Containerization using Docker

Assignment 1 – Working with Containers

Notes(s): NA

Task 1 (Installation/Verification)

1. Using standard installation procedure, install Docker on a Centos or Ubuntu VM.
2. Check the service status to ensure successful completion of installation and enable docker.
3. Run following command to validate:

```
root@server:~# docker -v

Docker version 20.10.7, build 20.10.7-0ubuntu1~20.04.1
```

Task 2 (Fundamental Operations – Working with images)

1. Check the list of docker images currently available locally in your docker system

   Note: Ideally there should be no images reflecting right now

2. Pull a latest hello-world image from docker registry and verify again list of images appearing locally.

   Note: Now at least one image for "hello-world" should be reflecting

3. For further practice, go to *hub.docker.com* search with some keywords and pull some popular images such as ubuntu, nginx, apache, etc. and list them locally using appropriate docker command.

Task 3 (Fundamental Operations – Working with Containers)

1. Check the list of docker containers running in the system currently.

   Ideally there should be zero containers running initially

2. Create a container using image “hello-world” naming it as “*myContainer*”

   Upon successful creation of container, “Hello from Docker” message should appear

3. List all containers i.e., running/stopped and validate if a container with name “*myContainer*” is appearing in the list which was created in the last step

4. Create a new container using either *nginx:latest* or *ubuntu:latest* image in detached mode.

5. Verify that the container you just created is in running state.

6. Check its networking details and capture the private IP address of the container.

Task 4 (Cleanup)

7. Delete all the image you have pulled.

8. Delete all the containers you have created in stopped/running state.

Note(s):

1. *You might get an Error saying the image is in use. In such cases, docker gives you option to delete the images forcefully as well.*
2. *You can also use docker system prune command.*