Interview Preparation Guide

Table of Contents

Data Engineer Role - Anand Raj

Resume Analysis Summary

Your updated resume is **interview-ready** and positions you well for AWS-focused data engineer roles. Here are the key strengths:

✓ Resume Strengths

- **3+ years** of data engineering experience with measurable impact
- **AWS Solutions Architect – Associate** certified
- **Hybrid cloud expertise** spanning AWS and Azure ecosystems
- Strong technical foundation in **PySpark, Airflow, Docker**
- **Quantifiable achievement**: 65% error reduction in regulatory reporting
- Modern architecture experience (**Medallion Architecture**)
- **Notice period** clearly stated (1 January 2026)
- **GitHub profile** linked for technical validation

Interview Question Categories

1. AWS Data Engineering Questions

Core AWS Services

- **AWS Glue**: Explain how you'd use Glue for schema evolution and ETL workflows
- **S3 Data Lake**: Design principles, partitioning strategies, and lifecycle policies
- **Lambda**: Best practices for serverless data processing, error handling, and timeout management
- **Step Functions**: Orchestrating complex multi-service workflows with error handling
- **EMR vs Glue**: When to choose each service for big data processing
- **Athena**: Query optimization techniques, partition projection, CETAS usage

Expected Deep-Dive Topics

- How do you handle incremental data loads in AWS?
- Explain your approach to data cataloging with AWS Glue
- Design a cost-optimized data lake architecture on S3
- How would you migrate Informatica ETL to AWS Glue?
- Explain DynamicFrames vs DataFrames in Glue
- Describe error handling and retry mechanisms in Lambda functions

2. PySpark Technical Questions

Spark Fundamentals

- Explain Spark architecture: Driver, Executors, Cluster Manager
- Difference between transformations (lazy) and actions (eager)
- RDD vs DataFrame vs Dataset - use cases for each
- How does Spark's DAG (Directed Acyclic Graph) work?

Performance Optimization

- How do you handle data skew in Spark jobs?
- Explain partitioning, bucketing, and their impact on performance
- When would you use broadcast joins vs shuffle joins?
- Caching and persistence strategies - when to use MEMORY_ONLY vs MEMORY_AND_DISK
- How do you tune Spark configurations (executor memory, cores, partitions)?
- Explain adaptive query execution (AQE) in Spark 3.x

Hands-On Coding

- Read data from S3, apply transformations, write to Delta Lake

- Implement window functions for running totals and rankings
- Handle null values and data quality checks
- Join multiple datasets efficiently
- Work with nested JSON structures

3. Apache Airflow Questions

Core Concepts

- What is a DAG and how do you define task dependencies?
- Explain different **Executors**: Sequential, Local, Celery, Kubernetes
- **XComs**: How they work and when to avoid them (large data)
- **TaskFlow API** vs traditional operators - benefits of decorators
- **Sensors**: Purpose and use cases (S3KeySensor, FileSensor)
- **Hooks**: Connection management for external systems

Advanced Topics

- How do you handle task failures, retries, and SLA monitoring?
- Explain dynamic DAG generation
- What are SubDAGs and TaskGroups? When to use each?
- How do you implement cross-DAG dependencies?
- Describe your containerization approach (Docker) for Airflow
- Best practices for DAG design and maintainability

Scenario-Based

- Your DAG is taking too long to execute - how do you optimize?
- How would you migrate legacy cron jobs to Airflow?
- Design an Airflow pipeline for daily incremental data loads
- How do you handle secrets and connection management?

4. Medallion Architecture & Data Modeling

Architecture Concepts

- **Bronze Layer**: Raw data ingestion, what goes here and why?
- **Silver Layer**: Data cleansing, deduplication, validation
- **Gold Layer**: Business-ready aggregated data, dimensional models
- When would you skip a layer or add a fourth layer?

Implementation Questions

- How do you implement Bronze → Silver → Gold in Azure Databricks?
- Explain your approach to schema evolution across layers
- How do you handle late-arriving data in the Silver layer?
- What data quality checks do you implement at each layer?
- Explain your use of **CETAS** in Synapse Analytics
- How do you ensure idempotency in your transformations?

Data Modeling

- Star schema vs Snowflake schema - when to use which?
- Explain **Slowly Changing Dimensions (SCD)** Type 1, 2, 3
- How do you implement **Change Data Capture (CDC)**?
- Partitioning strategies for large fact tables
- Handling many-to-many relationships in dimensional models

5. Project-Specific Questions

Azure E-commerce Pipeline Project

- Walk me through your Medallion Architecture implementation
- Why did you choose **parameterized pipelines** in Azure Data Factory?
- How did you integrate MongoDB with Databricks for enrichment?
- Explain your Bronze → Silver → Gold transformation logic
- How did you replicate the architecture on **GCP DataProc**?
- What performance optimizations did you implement?
- How did you handle incremental vs full loads?

