Universidad Simón Bolívar.

Departamento de Electrónica y Circuitos.

EC5756- Redes Definidas por Software.

Estudiante: Raúl Guaidó

Carnet: 16-10486

Practica 3. Docker.

A su jefe en DeLab le gustó el inventario automatizado que extrae su script, y quiere que el CSV que genera su script este siempre actualizado, por lo que le ha pedido que ese script se ejecute en la plataforma docker en producción de la compañía.

1. Modifique el código de inventario de la asignación anterior para que la consulta se ejecute cada 5 minutos, y cada vez que se ejecute, actualice el archivo CSV generado. (Utilice el módulo "time" de python para el temporizador).

Se utiliza un while para crear un ciclo que mientras sea válido siga ejecutándose. Se define un clock que ira aumentando su valor en el tiempo, hasta llegar a los 5min y cuando eso suceda, el timer se le asignara el valor de clock y se ejecutara la solicitud, luego será necesario que el clock aumente su valor durante 5min para que pueda repetir la solicitud.

```
104  ###Funcion Principal(se repite cada 5 min)
105  timer = 0
106  while True:
107      clock = time.time()
108      if(clock - timer > 300):
109          timer = time.time()
110          orgname = obtname()
111          IDorg = obtid(orgname)
112          listdevice = obtdevice(IDorg)
113          print("Se ha actualizado la lista de dispositivos en la red")
114
```

2. Escriba un Dockerfile para crear un contenedor que ejecute el código python del punto anterior.

Para el Dockerfile se escogió la versión 3.10.5-alpine3.16 de Python, se realiza el copy en la dirección /usr/src/app/ y luego se instalan via pip los requerimientos necesarios, guardados en requirements.txt, se hace expose del puerto 3000 y se define el comando “python ./DeLab.py” para ejecutar el script. A continuación, se muestra el dockerfile:

```
Dockerfile > ...
1   FROM python:3.10.5-alpine3.16
2
3   COPY . /usr/src/app/
4
5   WORKDIR /usr/src/app
6
7   RUN pip install -r requirements.txt
8
9   EXPOSE 3000
10
11  CMD ["python", "./DeLab.py"]
```

3. Construya y ejecute el contenedor desde la línea de comandos.

Para ejecutar el contenedor se usa el comando “docker build -t namecontenedor .” es necesario incluir el punto al final para indicar que el archivo dockerfile se encuentra ubicado en la dirección donde se esta ubicado con el cd. El resultado de este comando es:

```
C:\Users\RG501\Documents\abril julio 22\redes definidas software\ProyectoDeLab>docker build -t dockerdelab .
[+] Building 35.7s (9/9) FINISHED
 => [internal] load build definition from Dockerfile                                                0.7s
 => => transferring dockerfile: 32B                                                                 0.0s
 => [internal] load .dockerignore                                                                   1.0s
 => => transferring context: 2B                                                                     0.0s
 => [internal] load metadata for docker.io/library/python:3.10.5-alpine3.16                         1.6s
 => [internal] load build context                                                                   0.6s
 => => transferring context: 5.30kB                                                                 0.2s
 => CACHED [1/4] FROM docker.io/library/python:3.10.5-alpine3.16@sha256:a746f64081fca7d6368935750ffcbf04d4  0.0s
 => => resolve docker.io/library/python:3.10.5-alpine3.16@sha256:a746f64081fca7d6368935750ffcbf04d447cb013  0.3s
 => [2/4] COPY . /usr/src/app/                                                                      1.2s
 => [3/4] WORKDIR /usr/src/app                                                                      1.4s
 => [4/4] RUN pip install -r requirements.txt                                                      26.3s
 => exporting to image                                                                              2.3s
 => => exporting layers                                                                             1.8s
 => => writing image sha256:be6e97d2ec8a5ccc329ee70a51d11501ab4f4e31b16b249cafba0b4b3e29cd03        0.1s
 => => naming to docker.io/library/dockerdelab                                                      0.1s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

Se verifica que el contenedor, con la imagen dockerlab fue creado, con el comando “docker ps -a”, teniendo el siguiente resultado:

```
C:\Users\RG501\Documents\abril julio 22\redes definidas software\ProyectoDeLab>docker ps -a
CONTAINER ID   IMAGE         COMMAND              CREATED         STATUS                       PORTS     NAMES
d876f8eb0b83   dockerdelab   "python ./DeLab.py"   8 minutes ago   Exited (130) 3 minutes ago             goofy_cohen

C:\Users\RG501\Documents\abril julio 22\redes definidas software\ProyectoDeLab>
```

4. Haga un nuevo commit de su código añadiendo el Dockerfile en el repositorio que creó en la asignación anterior.

Para hizo commit de las modificaciones realizadas al requirements, al script y al README con la información de la actualización, también se hizo el commit del nuevo archivo dockerfile y se hizo el push.

