##Docker for DevOps Engineers – Day 18

Task 1:

1. **Learn how to use the docker-compose.yml file, to set up the environment, configure the services and links between different containers, and also to use environment variables in the docker-compose.yml file.**
   - **Docker-compose:**
     - It is a tool for defining and running multi-container docker applications.
     - It uses .yml file to configure application services (docker-compose.yml).
     - We can start all services with single command:

       docker-compose up

     - We can start all services with single command:

       docker-compose down

     - We can scale up selected services when required:

     docker-compose up -d –scale <app_name>=<no_of_replicas>

   - **YAML:**
     - It stands for Yet Another Markup Language.
     - It takes information in key-value pair.
     - YAML uses below components:
       - ♣ Key-value
       - ♣ Objects
       - ♣ List
       - ♣ List of objects
       - ♣ Multiline

   - **Step I:**

     Install docker-compose using one of below step:

     1. apt-get install docker-compose
     2. pip install docker-compose
     3. **$ curl -L https://github.com/docker/compose/releases/download/1.24.1/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose**

     Now give execute permission to docker-compose:

     $ chmod +x /usr/local/bin/docker-compose

Check if docker is installed by checking version:

docker-compose -v

- **Step II:**

  Now create docker-compose.yml file

```
version : "3.3"
services :
 web:
  image : nginx
      ports:
    - "8001:8001"
 database :
  image : mysql
  ports:
    - "3306:3306"
  environment :
   MYSQL_ROOT_PASSWORD : sample_password
```

- **Step III:**

  Now validate the docker-compose.yml file for syntax using below information:

docker-compose config

- **Step IV:**

  Run docker-compose.yml file to start the applications present in docker-compose file using below command:

docker-compose up -d

- **Step V:**

  Check the container is created or not.

  Now verify application if it is working or not.

- **Step VI:**

  If we want to take down the application, we can use below command:

docker-compose down

Task 2:

- **Pull a pre-existing Docker image from a public repository (e.g. Docker Hub) and run it on your local machine. Run the container as a non-root user. Make sure you reboot instance after giving permission to user.**

- **docker pull python**

```
$ docker pull python
Using default tag: latest
latest: Pulling from library/python
3e440a704568: Pull complete
68a71c865a2c: Pull complete
670730c27c2e: Pull complete
5a7a2c95f0f8: Pull complete
6d627e120214: Pull complete
f8c6dc678081: Pull complete
48bd2de548fc: Pull complete
e69bcee2d314: Pull complete
284a2f237609: Pull complete
Digest: sha256:08e538ee415a46998b19a6451da95d831f9e7e81be506925df51467b94e9cd43
Status: Downloaded newer image for python:latest
docker.io/library/python:latest
```

Now check if image is generated or not:

Docker images

```
$ docker images
REPOSITORY                    TAG       IMAGE ID       CREATED         SIZE
nginx                         latest    ac232364af84   20 hours ago    142MB
python                        latest    df3e9d105d6c   30 hours ago    921MB
mysql                         latest    483a8bc460a9   36 hours ago    530MB
```

Now add user permissions to docker:

sudo usermod -a -G docker $USER

Check if user permissions are reflected or not.

grep docker /etc/group

```
vagrant@vagrant:~$ grep docker /etc/group
docker:x:998:root,vagrant
```

Now run the container using below command:

docker run -it --name python_1 <image_id>

```
vagrant@vagrant:~$ docker run -it --name python_1 df3e9d105d6c
Python 3.11.2 (main, Mar 23 2023, 17:12:29) [GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exit
```

- **Inspect the container's running processes and exposed ports using the docker inspect command.**

docker inspect <cntr_id>

```
vagrant@vagrant:~$ docker inspect a076134138cc
[
    {
        "Id": "a076134138cc7d13c05078e18c6b804a663fcceb67d15a8c2e6f2be9fd456e69",
        "Created": "2023-03-24T12:49:43.477959817Z",
        "Path": "python3",
        "Args": [],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 30640,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2023-03-24T12:53:02.73901031Z",
            "FinishedAt": "2023-03-24T12:49:59.40187227Z"
        },
        "Image": "sha256:df3e9d105d6c5f8aa4410ba84b570db5f9fefbf14020b70fb63c8e7c32e51fb0",
        "ResolvConfPath": "/var/lib/docker/containers/a076134138cc7d13c05078e18c6b804a663fcceb67d15a8c2e6f2be9fd456e69/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/a076134138cc7d13c05078e18c6b804a663fcceb67d15a8c2e6f2be9fd456e69/hostname",
        "HostsPath": "/var/lib/docker/containers/a076134138cc7d13c05078e18c6b804a663fcceb67d15a8c2e6f2be9fd456e69/hosts",
```

- **Use the docker logs command to view the container's log output.**

docker logs <cntr_id>

```
vagrant@vagrant:~$ docker logs a076134138cc
Python 3.11.2 (main, Mar 23 2023, 17:12:29) [GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exit
Use exit() or Ctrl-D (i.e. EOF) to exit
>>>
Python 3.11.2 (main, Mar 23 2023, 17:12:29) [GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
vagrant@vagrant:~$
```

- **Use the docker stop and docker start commands to stop and start the container.**

docker start <cntr_id>

```
vagrant@vagrant:~$ docker start a076134138cc
a076134138cc
vagrant@vagrant:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED
a076134138cc   df3e9d105d6c                         "python3"                3 minutes ago
```

docker stop <cntr_id>

```
vagrant@vagrant:~$ docker stop a076134138cc
a076134138cc
```

- **Use the docker rm command to remove the container when you're done.**

Docker rm < cntr_id>

```
vagrant@vagrant:~$ docker rm a076134138cc
a076134138cc
```