**Lab 3 – Virtualization**

1. After executing the command docker pulls the `hello-world` image from docker-hub, creates a container based on the image and then executes it. After executing `hello-world` as there is no persistent command the command exits and the container exits.
2. The command gets the list of containers that are running and shows it in a table, it includes its container id, image name, time created, status, ports.
3. This command shows all the docker containers even the closed ones in a similar list as above.
4. This command lists the images cached/pulled to create containers. The image tag associated with my hello-world container is ‘feb5d9fea6a5’.
5. In the command the 18.04 represents the image version of the docker-image. In case of ubuntu container images the docker-image versions correspond to the ubuntu release versions.
6. The images currently in cache are nginx:latest, ubuntu:18.04, hello-world:latest, ubuntu 19.04. It also shows the tag, image id, time created and size of the image.
7. The image ubuntu:18.04 was created at ‘2023-01-02T18:48:56.081327405Z’
8. The image ubuntu:19.04 was created at ‘2020-01-16T01:20:46.938732934Z’
9. After inspecting the images, we can conclude the following: -
   1. They have different repo tags
   2. They have different “DockerVersion”
   3. Both have no ports exposed
   4. Size of the ubuntu 19.04 is larger compared to 18.04
10. After running the container (ubuntu:18.04) a container is started using the image ubuntu:18.04. As the argument -d is provided the command runs in detached mode so I can use the terminal meanwhile the container runs in background (or ends)
11. It shows status and details of all running and exited containers (that exist)
12. I get a long string as output, one line, seems some kind of ID
13. The image has the name I gave “ubuntu1”
14. Hostname of the container is 8026ffc0e8aa it is taken from the container id
15. The command "$ docker container exec -it ubuntu1 cat hello-world.txt" and "$ docker container exec -it ubuntu2 cat hello-world.txt" runs the cat command inside the running container instances "ubuntu1" and "ubuntu2" respectively, which prints the contents of the file "hello-world.txt" if the file exists in the container's file system.
16. The command "$ docker container stop ubuntu2" stops the container instance "ubuntu2".
17. Using the command "$ docker container ls" shows all the running container instances, if there are any.
18. When we run "$ docker container ls -a" command it shows all the containers that have been created, used, and stopped, including the ones that are currently running.
19. After verifying the container is running again by using the "$docker container ls" command, we can see the status of the newly started/restarted container.
20. Running "$ docker container ls -a" shows all the remaining containers in Docker. The output of this command will show the container ID, image, command, created, and status of all containers.
21. All the other containers we worked with may have been deleted from Docker by using the "docker rm" command from our host machine.