CSC 470 – Special Topics in Computer Science: Cloud Computing

Small Project #2 – AWS Simple Storage Service (S3)

Objective

Develop a small Java application that utilizes various S3 services and features to raise your awareness and comprehension of S3 and its documented API.

Due Date

This assignment is due by 9:00am, Friday, February 20th, 2015.

Specifics

Create a small Java application, which demonstrates the use of the S3 API (see http://goo.gl/4fTLNh). Your application should be coded against the AWS SDK for Java API and all required implementation activities should be executed programmatically by your application (the client).

Your solution can be a command line Java application, or a GUI-based Java application; the choice is yours. In either case, the interface that the user (me) uses should allow for me to confirm the execution of the action (PUT object, GET object) step-by-step via the source code, the output, and via the AWS console. That is, your application should not run without pausing allowing me to verify the most recent action. You will need to build in programmatic pauses that are controlled by the user (e.g. "Press return to continue.") so that I may verify each action.

This project will be graded by the amount of demonstrable, functioning features that your implementation incorporates as follows. Bolded items below are actions and should be executed as such. Each item for a grade must be implemented and clearly executing successfully for that grade to be awarded. That is, you cannot receive a "B" grade if you don't implement and successfully implement all three points (Put Object – Copy, Get Service, and Put Object). You cannot receive an "A" unless all of the "B", "C", and "D" items are satisfied.

The table below contains bolded actions. These actions should correspond directly to RESTful[1] operations on S3. However, you should use the AWS SDK for Java to form proper requests of the service (allowing for the API to do the work for you to make the REST call to the S3 service). Thus, you'll want to understand the REST operation and refer to the API documentation to determine how make a similar call to the service via the API. Note that the sample code in the SDK for Java download will be a helpful resource.

[1] See http://en.wikipedia.org/wiki/Representational_state_transfer for more information on REST.

Grade / # Points	Implementation Required *(higher grades must include required implementation from lower grades)*
A (4)	• **PUT Bucket versioning** (enables versioning on a bucket) • **PUT Object** (put one or more new versions of an object) • **DELETE Object** (delete a specific version of an object) • **HEAD Object** (retrieves the meta-data from an object)
B (3)	• **Put Object – Copy** (copies an object from one bucket to another) • **GET Service** (obtains a listing of all buckets) • **PUT Object** (set [at time of put or to an existing object, your choice] the expiration of the object to be 5 days in the future (from time of upload/execution), and set the storage class to Reduced Redundancy)
C (2)	• **PUT Bucket** (creates a new bucket) • **DELETE Bucket** (deletes an empty bucket) • **HEAD Bucket** (determines if a bucket exists & if you have permissions to access it)
D (1)	• *Create a new file (text or binary, used as the object below)* • **PUT Object** (places an object into an existing bucket) • **GET Object** (retrieves an object from an existing bucket) • **DELETE Object** (deletes an object from a bucket)
F (0)	Fails to execute, fails to compile, no working functionality

Finally, change the IAM group policy for my user account (which you created in Project 1) so that I can use the AWS console to confirm all of the above actions. I should be able to do so via the AWS console for all actions except for the GET and HEAD actions.

These actions will be verified by source code and user display/review from the application (that is, I should see the downloaded object when you save it to the file system, and you should display the HEAD information via the application once it is returned to the application from AWS).

Resources

- AWS SDK for Java - http://goo.gl/de5Hby
- AWS SDK for Java API Reference - http://goo.gl/4WZO1
- S3 Developer Guide - http://goo.gl/4fTLNh
- S3 API Reference - http://goo.gl/5Ea8BP
- Compiling Java code with included jar files - http://stackoverflow.com/a/9396410

Delivery

- Your delivery of this assignment should be in the form of a Java jar file that contains both your source code and .class files. I expect not to have to compile your code. Other files directly relating to the project are permitted as needed (e.g. a file containing AWS security credentials, if you wish; an ANT build file). Your jar file should not contain backup files or other items that are directly supportive of the program solution. Your code may be packaged if you wish.
- Do not include the aws-sdk jar file, I will use the AWS provided SDK for Java jar file (currently aws-java-sdk-1.9.18 available at http://goo.gl/OIaInr).
- You should code your solution to include the AWS SDK for Java jar file via the command line at both compilation and execution time. Do not uncompress the jar files from AWS and bundled them in your jar file. For example:
    - `javac –cp .:aws-java-sdk-1.9.18.jar Foo.java`
    - `java –cp .:aws-java-sdk-1.9.18.jar Foo`
- Upload only your jar file as a deliverable to the corresponding assignment in Canvas.