SPEC-1-Databricks Migration & Greenfield Mastery Plan

Background

You want **end-to-end mastery of Databricks** with strong **hands-on focus**, specifically covering **Azure, AWS, and GCP Databricks**, and positioning yourself as a **hands-on architect + delivery architect**.

This means you must be able to: - Design cloud-agnostic architectures - Implement cloud-specific networking & security - Lead migrations and greenfield builds - Own delivery, costing, and governance

Time commitment: **4–6 hours/day (weekdays)** and **~4 hours on weekends**, with a target completion of **Jan 26**.

Requirements (MoSCoW)

Must Have

- Understand **migration lifecycle** (source → Databricks)
- Ability to **size & price projects**
- Perform **greenfield Databricks setup**
- Choose **right ingestion & processing patterns**
- Estimate **pipeline run cost**
- Implement **data security & governance**
- Design **Unity Catalog, schemas & databases**

Should Have

- Hands-on with **Lakehouse, Delta, CDC**
- Hands-on with **streaming + batch pipelines**

Could Have

- Exposure to **proposal templates & effort estimation models**

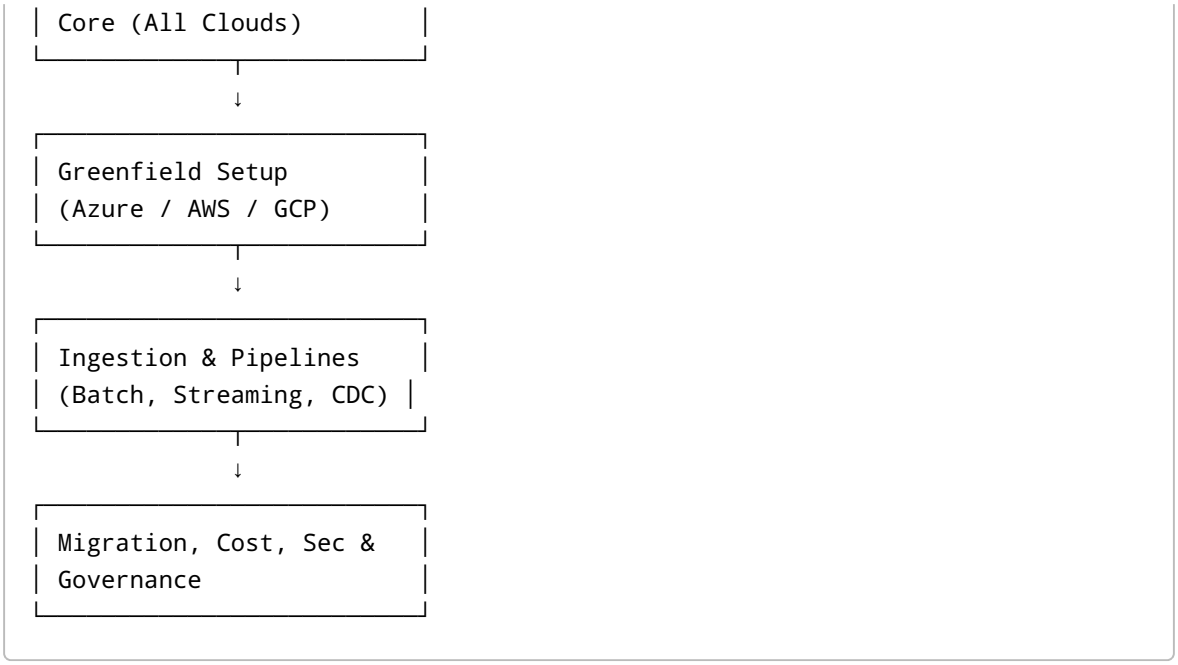
Won’t Have (for now)

- Deep ML / AI workloads

Method (Learning Architecture)

The learning approach is **cloud-agnostic first**, then **cloud-specific execution**.

```
┌──────────────────────────────┐
│ Lakehouse & Databricks       │
```



You will **build once conceptually**, then **execute on each cloud** where differences matter (networking, IAM, storage).

