

HU Extension

Handed out: 10/10/2015

Assignment 06

E-90 Cloud Computing

Due by 11:59 PM EST on Friday, 10/16/2015

Place all of your narratives and illustrations in a single Word or PDF document named E90_LastNameFirstNameHW06.docx [.pdf]. Use this assignment as the initial template. Add your steps and your code below problem statements used for that problem. Upload your homework file and your working code (e.g., filename.java) into your Assignment 6 folder. Do not include executables.

Problem 1.

- I. Manually populate an S3 bucket with Images and Resumes of movie stars and Nobel laureates:
 - a. Create folder structures with two folders: **stars** and **nobels**. Inside either folder you should have subfolders **images** and **resumes**.
 - b. From Google Images fetch images (pictures) of three movie stars and three Nobel winners. Produce 3+3 MS Word documents with bogus resumes for selected movie stars and Nobel winners on your own. The resumes should be very short; three sentences each.
 - c. Upload those images and resume into your S3 bucket manually.
 - d. **Manually** grant general public access to those images and resumes.
- II. Create one SimpleDB domain and programmatically insert one row for each movie star and each Nobel laureates - write your program either in Java or .Net or your favorite language supported by AWS:
 - a. Use their full names with hyphens between the first and the last name as the item keys.
 - b. In rows for movie stars record full name, most popular movie, S3 URL of the picture of the star and S3 URL of his or her resume.
 - c. Use those same attributes for the Nobel laureates and add the year they won the prize and the field of science in which they got the prize as two additional attributes.
 - d. Demonstrate that you can change (correct) the year of one Nobel prize award, programmatically.
 - e. Demonstrate that you can delete one movie star from SimpleDB domain programmatically.
 - f. Capture the content of your database as displayed in the Database Development perspective.

Provide working code and capture all stages of testing. Write your program either in Java or .Net or your favorite language supported by AWS. **[30 Points]**

Problem 2.

DynamoDB version. Use the same S3 bucket, images and resumes created in Problem 1. Use AWS DynamoDB Console for all the work

- a) Create a new table, "CELEBRITIES_CONSOLE", where primary key is of a Hash type, and is constructed as "firstName" + "-" + "lastName". (for example, "Julia-Roberts")

- b) Insert three rows for movie stars, with attributes: full name (= firstName + " " + lastName), most popular movie, S3 URL of the picture of the star and S3 URL of his or her resume
- c) Insert three rows for Noble laureates, with attributes: full name (same format as above), S3 URL of the picture of the lauret, S3 URL of his or her resume, award year, science field in which they got the prize
- d) Explore the table in the Console and search by providing filters on ID - both scans and get by ID types of queries
- e) Update an award year of one of the Nobles rows - demo that the data is updated. Capture all stages of testing. **[15 Points]**

Problem 3.

Use AWS SDK (Java or any other) to do the same work as in Problem 2, with minor changes:

- a) Same - except name the table "CELEBRITIES_SDK"
- b) Same
- c) Same
- d) Query both CELEBRITIES_CONSOLE and CELEBRITIES_SDK tables via SDK APIs - demonstrate that you see the inserted data
- e) Same

Provide working code and capture all stages of testing. **[25 Points]**

Problem 4.

Create a Lambda function that will monitor DynamoDB stream of table CELEBRITIES_SDK and write all operations performed on that table into CloudWatch logs. Use Java class developed in Problem 3 to generate load on table CELEBRITIES_SDK. Comment out the section of the class where you create the table itself. Capture all steps of your development and testing. **[20 Points]**

Problem 5.

When done, delete all DynamoDB tables and Lambda functions, programmatically or using AWS CLI. Provide working code and capture all stages of testing. **[10 Points]**

Files SimpleDBTest .java and DynamoDBSample.java are provided.

You can find several examples of DynamoDB code written in Node.js at:

<http://docs.aws.amazon.com/amazondynamodb/latest/gettingstartedguide/GettingStarted.NodeJs.html>

You can find several examples of DynamoDB code written in Java at:

<http://docs.aws.amazon.com/amazondynamodb/latest/gettingstartedguide/GettingStarted.Java.html>

You can find several examples of DynamoDB code written in Python at:

<http://docs.aws.amazon.com/amazondynamodb/latest/gettingstartedguide/GettingStarted.Python.html>