterized, incrementally loading pipeline system from raw sources to a serving layer