Implementation Plan (Day-by-Day)

Phase 1: Databricks Core, Spark & Lakehouse (Days 1–3)

Goal: Build strong Spark + Lakehouse fundamentals with SQL-to-PySpark mapping

---

Day 1 – Spark Fundamentals (SQL → PySpark)

Focus: Thinking like Spark, not just SQL

Concepts - Spark architecture (Driver, Executors) - Lazy evaluation - DataFrames vs SQL tables

Hands-on (MANDATORY) - Read CSV → DataFrame - Apply transformations (select, filter, join) - Write output as Delta

SQL → PySpark Mapping | SQL | PySpark | |----|--------| | SELECT | df.select() | | WHERE | df.filter() | | JOIN | df.join() | | GROUP BY | df.groupBy().agg() |

Deliverable - Notebook showing same logic in SQL and PySpark

Resources - Databricks: Spark Fundamentals Course - YouTube: "Spark for SQL Developers"

Day 2 – Delta Lake Internals

Focus: Why Delta beats traditional data lakes

Concepts - Delta transaction log - ACID guarantees - Time travel

Hands-on - Create Delta table - Perform UPDATE / DELETE / MERGE - Use time travel

Deliverable - Delta table with history + rollback demo

Resources - Delta Lake Documentation - YouTube: "Delta Lake Internals"

Day 3 – Medallion Architecture

Focus: Production-grade data modeling

Concepts - Bronze / Silver / Gold - Data quality & expectations - Schema evolution

Hands-on - Build Bronze → Silver → Gold pipeline - Add basic data quality checks

Deliverable - End-to-end medallion pipeline

Phase 2: Greenfield Implementation (Days 4–8)

Goal: Build Databricks from scratch on **all three clouds**

Topics (Cloud-agnostic) - What customer must provide - Workspace design principles - Environment separation (dev/test/prod)

Cloud-Specific Hands-on

Azure Databricks - VNet injection - ADLS Gen2 - Azure AD service principals

AWS Databricks - VPC, subnets, security groups - S3 + IAM roles - Cross-account access

GCP Databricks - VPC networking - GCS buckets - Service accounts

Resources - Databricks Cloud Setup Docs (Azure/AWS/GCP) - YouTube: "Databricks Networking Explained"

Deliverable - 3 cloud-specific architecture diagrams - Customer onboarding checklist (cloud-agnostic)

Phase 3: Data Ingestion & Patterns (Days 8–13)

Goal: Decide batch vs streaming correctly

Topics - Batch vs Streaming decision framework - CDC patterns - Tools comparison

When to use what | Use Case | Tool | |--------|------| | SaaS ingestion | Fivetran | | On-prem DB | CDC + Auto Loader | | Streaming events | Spark Structured Streaming | | Complex logic | Python / Spark |

Hands-on - Batch ingestion using Auto Loader - Streaming ingestion using Kafka/Event Hub - CDC using MERGE INTO

Resources - Databricks Auto Loader Docs - YouTube: "CDC in Databricks Delta"

Deliverable - Working batch + streaming pipeline

Phase 3: Ingestion & Processing Patterns – Balanced (Days 9–14)

Goal: Confidently choose and implement **batch, streaming, and CDC** based on use case

---

Day 9 – Decision Framework: Batch vs Streaming vs CDC

Focus: Architecture decisions, not tools

Decision Dimensions - Latency SLA (seconds / minutes / hours) - Data volume & burst - Change frequency - Operational complexity

Pattern Matrix | Use Case | Pattern | |--------|--------| | Daily reporting | Batch | | Near-real-time dashboards | Streaming | | DB replication | CDC |

Hands-on - Classify 10 real customer scenarios

Deliverable - Pattern decision cheat sheet

Day 10 – Batch Ingestion (Auto Loader)

Focus: Enterprise-scale file ingestion

Concepts - Incremental file discovery - Schema inference & evolution - Idempotency

Hands-on - Ingest files using Auto Loader - Write to Bronze Delta tables

Deliverable - Production-ready batch pipeline

Day 11 – Streaming Ingestion (Structured Streaming)

