CS454/554- ASSIGNMENT #4 25 POINTS

TEAM MEMBERS: **TEAM NUMBER:** _____

_____

_____

_____ *print only this coversheet for grading*

PROJECT GOAL

The objective of this assignment is to interface with OpenStack's CLI and BOTO3 to authenticate to the system and launch a VM. Mastery of these steps would enable one to develop software to manage and configure a cloud system without using the dashboard.

OVERVIEW

OpenStack has been integrated into several programing languages via API package. Python itself as several to choose from. In this assignment you will use python boto3 package to manage OpenStack. Reference sections 14.2.1.1, 14.2.1.2, and 14.2.1.3 in your textbook for more information on boto3 and integration with AWS.

PROJECT REQUIREMENTS

Requirement	Completion (To Be Verified by Instructor or TA) *Grader Comments*
1. Use OpenStack CLI to create a Key Pairs	*10 points*
2. Get an OS image ID and Flavor name via BOTO3 interface	*5 points*
3. Create a VM Instance	*5 points*
4. Login to your new instance using the IP netns commands from project 2	*5 points*

STEP 1: LOG INTO CS CLOUD MANAGE NODE

1. SSH to your Linux VM

```
Username: team#
Password: team#_pass
REPLACE # WITH YOUR TEAM NUMBER
```

STEP 2: CREATE KEY PAIRS

2. Get the openstack variables needed to access the openstack cli and aws ec2 cli

```
source ~/team#_openrc
REPLACE # WITH YOUR TEAM NUMBER
```

3. Create ec2 credentials to use the aws ec2 cli in python with boto3

```
openstack ec2 credentials create
*This will create and list your Access and Secret keys
*If you forget them or don't save them somewhere they can be accessed by running the command
openstack ec2 credentials list
```

4. Create a new keypair for your project and save the private key as a pem file

```
openstack keypair create NEW_KEYPAIR_NAME > NEW_KEYPAIR_NAME.pem
*This will create a new keypair and store the private key in a pem file named after your keypair
```

STEP 3: CREATE A PYTHON SCRIPT

5. Script example:

```
#!/usr/bin/python
###### NOTE
###### WHETHER YOU CREATE A SCRIPT OR RUN FROM A PYTHON PROMT
###### THESE LINES ARE ALWAYS THE FIRST STEP
import boto3
import os
##### If you stored your access key and secret key in an environment
variable you
##### can access them like so:
access_key = os.environ.get('ACCESS_KEY_VARIABLE')
secret_key = os.environ.get('SECRET_KEY_VARIABLE')

ec2=boto3.resource('ec2', aws_access_key_id='your access key value here',
                   aws_secret_access_key='your secret key value here',
                   region_name='RegionOne',
                   endpoint_url='http://192.168.122.251:8788')
###### END NOTE

##### Get Image id, using the code examples below
##### With the Image ID create a new instance and list them out
```

STEP 4: VERIFICATION

6. Open the dashboard by going to the following URL http://192.168.122.251/dashboard and logging in using the credentials you used to log into the controller node.

You should see under Project->Compute->Instances a new instance that matches the information you listed using the python commands.

ADDITIONAL PYTHON CODE SNIPPITS

PYTHON CODE EXAMPLE for BOTO3 execution:

```
// Code to run boto3
// You must have first installed boto3 in your manage node linux system.
// boto3 has already been installed in our environment

import boto3
ec2=boto3.resource('ec2',aws_access_key_id='your access key value here',
                   aws_secret_access_key='your secret key value here',
                   region_name='RegionOne',
                   endpoint_url='http:// 192.168.122.251:8788')


instances=ec2.instances.all()
for w in instances:                 // the first time you access
   print w                          // OpenStack you should have no
                                    // instances yet.

images=ec2.images.all()
for w in images:
   print w

image=ec2.Image('ami-b7378f53')  // note the image may be different
image.name                       // print the image name to make sure
                                 // it is cirros

ec2.create_instances(ImageId='YOUR_IMAGE_ID', InstanceType='m1.tiny',
                     KeyName='NAME_OF_NEW_KEYPAIR', MinCount=1,
                     MaxCount=1)
## Notice how the KeyName argument does not include the .pem extension

instances=ec2.instances.all()    // if successful, you should see your new
for w in instances:              // instance show up in this list
   print w                       // you can verify on the dashboard
```