Airflow NASA APOD Pipeline Project

- Why did you refactor from XCom to **TaskFlow API**?
- Explain your Docker containerization approach
- How did you set up PostgreSQL in a separate Docker container?
- How do you handle NASA API rate limits?
- What monitoring and alerting did you implement?
- Walk me through your data validation logic

Moody's Rating Project (Current Role)

- Explain your Python validation tool architecture (Pandas, NumPy)
- How did you achieve **65% error reduction**?
- What specific data quality checks did you implement?

- How do you generate XML reports for regulatory submissions?
- Describe your root cause analysis process for rating failures
- How did you modernize from **Sybase/Informatica to AWS**?
- What challenges did you face with European regulatory compliance?

6. Scenario-Based & System Design Questions

Design Challenges

- Design a real-time streaming pipeline using Kafka and Spark Streaming
- How would you build a data lake on AWS S3 with proper governance?
- Design a multi-region data replication strategy for disaster recovery
- How would you implement data lineage tracking across your pipelines?
- Design an incremental ETL pipeline for 100+ source tables
- How would you handle a data quality incident in production?

Troubleshooting Scenarios

- A Spark job is running out of memory - how do you diagnose and fix it?
- Your Airflow DAG is stuck in running state - what do you check?
- Data loaded to Silver layer doesn't match Bronze - how do you investigate?
- AWS Glue job is taking 3 hours instead of 30 minutes - how do you optimize?
- You discover duplicate records in the Gold layer - what's your approach?

Migration & Modernization

- How would you migrate a legacy Informatica pipeline to AWS Glue?
- Design a phased approach to move from on-premises Hadoop to AWS EMR
- How do you ensure zero data loss during migration?
- What testing strategy would you use for migrated pipelines?

7. SQL & Python Coding Questions

Advanced SQL

- Window functions: `ROW_NUMBER()`, `RANK()`, `DENSE_RANK()`, `LAG()`, `LEAD()`
- Complex joins (INNER, LEFT, RIGHT, FULL OUTER, CROSS, SELF)
- Common Table Expressions (CTEs) and recursive queries
- Subqueries (correlated and non-correlated)
- PIVOT and UNPIVOT operations
- Query optimization techniques (indexes, execution plans)

- Handling NULLs with `COALESCE()`, `NULLIF()`

Python for Data Engineering

- Pandas DataFrame operations (groupby, merge, pivot)
- Reading/writing various file formats (CSV, JSON, Parquet, Avro)
- Error handling with try-except blocks
- Working with nested JSON structures
- File I/O operations and directory handling
- Datetime manipulation for partitioning
- List comprehensions and lambda functions

Coding Practice Problems

- Find top N customers by revenue in each region
- Calculate running totals and moving averages
- Deduplicate records based on composite keys
- Implement SCD Type 2 logic in SQL
- Parse and flatten nested JSON in Python
- Identify gaps and islands in sequential data

8. Behavioral Questions (Leadership Principles)

Amazon and many tech companies assess behavioral competencies. Prepare STAR (Situation, Task, Action, Result) format answers:

Customer Obsession

- Tell me about a time you had to balance technical debt with customer needs
- Describe how you've prioritized data quality for downstream users

Ownership

- Tell me about a time you took ownership of a problem outside your direct responsibility
- Example: Taking initiative to document the validation tool for team use

Bias for Action

- Tell me about a time you made a decision with incomplete information
- Example: Prioritizing bug fixes for critical regulatory reports

Learn and Be Curious

- How do you stay updated with data engineering trends?
- Tell me about a time you learned a new technology quickly (e.g., TaskFlow API)

Insist on the Highest Standards

- Tell me about a time you improved data quality standards
- Example: Your 65% error reduction achievement

Earn Trust

- Tell me about a time you had a disagreement with a team member
- How do you handle feedback about your code or pipeline design?

Deliver Results

- Tell me about a project you delivered under tight deadlines
- Example: Meeting regulatory submission deadlines

Think Big

- How would you scale your current architecture for 10x data volume?
- What's your vision for modern data platforms?

Preparation Strategy

Week 1-2: Technical Deep Dive

- [ ] Study AWS Glue documentation and best practices
- [ ] Practice PySpark on Databricks Community Edition or local setup
- [ ] Review Airflow TaskFlow API and modern DAG patterns
- [ ] Deep dive into Medallion Architecture implementation examples
- [ ] Practice SQL on LeetCode (Medium to Hard problems)

Week 3: Project & Behavioral Prep

- [ ] Prepare 30-second, 2-minute, and 5-minute versions of each project
- [ ] Write STAR format stories for 10+ behavioral scenarios
- [ ] Practice whiteboard system design problems
- [ ] Review your GitHub repositories and be ready to walk through code
- [ ] Mock interviews with peers or use Pramp/Interviewing.io

Week 4: Advanced Topics

- [ ] Study data quality frameworks (Great Expectations, Deequ)
- [ ] Review schema evolution strategies (Delta Lake, Hudi)
- [ ] Practice error handling patterns in data pipelines
- [ ] Study AWS Well-Architected Framework for data analytics
- [ ] Review cost optimization strategies for cloud data platforms

Final Week: Polish & Confidence

- [ ] Daily mock interviews
- [ ] Review common interview mistakes and how to avoid them
- [ ] Prepare 5-7 thoughtful questions for the interviewer
- [ ] Relax and visualize success
- [ ] Sleep well before interview day

Key Talking Points for Your Background

Highlight Your Hybrid Cloud Expertise

"I have hands-on experience with both AWS and Azure, which allows me to design vendor-agnostic solutions. In my current role, I modernized legacy Sybase/Informatica systems to AWS while building scalable pipelines in Azure Databricks and Data Factory."