Focus: Event-driven pipelines

Concepts - Micro-batching - Exactly-once processing - Watermarking

Hands-on - Stream data from Kafka / Event Hub - Write to Delta

Deliverable - Running streaming pipeline

Day 12 – CDC with Delta Lake

Focus: Database replication patterns

Concepts - Change data capture - MERGE INTO semantics - SCD Type 1 & 2

Hands-on - Apply CDC using MERGE INTO - Maintain target tables

Deliverable - CDC-enabled Silver tables

Day 13 – Orchestration & Reliability

Focus: Production operations

Concepts - Databricks Workflows - Retry & alerting - Idempotent design

Hands-on - Build workflow with dependencies

Deliverable - End-to-end orchestrated pipeline

Day 14 – Tooling Comparison & Architecture Fit

Focus: When to use what (architect view)

Tool	Best Fit
Auto Loader	File ingestion
Streaming	Event pipelines
Python	Complex logic
Fivetran	SaaS sources

Deliverable - Integration architecture decision document

Phase 4: Migration Projects – Oracle, Hadoop & Snowflake (Days 15–19)

Goal: Master end-to-end migrations across the 3 most common customer sources

---

Day 15 – Migration Framework (Universal)

Focus: One framework, many sources

Migration Phases 1. Discovery & assessment 2. Target architecture 3. Code & data migration 4. Validation & reconciliation 5. Cutover & decommission

Hands-on - Create migration checklist usable for any source

Deliverable - Cloud-agnostic migration framework

Day 16 – Oracle / SQL Server → Databricks

Focus: RDBMS modernization

Key Challenges - Stored procedures - Index-heavy designs - Incremental loads

Hands-on - Migrate star schema - Replace stored procedures with Spark SQL - Implement CDC using MERGE

Deliverable - RDBMS-to-Lakehouse migration demo

Day 17 – Hadoop / Hive → Databricks

Focus: Platform consolidation

Key Challenges - HDFS → Cloud storage - Hive metastore → Unity Catalog - MapReduce / HiveQL refactoring

Hands-on - Convert Hive tables to Delta - Optimize with ZORDER

Deliverable - Hadoop migration playbook

Day 18 – Snowflake → Databricks

Focus: Warehouse-to-Lakehouse shift

Key Challenges - Performance expectations - Cost justification - SQL compatibility

Hands-on - Rebuild Snowflake ELT in Databricks SQL - Benchmark performance

Deliverable - Snowflake vs Databricks comparison doc

Day 19 – Parallel Run & Cutover Strategy

Focus: Production risk management

Concepts - Dual writes - Data reconciliation - Rollback strategy

Hands-on - Run parallel pipelines - Validate row counts & aggregates

Deliverable - Cutover & rollback plan

Phase 5: Costing, Proposal & Sizing (Days 18–21)

Goal: Architect-level project sizing across all clouds

Topics - DBU pricing differences (Azure/AWS/GCP) - Cluster sizing by workload type - Migration vs greenfield estimation

Hands-on - Cost estimation for: - Batch pipelines - Streaming pipelines - CDC workloads - Compare Photon vs non-Photon

Sizing Dimensions - Data volume - SLA / latency - Concurrency - Cloud infra costs

Deliverable - Cloud-agnostic proposal template - Cost comparison table (Azure vs AWS vs GCP)

Phase 6: Security & Governance (Days 22–24)

Goal: Enterprise-grade security

Topics - Unity Catalog - RBAC - Data masking - Lineage

Hands-on - Create catalogs, schemas - Assign roles - Enable column masking

Resources - Unity Catalog Documentation - Databricks Security Best Practices

Deliverable - Secured multi-tenant catalog

Phase 7: Catalog & Schema Design (Days 25–26)

Goal: Production-ready data model

Structure

```
Catalog
 └── Schema (domain)
     └── Tables (bronze/silver/gold)
```

Hands-on - Design enterprise catalog - Apply naming standards

Deliverable - Final architecture + documentation

Printable Daily Execution Schedule (Jan 1–26)

This section converts the plan into a **print-ready daily schedule**. You can copy–paste or print this as a checklist.

