Perfect ✅ — let’s merge both into one **6-Month Hardcore Roadmap (Backend + Data Engineering Hybrid)**.  
This way you don’t have to “choose,” you’ll **train as a Backend Engineer who also has Data Engineering depth** → a killer combo for product & data-driven companies.

**🚀 6-Month Backend + Data Engineering Mastery Plan**

**📅 Month 1 – Foundations + SQL Power**

**Backend (Java Core + DSA):**

* JVM internals, garbage collection, memory model.
* Concurrency: threads, locks, executors, CompletableFuture.
* DSA: 50–60 LeetCode Mediums (arrays, strings, sliding window, hashmaps).
* Read: *Effective Java* (cover to cover).

**Data (SQL First):**

* Master SQL:
  + Window functions, CTEs, JOINS, GROUP BY, indexes.
  + Platform: StrataScratch + LeetCode Database.
* Mini Project:
  + Create a SQL-heavy dashboard (Postgres) for analytics.
  + Example: Sales DB → queries like “Top 5 products by revenue in last 30 days.”

✅ **Checkpoint:**

* Write multi-threaded Java code confidently.
* Solve at least 50 SQL queries with complex joins & windows.

**📅 Month 2 – Spring Boot + ETL Basics**

**Backend:**

* Spring Boot advanced:
  + REST APIs, Spring Security (JWT), Spring Data JPA.
  + Reactive WebFlux.
* Databases:
  + Postgres (indexes, query plans).
  + Redis caching.

**Data:**

* Learn ETL concepts:
  + Batch vs Streaming pipelines.
  + Intro to Apache Airflow (task DAGs).
* Build a mini ETL pipeline:
  + CSV → Airflow DAG → Transform → Load into Postgres.
* DSA: Trees, Linked Lists, Stack/Queue (60–70 problems).

✅ **Checkpoint:**

* Deploy a Spring Boot REST API with JWT auth + Postgres DB.
* First Airflow DAG running on local machine.

**📅 Month 3 – Distributed Systems + Streaming**

**Backend:**

* System design basics: caching, load balancing, CAP theorem.
* Hands-on with Docker (multi-stage builds).
* Deploy Spring Boot service in Docker.

**Data:**

* Kafka deep dive:
  + Producers, consumers, partitions, offsets.
  + Build Kafka → consumer pipeline in Java.
* Spark basics (PySpark or Scala).
* Mini project:
  + Kafka (ingest logs) → Spark (transform) → Postgres (store results).

**DSA:**

* Graphs + BFS/DFS problems.
* At least 70–80 new LeetCode problems.

✅ **Checkpoint:**

* Kafka pipeline working end-to-end.
* Spark job processes real Kafka data → aggregates into DB.

**📅 Month 4 – Real Project Buildout**

**Main Project Start → “Distributed Job Scheduler with ETL Pipeline”**

* Components:
  + Spring Boot backend for job definitions.
  + Scheduler: DAG execution (topological sort, retries).
  + Kafka → logs all job executions.
  + Spark pipeline processes logs for analytics.
  + Store job run history in Postgres.
  + Redis caching for fast job lookups.
  + Airflow to orchestrate extra ETL jobs.

**Backend add-ons:**

* Monitoring: Prometheus + Grafana.
* Spring Boot integration tests (JUnit5 + Testcontainers).

**DSA:**

* Dynamic Programming + advanced graph problems.

✅ **Checkpoint:**

* Scheduler runs jobs in DAG order.
* Kafka → Spark → DB analytics pipeline works.
* Monitoring dashboard online.

**📅 Month 5 – Scaling + Cloud**

**Backend:**

* Kubernetes basics (minikube/kind).
* Deploy microservices on K8s.
* Service mesh basics (Istio/Linkerd).

**Data:**

* Cloud Storage:
  + AWS S3 → store pipeline results.
  + Glue/Athena for querying data.
* Extend ETL:
  + Kafka → Spark → S3 → Redshift/Parquet.

**Testing:**

* Load testing with JMeter.
* Contract testing with Pact.

**DSA:**

* 70–100 hard problems (DP on graphs, union-find, advanced trees).

✅ **Checkpoint:**

* Project running on Kubernetes cluster.
* ETL pipeline stores results in S3 + queries via Athena.

**📅 Month 6 – Resume Polish + Open Source**

**Finalize Project:**

* “Distributed Job Scheduler + ETL Analytics.”
  + Spring Boot microservices.
  + Kafka + Spark ETL pipeline.
  + Airflow orchestrating extra jobs.
  + Monitoring (Prometheus, Grafana).
  + Deployed on Kubernetes + AWS.

**Resume Boosters:**

* Open source:
  + Contribute a small feature/bug fix to Airflow, Spark, or Kafka.
* Write technical blog:
  + “How I built a distributed job scheduler with Kafka + Spark.”

**Interview Prep:**

* Mock system design interviews weekly.
* SQL challenges + backend coding problems.
* DSA final push: 50 hardest problems.

✅ **Final Outcome:**

* One resume-killer hybrid project (Backend + Data Eng).
* Spring Boot + Cloud + System Design ready.
* SQL + ETL + Kafka + Spark exposure.
* You can apply confidently to **Backend Engineer** *and* **Data Engineer** roles.

⚡ This roadmap makes you:

* **Backend-strong** (Spring, microservices, system design).
* **Data pipeline-capable** (ETL, Kafka, Spark, Airflow).
* **Resume-proof** with one hybrid project that showcases *both*.

👉 Do you want me to **sketch the architecture diagram** for the “Distributed Job Scheduler + ETL Analytics” project (with components + flow), so you have a clear system blueprint before building?