CIS*4010 (01) F20 - Cloud Computing MV

Assignment 2: One-Click Deployment of VMs and Containers

Grading Instructions

Put all code and documentation in your A2 repo on GitLab.

For Part 1

You are required to install 2 VMs

1. Amazon Linux 2 (ami-0947d2ba12ee1ff75) type=t2.micro size=12GiB

Security group must contain 80 (http) and 8080 (custom TCP open to all)

- you can great an appropriate security group via the console

Zone=us-east-1

Azure: Linux (debian 10.6) size=DS1_v2 Security/Networking=open ports 22, 80, and 8080

Everything else default

Region=Canada East

2. Ubuntu Server 20.04 LTS (ami-0dba2cb6798deb6d8) type=t2.micro size=10GiB

Security group must contain 80 (http) and 8080 (custom TCP open to all)

- you can great an appropriate security group via the console

Zone=us-east-1

Azure: Linux (centos 8.2.2004) size=DS1_v2 Security/Networking=open ports 22, 80, and 8080

Everything else default

Region=Canada East

You can name them anything that you want. You can also create the key files

any way that you want.

There is only one container package containing 2 containers:

1. **hello-world** - to be pulled from Docker Hub

2. **nginx** - to be pulled from Docker Hub and run using the instructions in

https://hub.docker.com/_/nginx. Use the following part of the instructions on this page:

"Alternatively, a simple Dockerfile can be used to generate a new image that includes the necessary content

(which is a much cleaner solution than the bind mount above):

```
FROM nginx
```

```
COPY static-html-directory /usr/share/nginx/html
```

Place this file in the same directory as your directory of content ("static-html-directory"),

run

```
$ docker build -t some-content-nginx .
```

then start your container:

```
$ docker run --name some-nginx -d -p 8080:80 some-content-nginx
```

You should then be able to see the message that you put in the content file on http://xx.xx.xx.xx/index.html "

Hints

1. On your VM create a directory called "webcontent" in your home (ubuntu/ec2-user) directory. Therefore, static-html-directory is webcontent

2. In that directory create a file called index.html with html code such as

```
<html>
```

```
<body>
```

```
It works!

</body>

</html>
```

Make sure that this file has rw-rw-rw permissions.

I have tested these instructions and you can see the results at http://34.238.251.133:8080/ (Amazon Linux 2) and http://3.83.123.46:8080/ (Ubuntu 20.04).

You should also see the following on your instances:

```
$ sudo docker image ls

REPOSITORY           TAG          IMAGE ID          CREATED           SIZE

some-content-nginx   latest       7675f8d365dc      26 minutes ago    133MB

nginx                latest       f35646e83998      2 weeks ago       133MB

hello-world          latest       bf756fb1ae65      9 months ago      13.3kB
```

```
$ sudo docker container ls

CONTAINER ID       IMAGE                    COMMAND                  CREATED
STATUS             PORTS                    NAMES

09689d9a4119       some-content-nginx   "/docker-entrypoint.…"   27 minutes ago      Up
26 minutes       0.0.0.0:8080->80/tcp   some-nginx
```

Grading Demo

1. Book a time with Marshall through the Doodle Poll that will be posted by the end of the week.

2. The grading will be done on Zoom (the link will be the one used for Marshall's office hours).

3. Log on to Zoom at the time allotted to you. You will be using your own AWS or Azure account to demo to Marshall so you will have to share your screen when he asks you to.

4. Here are the steps for the demo of Part 1:

a) Show Marshall the template files that you have constructed for the required VM's and anything that you have pre-created (such as keys, security groups, etc.).

b) Explain briefly how your system works and if there are functions that it does not do.

c) Run your program **launch.py** to instantiate the VM's and load up the Docker containers.

d) Ssh into each VM and show Marshall that the required images and containers are present and go to the website on each instance to show him the web message.

e) Display your Python script **monitor.py** to demonstrate that you can monitor your VMs.

e) If you have any other VM's and/or Docker containers that you can load that are different than the required ones, show Marshall now - explain why this goes beyond the requirements.

Part 1 Grading Scheme

launch.py

- successfully instantiates the required VM's (12 marks)
- successfully loads the containers (6 marks)
- extra VM's or containers demonstrated (4 marks)

monitor.py

- successfully monitors the VMs (4 marks)
- style of the monitor (2 marks)

GitLab

- README file (3 marks)
    - documentation of how to run each program and what that program does and does not do
- Template files (1 mark)

Part 2 Grading Scheme

The answer to the question posed in Part 2 should appear in a file in your A2 repo and should be named Part2.txt or Part2.pdf depending on its format. This will be graded outside of your Zoom grading session. It should not be any larger than 500 words. (3 marks)