Days 1–3: Spark & Lakehouse Foundations (4–6 hrs/day)

Day 1 - Spark architecture (Driver, Executors, DAG) - SQL → PySpark transformations - Hands-on: CSV → Delta (SQL + PySpark)

Day 2 - Delta Lake internals - UPDATE / DELETE / MERGE - Time travel & rollback

Day 3 - Medallion architecture - Bronze → Silver → Gold pipeline - Basic data quality checks

---

Days 4–8: Greenfield Implementation (All Clouds)

Day 4 - Customer onboarding checklist - Cloud prerequisites & assumptions

Day 5 - Networking deep dive (Azure vs AWS vs GCP) - Draw 3 network diagrams

Day 6 - Storage + identity integration - ADLS / S3 / GCS access from Databricks

Day 7 - Dev/Test/Prod strategies - Workspace vs catalog isolation

Day 8 - Final greenfield reference architecture - Risks & design decisions

---

Days 9–14: Ingestion & Processing Patterns

Day 9 - Batch vs Streaming vs CDC decision framework

Day 10 - Batch ingestion using Auto Loader

Day 11 - Streaming ingestion (Structured Streaming)

Day 12 - CDC using MERGE INTO (SCD 1 & 2)

Day 13 - Orchestration with Databricks Workflows

Day 14 - Tooling comparison (Auto Loader, Streaming, Fivetran, Python)

Days 15–19: Migration Mastery

Day 15 - Universal migration framework

Day 16 - Oracle / SQL Server → Databricks

Day 17 - Hadoop / Hive → Databricks

Day 18 - Snowflake → Databricks

Day 19 - Parallel run, reconciliation & cutover

Days 20–23: Costing, Sizing & Proposal

Day 20 - Databricks cost model & DBUs

Day 21 - Cluster sizing by workload

Day 22 - Customer-facing proposal creation

Day 23 - Cost optimization & defense

Days 24–26: Security, Governance & Wrap-up

Day 24 - Unity Catalog setup & RBAC

Day 25 - Masking, row-level security, audits

Day 26 - Catalog & schema design - Final review & self-assessment

---

Milestones & Weekly Execution Tracker

(Use the checklist below to track progress)

Week 1 (Days 1–3) – Spark & Lakehouse Foundations

- [ ] Spark architecture understood (Driver, Executors)
- [ ] Same transformations written in SQL and PySpark
- [ ] Delta table created and versioned
- [ ] Time travel demonstrated
- [ ] Bronze → Silver → Gold pipeline built

Exit Criteria: You can explain Spark execution and Delta benefits without slides.

Week 2 (Days 4–8) – Greenfield (Azure + AWS + GCP)

- [ ] Customer onboarding checklist prepared
- [ ] Azure Databricks network diagram
- [ ] AWS Databricks network diagram
- [ ] GCP Databricks network diagram

- [ ] Storage connected securely on all clouds
- [ ] Dev/Test/Prod strategy documented

Exit Criteria: You can whiteboard a greenfield Databricks setup in any cloud.

Week 3 (Days 9–14) – Ingestion & Processing Patterns

- [ ] Batch ingestion using Auto Loader
- [ ] Streaming pipeline running
- [ ] CDC implemented using MERGE INTO
- [ ] Workflow orchestration built
- [ ] Tooling decision document completed

Exit Criteria: You can justify batch vs streaming vs CDC confidently.

Week 4 (Days 15–19) – Migration Mastery

- [ ] Migration framework documented
- [ ] Oracle/SQL Server migration completed
- [ ] Hadoop/Hive migration completed
- [ ] Snowflake migration completed
- [ ] Parallel run & reconciliation executed

Exit Criteria: You can lead a migration cutover discussion.

Week 5 (Days 20–23) – Costing & Proposal

- [ ] DBU cost model understood
- [ ] Cluster sizing matrix created
- [ ] Proposal template completed
- [ ] Cost optimization documented

Exit Criteria: You can defend cost in front of finance & customers.

Week 6 (Days 24–26) – Security, Governance & Final Review

- [ ] Unity Catalog configured
- [ ] RBAC & masking applied
- [ ] Audit logs & lineage reviewed
- [ ] Catalog & schema blueprint finalized

Exit Criteria: You can pass an enterprise security review.

