AWS Data Store Practice Tasks (Free Tier Friendly)

AWS Data Store Practice Tasks (Free Tier Friendly)

TASK 1: DynamoDB CRUD with Balanced Access Patterns

Objective: Create a serverless backend using AWS Lambda and DynamoDB that handles full CRUD for a user profile system.

What to do:

- Create a DynamoDB table `Users` with:
  - Partition key: `userId` (UUID)
  - Sort key: `createdAt`
  - Use high-cardinality values for `userId`
- Implement Lambda functions for:
  - createUser, getUser, updateUser, deleteUser
- Explore query vs scan for filtering by attribute

Covers: CRUD, high-cardinality partition keys, query vs scan, serialization, DynamoDB basics

TASK 2: Eventual vs Strong Consistency Testing

Objective: Observe consistency behavior in DynamoDB queries.

What to do:

- Write a Lambda that:
  - Updates a DynamoDB item
  - Immediately queries it with both strong and eventually consistent reads
- Introduce a delay and observe changes

Covers: Consistency models, read behavior under load

TASK 3: DynamoDB Indexing and TTL Cleanup

Objective: Build a session store with auto-expiry using DynamoDB TTL.

What to do:

- Create a `Sessions` table with:
  - Partition key: `sessionId`
  - Attributes: `userId`, `expiresAt`
- Enable TTL on `expiresAt`
- Add a Global Secondary Index (GSI) on `userId`

Covers: TTL, GSI, indexing strategy, ephemeral storage patterns

TASK 4: Caching with Amazon ElastiCache or Simulated Cache Layer

Objective: Use caching strategies with Lambda + DynamoDB.

What to do:

- Simulate read-through caching with Lambda:
  - On cache miss, fetch from DynamoDB and store in a "cache"
- Implement lazy loading and compare performance
- Use `ttl` attribute to simulate cache expiration

Covers: Caching (read-through, lazy loading), TTL, performance tuning

TASK 5: Using Amazon S3 for Object Storage

Objective: Store profile pictures or documents in S3.

What to do:

- Use Lambda to upload, read, and delete S3 files
- Store the S3 `objectKey` in DynamoDB
- Enable versioning and server-side encryption
- Access files using presigned URLs

Covers: S3 object storage, metadata linking, security, persistence

TASK 6: S3 Lifecycle Management and Tiers

Objective: Apply lifecycle rules to manage S3 storage cost-effectively.

What to do:

- Upload files with tags or prefixes like `logs/`, `temp/`, `archived/`
- Set lifecycle rules to:
  - Move to Glacier or IA after a few days
  - Delete objects after 7 days in `temp/`

Covers: S3 tiers, lifecycle management, persistent vs temporary storage

TASK 7: RDS Integration (Optional if budget allows)

Objective: Set up Amazon RDS and run relational queries.

What to do:

- Create a free-tier RDS MySQL or PostgreSQL instance
- Create a `Products` table

- Use Lambda with RDS Proxy to run queries

Covers: Relational DB use, persistence, connection pooling, SQL

TASK 8: Serialize & Deserialize Structured Data

Objective: Persist complex objects in DynamoDB and S3.

What to do:

- Convert a nested object to JSON and store in DynamoDB and S3
- Retrieve and deserialize

Covers: Serialization, structured storage, S3 vs DynamoDB trade-offs

TASK 9: Analyze and Optimize Queries

Objective: Improve data access efficiency in DynamoDB.

What to do:

- Simulate poor design and identify bottlenecks with CloudWatch
- Redesign with better partitioning/indexing

Covers: Performance tuning, indexing, balanced access