Cost Optimization Challenge: Managing Billing Records in Azure

Problem Summary

- Cosmos DB is being used to store 2M+ large billing records (~300 KB each).
- Read-heavy workload, but older than 3 months = rarely accessed.
- Cosmos DB costs are high due to storage and throughput.
- Requirements: Cost optimization, zero downtime, no data/API contract changes, and data must still be available within a few seconds.

Solution Overview: Tiered Storage using Azure Serverless Components

Split your data into two storage tiers:

1. Hot Tier ? Cosmos DB (for recent 3 months data)
2. Cold Tier ? Azure Blob Storage (for >3 months data)

A read-through cache pattern will ensure old data access remains seamless without modifying existing APIs.

Architecture Components

- Cosmos DB: Store latest 3 months of billing records
- Azure Blob Storage: Store archived billing records (>3 months) as JSON/Avro/Parquet
- Azure Functions (Serverless): Middleware to fetch from Blob if not found in Cosmos DB
- Azure Durable Functions: For background archival processing (cold data migration)
- Azure Table / Queue Storage: Track archival status or flag moved documents

Implementation Plan

Step 1: Archival Logic (Background)

- Use Azure Durable Functions (timer triggered daily/weekly)
- Query Cosmos DB for records older than 3 months
- Move them to Azure Blob Storage (partitioned by YYYY/MM)
- Delete from Cosmos after confirmation

Step 2: Middleware for Seamless Reads

- Read API first queries Cosmos DB

- If not found, use Azure Function to fetch from Blob Storage
- Return result to caller; optional: cache into Cosmos DB

Step 3: Storage Format
- Use compressed JSON or Parquet
- Organize by: billing-records/year=2023/month=02/record_1234.json

Benefits of the Solution

- ? Cost Optimization: Blob Storage is ~90% cheaper than Cosmos
- ? No Data Loss: Data is backed up before deletion
- ? No Downtime: Durable Functions run in background
- ? No API Contract Change: Middleware handles fallback
- ? Simplicity: Azure Functions + Blob is easy to manage

Optional Enhancements

- Use Azure Data Lake Gen2 if analytics needed
- Use Azure Cache for Redis to cache cold records
- Add diagnostic logs/alerts for failed archival reads

Cost-Saving Estimation

- Cosmos DB: ~$0.25?0.30/GB/month
- Azure Blob (Hot): ~$0.018/GB/month
- Azure Blob (Cool): ~$0.01/GB/month

Archiving 1M records (~300 GB) can save ~$70?80/month or more.

Architecture Diagram (Text View)

```
+------------------+        Not Found        +--------------------+
      |   Cosmos DB     | <----------------------- | Azure Function    |
      +--------+---------+                        +----+---------------+
              |                                  |
    Query (3 mo recent)                          Fetch from Blob
              |                                  |
              v                                   v
```

```
Return if found                    Return if archived
      |                                  |
      +--------------------->  Return to API layer  <-------------+
                             (No change)                 |
                                                         |
        +----------------+  Background Archival  +------------+
        | Durable Func   | <---------------------| Cosmos DB  |
        +----------------+                       +------------+
              |
   Move data to Blob (Cold)
```