Assignment 3

CSCI 5410 (Serverless Data Processing)
Date Given: Jul 10, 2023
Due Date: Jul 24, 2023 at 11:59 pm
Late Submissions are not accepted.
A deduction of 10% per day will be applied for late submission.

To avoid any additional charges for resource consumption - Delete AWS services after fulfilling the assignment submission requirements.

Objective:
This assignment covers concepts of Lambda, a very important computing framework for Serverless cloud computing. In addition, this assignment provides an opportunity to build event driven application using SNS and SQS.

Plagiarism Policy:

- This assignment is an individual task. Collaboration of any type amounts to a violation of the academic integrity policy and will be reported to the AIO.
- Content cannot be copied verbatim from any source(s). Please understand the concept and write in your own words. In addition, cite the actual source. Failing to do so will be considered as plagiarism and/or cheating.
- The Dalhousie Academic Integrity policy applies to all material submitted as part of this course. Please understand the policy, which is available at: https://www.dal.ca/dept/university_secretariat/academic-integrity.html

Assignment Rubric - based on the discussion board rubric (McKinney, 2018)

	Excellent (25%)	Proficient (15%)	Marginal (5%)	Unacceptable (0%)	Problem # where applied
Completeness including Citation	All required tasks are completed	Submission highlights tasks completion. However, missed some tasks in between, which created a disconnection	Some tasks are completed, which are disjoint in nature.	Incorrect and irrelevant	**Part A** **Part B** **Part C**
Correctness	All parts of the given tasks are correct	Most of the given tasks are correct However, some portions need minor modifications.	Most of the given tasks are incorrect. The submission requires major modifications.	Incorrect and unacceptable	**Part A** **Part B** **Part C**
Novelty	The submission contains novel contribution in key segments, which is a clear indication of application knowledge.	The submission lacks novel contributions. There are some evidence of novelty, however, it is not significant	The submission does not contain novel contributions. However, there is an evidence of some effort.	There is no novelty	**Part A** **Part B** **Part C**

Clarity	The written or graphical materials, and developed applications provide a clear picture of the concept and highlights the clarity.	The written or graphical materials, and developed applications do not show clear picture of the concept. There is room for improvement	The written or graphical materials, and developed applications fail to prove the clarity. Background knowledge is needed.	Failed to prove the clarity. Need proper background knowledge to perform the tasks.	**Part A** **Part B** **Part C**

Citation:
McKinney, B. (2018). The impact of program-wide discussion board grading rubrics on students' and faculty satisfaction. Online Learning, 22(2), 289-299.

Tasks:

This assignment has 3 parts. Part A is related to research. Part B, and Part C is related to event driven application design and implementation.

Part A. Explore & Build a Use Case:

Read an overview of AWS Kinesis, and check how it works. Now, build a use case based on a hypothetical scenario, where you can use AWS Kinesis, and any other required AWS service(s). The use case should be unique (not copied from online sources/ friends/ colleagues), and it should reflect your understanding of AWS or any other cloud services.

You need to write about your hypothetical scenario and the use case in two paragraphs (less than 1 page). In addition, you need to provide a block diagram or activity diagram or workflow of the use case. Two things are very important in this assignment (1) Novelty and (2) Use of the appropriate service

****Please do not write in bullet points. Write proper paragraphs**

Part A - Submission requirement: A pdf file with the use case, graphical representation, and citation(s) in IEEE format. Do not forget to provide inline citation (if needed)

Saurabh.dey@dal.ca

Part B. **Build an event-driven serverless application using AWS Lambda.**

In this assignment, you need to use S3 bucket, DynamoDB, and Lambda Functions.
**[B00xxxxxx = your B00 number] used in bucket naming

take screenshots at every step and submit as part of the PDF:

Step 1. Create your 1st S3 bucket SampleDataB00xxxxxx and upload the files given in the Tech folder one at a time with a delay of 100 milliseconds. You need to write a script or use the SDK to upload the files one at a time to the S3 bucket.

Step 2. If a file is available on the 1st bucket, then it triggers **extractFeatures** Lambda function, which is the 1st lambda function. This lambda function extracts the Named entities from the file and creates a JSON array of named entities* for that file.
E.g. 001.txt contains Asia, Soviet, Serbia etc., then the JSON array created by the function should be **"001ne": {"Asia":1, "Soviet":1.....etc.}.**

Step 3. This file will be saved as **001ne.txt** in a new bucket - **TagsB00xxxxxx**.