Phase 6 – Security & Governance (Days 24–25)

Perspective: Databricks-native + enterprise compliance

Day 24 – Unity Catalog Foundations

Goal: Centralized governance across clouds

Key Concepts - Metastore vs catalog vs schema - RBAC vs ABAC - Workspace binding

Hands-on - Create Unity Catalog metastore - Create catalogs per domain - Bind workspaces

Deliverable - Multi-workspace Unity Catalog setup

Day 25 – Enterprise Security & Compliance

Goal: Pass security & audit reviews

Security Controls - Row-level security - Column masking (PII) - Audit logs - Data lineage

Hands-on - Implement column masking - Apply row filters - Enable audit logging

Deliverable - Compliance-ready security model

Phase 7 – Catalog, Schema & Data Modeling (Day 26)

Goal: Production-grade data organization

Recommended Structure

```
Catalog (per domain / BU)
 └── Schema (bronze / silver / gold)
     └── Tables
```

Hands-on - Design enterprise catalog hierarchy - Apply naming standards - Validate access isolation

Deliverable - Final catalog & schema blueprint

Final Self-Assessment – Architect Readiness Test

Use this section to **objectively validate** whether you are ready to operate as a **hands-on Databricks Delivery Architect**. Answer **YES / NO** honestly. Any NO means revisit that area.

---

1. Foundations (Spark & Lakehouse)

- [ ] I can explain Spark execution (Driver, Executors, DAG, shuffle)

- [ ] I can convert complex SQL logic into PySpark confidently
- [ ] I understand Delta Lake internals (transaction log, ACID)
- [ ] I can debug performance issues conceptually

Pass Criteria: All YES

---

2. Greenfield Architecture (Azure / AWS / GCP)

- [ ] I can whiteboard Databricks architecture on any cloud
- [ ] I know customer prerequisites for greenfield setup
- [ ] I can explain private networking clearly
- [ ] I understand storage + identity integration on all clouds

Pass Criteria: All YES

---

3. Ingestion & Processing Patterns

- [ ] I can choose batch vs streaming vs CDC without hesitation
- [ ] I have built Auto Loader pipelines
- [ ] I have built streaming pipelines
- [ ] I can implement CDC using MERGE INTO

Pass Criteria: All YES

---

4. Migration Leadership

- [ ] I can lead discovery for migration projects
- [ ] I know how to migrate RDBMS to Databricks
- [ ] I can migrate Hadoop/Hive workloads
- [ ] I can explain Snowflake → Databricks tradeoffs
- [ ] I know how to run parallel systems and cutover safely

Pass Criteria: All YES

---

5. Costing, Sizing & Proposals

- [ ] I can estimate Databricks costs confidently
- [ ] I can size clusters based on workload
- [ ] I can explain DBUs to non-technical stakeholders
- [ ] I can defend cost optimizations

Pass Criteria: All YES

---

6. Security & Governance

- [ ] I understand Unity Catalog deeply
- [ ] I can design RBAC across teams

- [ ] I can implement PII masking & row-level security
- [ ] I understand audit logging & lineage

Pass Criteria: All YES

---

7. Catalog & Data Modeling

- [ ] I can design domain-driven catalogs
- [ ] I understand schema evolution strategies
- [ ] I can explain Bronze/Silver/Gold clearly

Pass Criteria: All YES

Final Verdict

- ✅**All sections passed:** You are architect-ready
- ⚠️**Any NO:** Revisit that phase before claiming readiness

Repository README (Use as-is)

Databricks Architect Playbook

This repository represents my **hands-on journey to mastering Databricks as a delivery architect** across **Azure, AWS, and GCP**.

What this repository demonstrates

- Greenfield Databricks architecture (multi-cloud)
- Batch, streaming, and CDC pipelines
- Enterprise migration strategies
- Costing, sizing, and proposal design
- Security, governance, and Unity Catalog

Repository Structure

```
├── foundations/
├── greenfield/
├── ingestion-patterns/
├── migration/
├── costing/
├── security/
├── catalog-design/
└── TRACKER.md
```

How to use this repository

- Each folder contains **design notes + hands-on artifacts**

- TRACKER.md is used to track execution progress
- This repo can be used for **interview discussion, delivery reference, or onboarding**

Outcome

By completing this playbook, I can: - Design and implement Databricks platforms end-to-end - Lead enterprise migrations - Own architecture, delivery, and cost discussions

---

Need Professional Help in Developing Your Architecture?

Please contact me at https://sammuti.com :)