Emphasize Quantifiable Impact

"I reduced regulatory submission errors by 65% through a Python-based validation framework using Pandas and NumPy. This directly improved compliance and significantly reduced manual remediation efforts, saving approximately 20 hours per week for the team."

Showcase Modern Best Practices

"I've implemented the Medallion Architecture with Bronze/Silver/Gold layers, containerized Airflow pipelines with Docker, and migrated DAGs to TaskFlow API for better maintainability. This demonstrates my commitment to modern, scalable data engineering practices."

Leverage Your AWS Certification

"As an AWS Solutions Architect Associate, I bring architectural thinking to data engineering. I design solutions considering cost optimization, security, reliability, and operational excellence - not just functionality."

Address Legacy Modernization Experience

"My experience with legacy systems (Sybase, Informatica, SoapUI) and successfully migrating them to cloud-native solutions positions me well for companies undergoing digital transformation."

Potential Concerns to Address Proactively

Limited Production-Scale AWS Experience

"While my production AWS experience is growing, I've been actively building projects on EMR, Glue, and Step Functions. My AWS certification demonstrates my theoretical knowledge, and I'm eager to apply it at scale in this role."

No Real-Time Streaming on Resume

"My resume focuses on batch ETL, but I have foundational knowledge of Kafka and have been learning Spark Structured Streaming through personal projects. I'm excited to expand into real-time data processing."

Single Employer Since 2022

"I've stayed at Cognizant to gain deep expertise across diverse projects - from regulatory reporting to AWS migration. I've grown from associate to mid-level engineer and am now ready for new challenges that will push my technical boundaries."

Questions to Ask the Interviewer

About the Role

- What does the data architecture currently look like, and where do you see it evolving?
- What's the balance between greenfield projects and maintaining existing pipelines?
- What does success look like in the first 3, 6, and 12 months?

About the Team

- How is the data engineering team structured?
- What's the collaboration model with data scientists and analysts?
- How does the team balance technical debt with new feature development?

About Technology

- What's your current tech stack and are there plans to adopt new technologies?
- How do you handle schema evolution and data quality?
- What's your approach to CI/CD for data pipelines?

About Growth

- What learning and development opportunities are available?
- How does the team stay current with data engineering trends?
- Is there a clear career progression path?

Final Checklist

Before the Interview

- [ ] Review your resume thoroughly - every line
- [ ] Test your internet connection, camera, and microphone
- [ ] Have a backup device ready (phone as hotspot)
- [ ] Prepare your environment (quiet room, good lighting)
- [ ] Have water, pen, and paper ready
- [ ] Open a code editor if asked to share screen

During Technical Rounds

- [ ] Think aloud - explain your reasoning
- [ ] Ask clarifying questions before diving into solutions
- [ ] Start with a high-level approach, then go into details
- [ ] Consider trade-offs (cost, performance, complexity)
- [ ] Use proper technical terminology
- [ ] Admit if you don't know something and explain how you'd find out

During Behavioral Rounds

- [ ] Use the STAR format consistently
- [ ] Be specific with examples
- [ ] Show what YOU did (use "I" not "we")
- [ ] Highlight results and learnings
- [ ] Be honest - interviewers can detect embellishment

After the Interview

- [ ] Send a thank-you email within 24 hours
- [ ] Reflect on what went well and what to improve
- [ ] Follow up with your recruiter on timeline

Resources for Practice

Coding Practice

- LeetCode SQL (Medium/Hard problems)
- HackerRank Python & SQL challenges
- Stratascratch (data engineering specific)

System Design

- "Designing Data-Intensive Applications" by Martin Kleppmann
- AWS Well-Architected Framework documentation
- System Design Interview resources on GitHub

Mock Interviews

- Pramp (free peer mock interviews)
- Interviewing.io (anonymous technical interviews)
- Blind's interview preparation forum

Community Learning

- r/dataengineering on Reddit
- Data Engineering Weekly newsletter
- AWS re:Invent and Azure Friday videos
- Databricks blog for Medallion Architecture examples

Confidence Boosters

✓ You have **3+ years of real-world experience**
✓ You have **AWS certification** backing your knowledge
✓ You've achieved **measurable impact** (65% error reduction)
✓ You've worked with **modern tools** (Airflow, Docker, Databricks)
✓ You have **diverse experience** (legacy + cloud, Azure + AWS)
✓ Your **notice period is clear** - ready to join by early Jan 2026

You are well-prepared. Trust your experience and be yourself!

Good luck with your interviews! Remember: They're evaluating if you're a good fit, but you're also evaluating if they're the right opportunity for you.