AWS Data Pipeline Challenge - Healthcare Facility Analytics

Overview

You'll be building a complete data pipeline on AWS to analyze healthcare facility accreditation data. This challenge simulates real scenarios you'd encounter in our data engineering role.

Due Date: 5 days from initial assignment

Rolling Review Process: Submissions are reviewed on a first-come, first-served basis. Early submissions may receive interview invitations before the deadline, so we encourage submitting as soon as you're satisfied with your solution.

Challenge Context

You're working with healthcare facility data stored as JSON in S3. Each facility record contains information about services, employee counts, and accreditations with expiry dates. Your task is to build an automated pipeline that processes this data and provides insights into facilities with expiring accreditations.

Prerequisites

- AWS Free Tier account
- Basic knowledge of Python, SQL, and AWS services
- **AI Usage Policy**: You may use AI tools (ChatGPT, Claude, etc.) as coding assistants and guides, but the final implementation and understanding must be your own

Technical Stages

Complete a minimum of TWO of the following four stages. Choose the stages that best demonstrate your strengths and interests.

Combination recommendations for showcasing skills:

- **SQL + Python** (1+2): Data engineering pipeline skills
- **Lambda + Step Functions** (3+4): Serverless architecture expertise
- **SQL + Lambda** (1+3): End-to-end data processing
- **Python + Step Functions** (2+4): Complex workflow orchestration

Stage 1: Data Extraction with Athena

Objective: Write an Athena SQL query to extract key facility metrics from JSON data in S3.

Requirements:

- Extract: `facility_id`, `facility_name`, `employee_count`, `number_of_offered_services`, and `expiry_date_of_first_accreditation`
- Handle nested JSON structure efficiently
- Save query results to S3
- Sample data file will be provided for testing

Stage 2: Data Processing with Python

Objective: Create a Python script using boto3 to filter facilities with soon-to-expire accreditations.

Requirements:

- Read JSON records from S3 bucket
- Filter facilities with any accreditation expiring within 6 months
- Write filtered records to a different S3 location
- Include proper error handling and logging

Stage 3: Event-Driven Processing with Lambda

Objective: Build a Lambda function that automatically processes new data uploads.

Requirements:

- Trigger on new JSON file uploads to S3
- Execute Athena query to count accredited facilities per state
- Store query results appropriately

- Handle Lambda timeouts and retries

Stage 4: Workflow Orchestration with Step Functions

Objective: Design a Step Functions state machine for the complete pipeline.

Requirements:

- Trigger on new S3 data arrival
- Invoke Lambda to run Athena query
- Wait for query completion
- On success: copy results to production location
- On failure: send SNS notification
- Include proper error handling and retry logic

Deliverables

Required Submissions

1. **Code Repository**: GitHub repository with code for your 2 chosen stages, documentation, and architecture diagrams
2. **AWS Resources**: Screenshots/evidence of deployed resources for your chosen stages
3. **Demo Video**: Walkthrough of your solution
4. **Stage Selection Rationale**: Brief explanation (1 paragraph) of why you chose those specific stages

Submission Format

Rolling Review: We review submissions as they arrive and schedule interviews on a first-come, first-served basis. Submit early if you're ready!

Email the following to **rwdostert@medlaunchconcepts.com** with subject "AWS Challenge Submission - [Your Name]":

1. **GitHub Repository Link**
2. **Stage Selection**: Which 2 stages you completed and why
3. **AWS Architecture Screenshot** (CloudFormation diagram or manual architecture drawing)

4. **Video Demo Link** (YouTube, Loom, or similar - unlisted/private is fine)
5. **Cost Report**: Screenshot of AWS billing dashboard showing resource usage
6. **Brief Summary** (2-3 paragraphs) explaining your approach and any challenges faced

Easy Evidence Collection Tips

- Use AWS CloudFormation/CDK for infrastructure as code
- Take screenshots of AWS console showing deployed resources
- Use `aws logs` CLI to capture Lambda execution logs
- Record your screen while testing the pipeline end-to-end

Evaluation Criteria

Technical Implementation (40%)

- Code quality, structure, and best practices
- Proper error handling and logging
- Security considerations (IAM roles, least privilege)

AWS Knowledge (30%)

- Appropriate service selection and configuration
- Understanding of serverless patterns
- Cost optimization awareness

Documentation & Communication (20%)

- Clear README and code comments
- Architecture explanation
- Demo quality and presentation

Innovation & Efficiency (10%)

- Creative problem-solving approaches
- Performance optimizations
- Additional features or improvements

Resources & Hints

AWS Free Tier Limits

- Lambda: 1M requests/month, 400,000 GB-seconds compute
- Athena: 1TB data scanned/month
- S3: 5GB storage, 20,000 GET requests, 2,000 PUT requests
- Step Functions: 4,000 state transitions/month

Helpful Documentation

- AWS Athena JSON SerDe
- Boto3 Documentation
- AWS Lambda Best Practices
- Step Functions Patterns

Sample Test Data

We'll provide a sample dataset, but you're encouraged to create additional test cases to demonstrate robust error handling.

Important Notes

- **Stage Selection Strategy**: Choose stages that showcase different skills (e.g., SQL + Python, or Lambda + Step Functions) rather than similar technologies
- **Clean Up**: Remember to delete AWS resources after submission to avoid charges
- **Questions**: Feel free to email with clarifying questions (we encourage this!)
- **Time Management**: With 7 days total, aim to complete your chosen stages efficiently rather than perfecting every detail
- **Early Submission Advantage**: Rolling reviews mean earlier submissions may secure interview slots sooner
- **Real-World Focus**: Think about production considerations like monitoring, alerting, and scalability

Next Steps

After submission, we'll review your solution and discuss your implementation choices, architectural decisions, and potential improvements during the interview.

Good luck! We're excited to see your solution.

This challenge is designed to assess practical AWS skills relevant to our data engineering role. Focus on demonstrating understanding of cloud-native patterns and best practices.