```
RG501@DESKTOP-ST05M5A MINGW64 ~/Documents/abril julio 22/redes definidas softwar
e/ProyectoDeLab (master)
$ git push -u origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 665 bytes | 166.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/rguaido50/Practica2DeLab.git
   7466b17..9d68408  master -> master
branch 'master' set up to track 'origin/master'.
```

5. Cree un volumen para su contenedor, en la dirección donde su código almacena el archivo CSV para hacer que este persista en memoria aun cuando se detenga el contenedor. (Modifique el Dockerfile de ser necesario)

Para crear el volumen no se modifico el dockerfile, ya que al realizar esto el volumen creado adquiere un nombre dado por Docker, por eso se utilizo el comando “run -v namevolume:direccciondelcsv –name namecontenedor -itd nameimagen”. Se puede verificar que le contenedor fue creado y el volumen usando el comando “docker ps” y “docker volumen ls” respectivamente. A continuación, se muestra el resultado de la creación del volumen y la verificación de que fue creado:

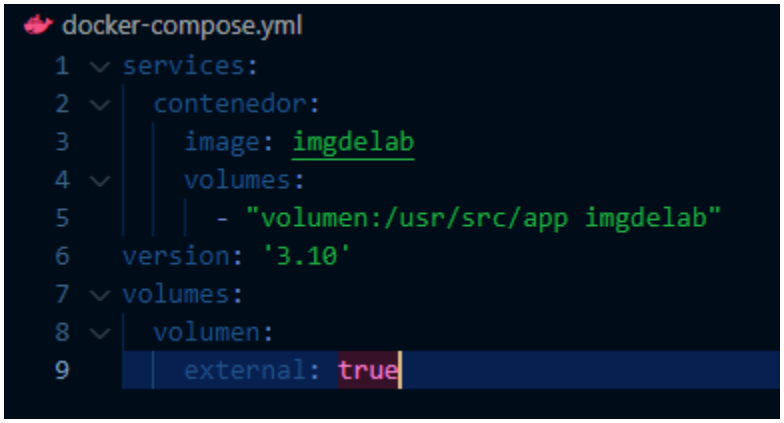
```
C:\Users\RG501\Documents\abril julio 22\redes definidas software\ProyectoDeLab>docker run -v volumen:/usr/src/app --name
contdelab -itd imgdelab
de5a5ea9fe0fabbe28baff13a98bf7dd944b8dfbb876459b6115ee376c48b495

C:\Users\RG501\Documents\abril julio 22\redes definidas software\ProyectoDeLab>docker ps
CONTAINER ID   IMAGE      COMMAND              CREATED         STATUS         PORTS      NAMES
de5a5ea9fe0f   imgdelab   "python ./DeLab.py"   8 minutes ago   Up 8 minutes   3000/tcp   contdelab

C:\Users\RG501\Documents\abril julio 22\redes definidas software\ProyectoDeLab>docker volume ls
DRIVER    VOLUME NAME
local     volumen
```

6. Construya un archivo docker-compose.yml a partir del cual se pueda iniciar su contenedor con todos los volúmenes necesarios.

Según la documentación en internet se especifico la imagen y el volumen con su dirección, el archivo Docker-compose.yml para este script es el siguiente:



7. Construya y ejecute su contenedor desde la línea de comandos usando el archivo docker compose que creó.

Se ejecutó el comando “docker’compose up --build” para construir el contenedor y ejecutarlo. Se probo la variación del comando anterior con el flag -d y ambos fueron realizados satisfactoriamente. Luego se verifico con el comando “docker ps -a” y “docker ps” para ver si fue creado y si estaba up, teniendo el siguiente resultado:

```
C:\Users\RG501\Documents\abril julio 22\redes definidas software\ProyectoDeLab>docker-compose up --build
Creating proyectodelab_contenedor_1 ... done
Attaching to proyectodelab_contenedor_1
Gracefully stopping... (press Ctrl+C again to force)
Stopping proyectodelab_contenedor_1 ... done

C:\Users\RG501\Documents\abril julio 22\redes definidas software\ProyectoDeLab>docker-compose up -d
Starting proyectodelab_contenedor_1 ... done

C:\Users\RG501\Documents\abril julio 22\redes definidas software\ProyectoDeLab>docker ps -a
CONTAINER ID   IMAGE      COMMAND              CREATED         STATUS          PORTS      NAMES
30f9808da165   imgdelab   "python ./DeLab.py"   8 minutes ago   Up 47 seconds   3000/tcp   proyectodelab_contenedor_1

C:\Users\RG501\Documents\abril julio 22\redes definidas software\ProyectoDeLab>docker ps
CONTAINER ID   IMAGE      COMMAND              CREATED         STATUS              PORTS      NAMES
30f9808da165   imgdelab   "python ./DeLab.py"   9 minutes ago   Up About a minute   3000/tcp   proyectodelab_contenedor_1

C:\Users\RG501\Documents\abril julio 22\redes definidas software\ProyectoDeLab>
```

8. Haga commit nuevamente a su repositorio, añadiendo el archivo docker-compose.

Se realizo el commit del archivo docker-compose.yml al repositorio de github, y se hizo push.

```
RG501@DESKTOP-STO5M5A MINGW64 ~/Documents/abril julio 22/redes definidas softwar
e/ProyectoDeLab (master)
$ git commit
[master 5a6d258] Agregado archivo docker-compose.yml
 1 file changed, 9 insertions(+)
 create mode 100644 docker-compose.yml

RG501@DESKTOP-STO5M5A MINGW64 ~/Documents/abril julio 22/redes definidas software/ProyectoDeLab (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

RG501@DESKTOP-STO5M5A MINGW64 ~/Documents/abril julio 22/redes definidas software/ProyectoDeLab (master)
$ git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 392 bytes | 196.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/rguaido50/Practica2DeLab.git
   9d6840d..5a6d258  master -> master
branch 'master' set up to track 'origin/master'.
```

El link donde se encuentra todos los archivos es:

https://github.com/rguaido50/Practica2DeLab