CIS4010 Individual Assignment 1

Due: Monday, January 27 @ 8AM

Task 1

In both AWS and Azure, do the following in Python or Jupyter Notebooks:

a) Create containers named CIS1300, CIS3110, and CIS4010.
b) Populate the containers with the following objects:

CIS1300	CIS3110	CIS4010
1300Lecture1.pdf	3110Lecture1.pdf	4010Lecture1.pdf
1300Lecture2.pdf	3110Lecture2.pdf	4010Lecture2.pdf
1300Assignment1.pdf	3110Lecture3.pdf	4010Assignment1.pdf
1300Assignment2.pdf	3110Assignment1.pdf	
1300Assignment3.pdf		
1300Assignment4.pdf		

The files that you will use for objects are found in the file, Objects.tar.gz, and is available on CourseLink.

c) Using this object store, write an interactive Python/Jupyter Notebooks program that displays the objects in
   i. All containers
   ii. A specified container
   iii. With a specified name in any of the containers (i.e. must search all containers)
   iv. For a specified object, downloads that object to a file (on the client system) using the same name as the object

Task 2

Create a NoSQL table with the Movies information available in JSON format in the file, Movies.zip, on the CourseLink site.

Write an interactive Python/Jupyter Notebooks program that will use the Movies database to allow a user to do the following type of queries and then chose to save the displayed results to a **csv** file.

Query and display parameters

- Partition and/or sort key – individual value or range of values
- Filter by any attribute
- Sort by key or attribute
- Only display certain attributes

Task 3

Compare the two implementations for both Task 1 and Task 2. Cover aspects such as implementation, performance, permissions, available documentation and tutorials, etc. Also provide a reference list of any resources used to do Tasks 1 and 2.

Task 4

For all aspects that you needed to use for the implementation, create a user manual that

a) explains each procedure used in AWS and maps it to the equivalent procedure in Azure, along with an explanation of that Azure procedure;
b) identifies any aspect of either AWS or Azure that does not easily map to one in the other cloud service.