CLOUD & INFRASTRUCTURE

Data Engineering on Azure

Building analytics platforms with Data Factory, Synapse and Fabric

Houssam Kodad

PDF · DATAFORGE BOOKS

Questions about this sample or the full edition: **support@dataforgebooks.com**

CONTENTS

Table of Contents

CHAPTER 01

The Azure Data Landscape

Microsoft's data stack is powerful, sprawling, and moves quickly — and the arrival of Fabric reshuffled how the pieces fit together. The hardest part of building on Azure is often not mastering any single service but choosing intelligently among the several that appear to do the same job. This book is a practical guide to making those choices and assembling them into a platform that is reliable, secure and affordable.

This chapter is the map for everything that follows. We survey the Azure data landscape and the deliberate overlaps within it, untangle the genuine confusion between Synapse, Databricks and Fabric, introduce the lakehouse foundation on Data Lake Storage, and sketch the reference architecture the book builds toward. The goal is to give the hands-on chapters a frame to hang on.

Throughout, the emphasis is on judgement over feature tours. Azure will sell you a dozen ways to ingest data and several ways to query it; what you need is a principled basis for picking, sized to your team's skills, your governance requirements, and your budget. We start building that judgement here, with the big picture, before descending into specifics.

The Azure Data Landscape

Azure offers many ways to store, move and transform data, and the overlap is deliberate rather than accidental. Data Lake Storage Gen2 is the foundation for files. Data Factory orchestrates movement. Synapse provides SQL and Spark over the lake. Databricks offers a best-in-class Spark and lakehouse experience. Fabric bundles much of this into a single software-as-a-service surface. Each is an entry point for a different kind of team.

Seeing the landscape as a set of entry points rather than a pile of competing products is the key mental shift. The skill is matching the tool to your constraints — existing skills, governance needs, billing model, appetite for managing infrastructure — instead of collecting services because they exist. This chapter gives you the vocabulary to make those matches deliberately, and the rest of the book gives you the depth.

Synapse vs Databricks vs Fabric

These three cause more confusion than anything else in the Azure data world, so let us be direct. Synapse Analytics integrates SQL pools, Spark and pipelines in one workspace, tightly bound to the Azure ecosystem. Databricks is the lakehouse platform built around Spark and Delta Lake, strongest

where heavy data engineering meets machine learning and portability matters. Fabric is Microsoft's newer SaaS bet, unifying engineering, warehousing and BI on top of a single store called OneLake.

There is no universal winner, and anyone who tells you otherwise is selling something. Fabric simplifies operations and billing at the cost of some control and maturity. Databricks gives power and portability at the cost of more to manage. Synapse sits in between for teams already deep in Azure. We are explicit throughout about which workload each suits, so you can choose with eyes open rather than by marketing.

Storage Foundations on ADLS Gen2

Everything starts with storage, and on Azure that means Data Lake Storage Gen2 — object storage with a hierarchical namespace that makes folder operations efficient and permissions manageable. How you lay out this storage determines, more than any other early decision, whether your platform stays fast and organised or degrades into an unnavigable swamp of files.

The pattern we adopt is the medallion architecture: raw data lands in a bronze layer exactly as it arrived, is cleaned and conformed into silver, and is served as business-ready gold tables. Storing these as an open table format like Delta on top of the lake gives you transactions, time travel and schema enforcement directly on cheap storage — the lakehouse promise. We design this foundation carefully because everything above it inherits its strengths and its flaws.

```
# Medallion layout on ADLS Gen2 — one container, clear zones.
abfss://lake@account.dfs.core.windows.net/
├── bronze/   # raw, immutable, exactly as ingested
│   └── shop/orders/ingest_date=2026-06-01/
├── silver/   # cleaned, conformed, deduplicated (Delta)
│   └── orders/
└── gold/     # business-ready facts and dimensions (Delta)
    └── fct_orders/
```

Ingestion with Data Factory

Azure Data Factory is the workhorse for getting data into the lake. It connects to a large catalogue of sources, copies data at scale, and orchestrates multi-step pipelines with triggers, parameters and dependencies. For most teams it is the default ingestion and orchestration tool, and learning to use it well — especially its parameterisation — is the difference between a handful of brittle pipelines and a reusable framework.

The trap newcomers fall into is building one bespoke pipeline per source, which becomes unmaintainable at the tenth source and unthinkable at the hundredth. The professional approach uses metadata-driven, parameterised pipelines that ingest many sources from a configuration table. We build toward that pattern, because it is what lets a small team onboard new data without writing new pipelines each time.

Transforming with Synapse and Spark

Once raw data is in the lake, it must be transformed into the silver and gold layers, and Azure gives you two main engines. Synapse serverless SQL lets you query files in place with familiar SQL, ideal for exploration and lighter transformation without provisioning anything. Spark pools, in Synapse or Databricks, handle heavy, distributed transformation and machine-learning workloads that SQL cannot express comfortably.

Choosing between them is a recurring decision rather than a one-time pick. SQL is more accessible and often cheaper for set-based work; Spark is more powerful for complex, programmatic transformation and large-scale processing. Many mature platforms use both, SQL where it fits and Spark where it is needed, and we show how to combine them without the result becoming an incoherent mess of half-overlapping tools.

Security and Governance

A data platform is only as trustworthy as its security, and on Azure that rests on Entra ID for identity and role-based access control for permissions. Getting this right means least-privilege access, private networking so data never traverses the public internet unnecessarily, and a clear model of who can see what. These are not optional extras for a platform that will hold real business or personal data.

Governance goes beyond access control to cataloguing and lineage — knowing what data exists, where it came from, and who is responsible for it. Microsoft Purview provides this across the estate, and it matters more every year as data-protection regulation tightens. We treat security and governance as load-bearing parts of the architecture, designed in from the start rather than retrofitted after an audit demands them.

Cost, Operations and the Road Ahead

Cloud data platforms have a way of becoming expensive quietly, and Azure is no exception. The levers — pausing idle compute, choosing the right storage tiers, sizing Fabric capacity, tagging resources so spend is attributable — are straightforward individually but easy to neglect collectively.

We build cost-awareness into the architecture so that the platform stays affordable as it scales, rather than triggering an annual panic.

Operationally, the platform needs monitoring, alerting and a deployment process that promotes changes through environments safely. The chapters ahead take each component introduced here — storage, ingestion, transformation, security, cost — and build it out in production-grade detail, until you have a complete, operable Azure data platform you understand from the storage account up.

This is a free sample

You've reached the end of the sample chapter.

Get the complete book — every chapter, fully worked — at **dataforgebooks.com**.

FULL EDITION · 296 PAGES · PDF

Read the full title at **dataforgebooks.com**

Questions? support@dataforgebooks.com