Step 4. Once a file is available on this 2nd bucket, then **accessDB** Lambda function will automatically be triggered. accessDB is your 2nd Lambda function. This Lambda function reads each named entity JSON file and updates the DynamoDb database table (two fields - key, value).
E.g. 001ne.txt contains "001ne": {"Asia":1, "Soviet":1.....etc.}. Then this Lambda function will update a DynamoDb database table where "Asia" will be a key for field "1" will be the value.

Step 5. Test your Lambda functions, and the entire application and provide screenshots. Please write proper test cases. (Note: This is functional testing of the application, not unit test of the code)

Part B - Submission requirement:

From Step 1 to Step 5, submit screenshots of every steps. Please do not exclude any steps. Include all screenshots as part of a PDF file. In addition, provide the program/scripts as part of the PDF file.

Additional Information:

Dataset Citation: D. Greene and P. Cunningham. "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering", Proc. ICML 2006.

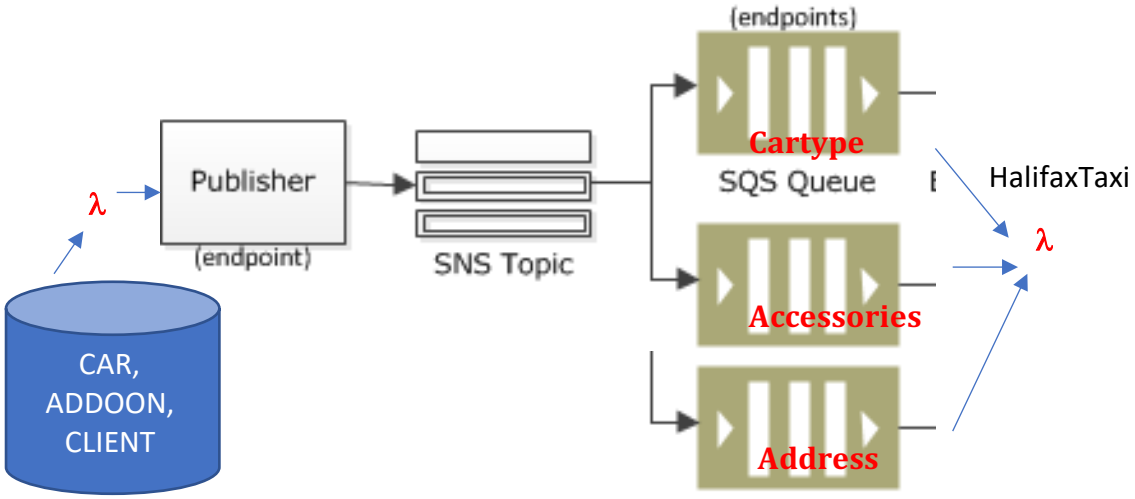
Named Entity: “In information extraction, a named entity is a real-world object, such as persons, locations, organizations, products, etc., that can be denoted with a proper name. It can be abstract or have a physical existence. Examples of named entities include Barack Obama, New York City, Volkswagen Golf, or anything else that can be named.” – Wikipedia

Hint: Any word that starts with an **uppercase** or all letters of the word is in uppercase, then it can be considered as a named entity. E.g. Saurabh, IBM, DAL, Halifax, Serverless etc.

Part C. Use AWS Lambda-SQS-SNS:

take screenshots at every step and submit as part of the PDF:

1. Bob owns HalifaxTaxi, which is an online Car delivery service.
2. They receive orders online. You deliver the Car, and Bob does the paper work.
3. The customers' orders are added to a Queue (standard SQS)–
    a. Assume a program (λ Function) is sending random order messages to HalifaxTaxi
    b. This message simulates how a customer places an order to an online car rental site.
    c. The program should randomly pick **car types** (Compact, mid-size Sedan, SUV, Luxury), **car accessories** (GPS, Camera) from lists **CAR, ADDON.**
    d. The program also randomly picks 1 street address from another list **CLIENT**. You can randomly write some addresses like "6050 University Avenue"
    e. Create three messages containing car type, car accessories, and street address and send to HalifaxTaxi, where these messages are received in three standard queues



4. Bob periodically (every 2 minutes) checks (λ Function), if there is any order in the Queue.
5. If message is available, it is assumed Bob has prepared the paper work, and then a notification service (SNS) is triggered which sends the details to your email, so that you can deliver the car to the client's address.

Part C - Submission requirement:

From 1 to 5, submit screenshots of every steps. Please do not exclude any steps. Include all screenshots as part of a PDF file. In addition, provide the program/scripts as part of the PDF file.

