Assignment 2 (30%)

Objective:

You will write a program that allows you to automate and manage your cloud services. This assignment will test your ability to use the Boto and Apache LibCloud libraries against AWS and OpenStack cloud services as well as your general Python scripting skills. Use free tier on AWS wherever possible.

Requirements:

1. The program will present the user with a Python menu for AWS or OpenStack. This will direct the user to subsequent submenus outlined below. A back option should be provided to allow the user to easily navigate the menu and to choose another service after an initial choice.

    - Compute
        - AWS
            - List all running instances printing out a message "Running AWS EC2 instances" followed by information such as the AMI ID, instance type, region the instance is deployed in and the time it was launched (example screenshot provided).
            - Allow users to choose from a list of running instances or enter an instance ID. Display the same information as above.
            - Start a new instance based on an existing AMI
            - Stop all instances
            - Stop a specific instance (prompt user for the instance ID)
            - Attach an existing volume to an instance
            - Detach a volume from an instance
            - Launch a new instance (not previously addressed). This method should create a brand new AMI EC2 VM (EBS storage). Prompt the user to enter either "windows" or "linux" depending on the free tier usage OS they wish to deploy. Assume all instances will be deployed in the EU west region.
        - OpenStack
            - List all running instances
    - Storage
        - AWS
            - List all buckets
            - List all objects in a bucket (Allow users to choose from a list of buckets or enter a bucket name)
            - Upload an object
            - Download an object
            - Delete an object
        - Openstack

        - As above
- Monitoring (AWS Only)
    - Display all default performance metrics gathered for a particular EC2 instance (Prompt the user for the EC2 instance in question).
    - Set an alarm such that if the CPU utilization is less than 40% an alarm will be raised. When an alarm is raised, use the AWS SNS to send a notification to an email account. The user should be able to specify the email address as a menu option at the command line.
- Choose from **one** of the following free tier Amazon service. Describe what it does (in the python script) and provide a detailed example of how you can manipulate it using boto (give 4 short examples). You should describe how you tested these features.
    - Autoscaling
    - AmazonDB
    - Relational DB Service
    - Elastic Load Balancing
- Choose a service you have never used before and that there is no tutorial available for (just the API reference). Demonstrate how to use it with one simple method. http://boto.cloudhackers.com/en/latest/index.html#currently-supported-services

Importantly, please note that you should use best practice when writing your code i.e. use classes/methods where sensible, minimise hardcoding paths i.e. use the Python *os* module etc and minimising code duplication. Marks will be awarded for this. Code should be robust and include error checking so it can deal accordingly with error messages. A boto configuration file should be used so that credentials are imported in the Connections.py class.

Submission Details:

Due Date: 13th April 2014, 1pm.

Submit your code (all Python scripts) along with a read me file in a zipped directory via Blackboard. This directory should have your name and student number e.g. Aisling O Driscoll R00012345. An assignment submission facility will be made available in Blackboard. The read me should include a program description, requirements and details of how to run the script (refer to sample readmes on GitHub if necessary).

Your code must include very detailed, line by line comments explaining what each line of code is doing. This should clearly demonstrate your understanding of the submitted code. An example is provided below and the marking scheme will be heavily weighted towards this. Please include these comments after your code in the same file using **‘’’** and **‘’’** characters, with each explanation corresponding to a particular line of code.

The readme is a brief file detailing how I should use the scripts, what menu items are expected **i.e. everything I need to test and run your code.**

At the discretion of the examiner, students may be briefly interviewed to demonstrate their understanding of the submitted code.

Example:

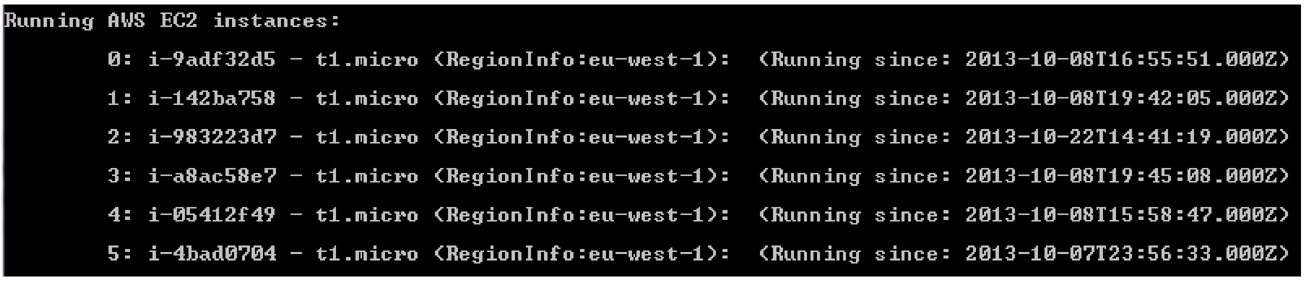
’’’

line 17. The run method returns a Reservation object which represents a collection of instances etc. There are a number of arguments that need to be specified including etc. Argument 1 is of type etc.

line 24. The image is being loaded on the AMI and update the user that this process is running. It will update the user every 10 seconds until it is completed. To do this I am using the inbuilt ‘x’ method in Python which returns etc etc.

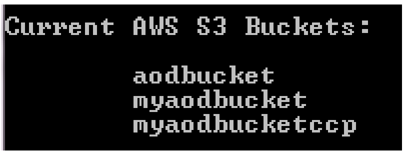
‘‘

Sample Screen for List all Instances:



```
Running AWS EC2 instances:
        0: i-9adf32d5 - t1.micro (RegionInfo:eu-west-1):  (Running since: 2013-10-08T16:55:51.000Z)
        1: i-142ba758 - t1.micro (RegionInfo:eu-west-1):  (Running since: 2013-10-08T19:42:05.000Z)
        2: i-983223d7 - t1.micro (RegionInfo:eu-west-1):  (Running since: 2013-10-22T14:41:19.000Z)
        3: i-a8ac58e7 - t1.micro (RegionInfo:eu-west-1):  (Running since: 2013-10-08T19:45:08.000Z)
        4: i-05412f49 - t1.micro (RegionInfo:eu-west-1):  (Running since: 2013-10-08T15:58:47.000Z)
        5: i-4bad0704 - t1.micro (RegionInfo:eu-west-1):  (Running since: 2013-10-07T23:56:33.000Z)
```

Sample Screen for List all Buckets:



```
Current AWS S3 Buckets:

        aodbucket
        myaodbucket
        myaodbucketccp
```