# Fall 2020 CIS4010

## Assignment 2: One-Click Deployment of VMs and Containers

## Part 1

Your task is to automate deployment of VMs to the AWS or Azure Cloud platform. You will only deploy Linux VMs. On these VMs you can also load multiple Docker images and start some of them. Your goal is to make this as close to a one-click action as possible.

If you are working on AWS you will use Python3 and Boto3. If you are working on Azure you will use the Azure CLI (using either a Python3 program or a bash shell to generate and execute the calls for Azure CLI).

**Pre-Conditions**

You may establish pre-conditions before your "one-click" script is activated. Everyone will have the first pre-condition but others are possible as long as you document them and they do not require onerous amounts of work (*i.e.* do not try to game the assignment – most of the work is done in the script).

**Pre-Condition 1: Deployment description files**

Use a freely available editor, spreadsheet, etc. to create a files in csv, JSON, or YMAL format.

- Items to be described in the Template file (**template.xxx** where is csv, json, *etc*.) include:

| **Template Name** | **Amazon Machine Image (AMI)/Azure Image Name** | **Instance Type / Azure Size** | **Root Volume Size (GiB)** | **Security Group Name / Azure Inbound Ports** | **Zone / Region** |
|---|---|---|---|---|---|
| *WebPortals* | *ami-084672f05313c1fb1* | *t2.micro* | *default* | *Launch-wizard-3* | *ca.central-1* |
| *DataAnalysis* | *ami-04312317b9c8c4b51* | *t2.micro* | *16* | *default* | *ca.central-1* |

- For each template provide the following information in the Instances file (**instances.xxx** where is csv, json, *etc*.):

| **Template Name** | **Instance/VM Name** | **ssh key / Azure Key pair name** | **Container Package Name** |
|---|---|---|---|

| | | | |
|---|---|---|---|
| *WebPortals* | *WP-1* | *WPKey.pem* | *conPac-1* |
| *WebPortals* | *WP-2* | *WPKey.pem* | *conPac-1* |
| *DataAnalysis* | *DA-1* | *DAKey.pem* | *conPac-2* |
| *DataAnalysis* | *DA-2* | *DAKey.pem* | *conPac-2* |
| *DataAnalysis* | *DA-3* | *DAKey.pem* | *conPac-3* |
| *DataAnalysis* | *DA-4* | *DAKey.pem* | *conPac-3* |

- The Docker images that are to be deployed on each instance and where those images reside (*i.e.* Docker Hub public registry, your private registry on Docker Hub, docker file in your directory) will be in the Container file (**container.xxx** where is csv, json, *etc*.):

| **Container Package Name** | **Container** | **Location** | **Start script** |
|---|---|---|---|
| *conPac-1* | *mongo* | *Docker hub* | *startMongo.sh* |
| *conPac-2* | *golang* | *Docker hub* | |
| *conPac-2* | *r-base* | *Docker hub* | *batchScriptMode.sh* |
| *conPac-3* | *hellocloud* | *Docker hub/dastacey* | |

**Example of a Start Script**
*batchScriptMode.sh*
docker run -ti --rm r-base bash

**Pre-Condition 3**: Create ssh keys

**Pre-Condition 4:** Populate your Docker Hub account with containers or create some containers and put them on your own machine.

**One-Click Deployment**

- Write a Python3 script (**launch.py**) that will use all the pre-condition information to start creating VMs and configuring them.
- This script must report on the status of what is happening and report errors if they occur.
- For authentication purposes, your personal AWS authentication information stored on your system will be used.

**Monitoring**

- After you have executed your "one-click" deployment, how will you demonstrate that everything has been done correctly?
- Write another Python program called **monitor.py** that will display all of the information that you can get about your running instances via boto3. You must select the information to display and design how it is to be displayed.

## Part 2

For the tasks that you were asked to do in Part 1, suggest what existing AWS services could be used to do the similar tasks and compare the capabilities and easy of use for your programs and the AWS services.