AWS Lambda Practice Tasks (Free Tier Friendly)

AWS Lambda Practice Tasks (Free Tier Friendly)

==========

1. Basic Lambda Setup & Configuration

Objective: Learn how to set up and configure Lambda functions.

- Create a Lambda function from scratch using Python or Node.js.
- Configure:
  - Runtime
  - Memory (start with 128-512 MB)
  - Timeout (10s)
  - Handler (e.g., index.handler)
- Define environment variables like ENV=dev, S3_BUCKET=my-test-bucket.
- Package and attach a Lambda Layer with a utility library (e.g., requests for Python).
- Set a trigger using S3 (uploading a file) or API Gateway (HTTP event).

Covers: Configuration, environment variables, runtime, handler, layers, triggers

2. Statelessness & Event Source Mapping

Objective: Work with an event source and ensure your function remains stateless.

- Create a DynamoDB table.
- Enable DynamoDB Streams.
- Create a Lambda that is triggered by insert/update events from the table.
- Use Event Source Mapping to bind the stream to the Lambda.
- Insert a few items and log the event data in CloudWatch Logs.

Covers: Event source mapping, stateless behavior, event-driven architecture

3. Event Lifecycle, Error Handling & Destinations

Objective: Learn how Lambda handles errors and how to route results.

- Modify your Lambda to throw errors based on input (e.g., a missing field).
- Attach a Dead-Letter Queue (DLQ) using Amazon SQS.
- Add Lambda Destinations:
  - One SNS topic for successful execution
  - One for failure
- Test with different inputs to trigger success/failure flows and observe results.

Covers: Error handling, DLQ, Lambda Destinations, event lifecycle

4. Access Private Resources in a VPC

Objective: Explore how to connect a Lambda function to private AWS resources.

- Create a VPC with public and private subnets.
- Launch a free-tier RDS (MySQL/PostgreSQL) in the private subnet.
- Create a Lambda function inside the VPC.
- Use environment variables for DB connection strings.
- Confirm successful connection from Lambda to the RDS instance.

Covers: VPC access, private networking, environment configuration

5. Performance Tuning and Concurrency

Objective: Test how Lambda handles performance tuning and concurrency limits.

- Use a Step Function or custom script to invoke the Lambda concurrently.

- Set Reserved Concurrency to 2 and observe throttling.

- Monitor cold starts and duration in CloudWatch Logs.

- Adjust memory from 128 MB to 1024 MB and compare performance metrics.

Covers: Tuning, concurrency, cold starts, CloudWatch metrics

6. Unit Testing & Deployment

Objective: Write tests and deploy Lambda using automation tools.

- Write unit tests using pytest (Python) or jest (Node.js).

- Use AWS SAM CLI:

  - sam init, sam build, sam local invoke

  - sam deploy --guided for deployment

- Store configuration in samconfig.toml.

- Validate function behavior locally and on AWS.

Covers: Unit testing, SAM, local testing, deployment automation

7. Service Integrations

Objective: Build a real-world mini application with multiple AWS services.

- Create an API Gateway HTTP API that triggers a Lambda function.

- Lambda reads data from S3 and writes results to DynamoDB.

- Publish a success notification to SNS.

- Secure all services with proper IAM roles.

Covers: Multi-service integration, IAM, practical serverless architecture