**LABSHEET 2**

**DOCKER INSTALLATION AND BASIC DOCKER COMMANDS MANUAL**

1. Run Powershell as admin and check for wsl2-

a. type wsl and check if installed.

b. if not installed, check the wsl2 installation manual at <https://learn.microsoft.com/en-us/windows/wsl/install>

c. type wsl --install and install

d. wsl --set-default-version 2

e. wsl --list --verbose

f. to change linux distribution, type wsl --list --online

g. type wsl --install -d <Distribution Name> to install multiple distros

h. type wsl.exe to check the version after installing

i. type wsl.exe --status

j. type wsl.exe –l –v to check the wsl mode

k. wsl --set-default <distribution\_name>

l. wsl.exe -d Ubuntu-22.04 [to launch Ubuntu-22.04]

2. Install Docker Desktop from <https://docs.docker.com/desktop/install/windows-install/>

3. Create an account in Dockerhub

Basic Commands-

1. docker info
2. docker pull hello-world:latest
3. docker images
4. docker image COMMAND //*displays list of options to work with images*
5. docker ps
6. docker ps –a
7. docker run hello-world
8. docker inspect <container-id> //*to inspect about a container*
9. docker run --name hellocontainer hello-world //*to rename a container*
10. docker run –name <container-name> -i –t –d <image:tag> //*to run a container in interactive mode*

or

docker run - -name <container-name> -itd <image:tag>

docker run name <container-name> -it –d <image:tag>

*where -i 🡪to open image in interactive mode and allow us for standard input or interact with the command-line of the container*

*-t 🡪 provide an interactive terminal session*

*-d 🡪 detach( run containers in background and print an ID)*

1. docker exec -it <container-name> <image> // *to open the terminal*
2. exit() // *to exit from the container terminal*
3. docker run - -name <container-name> -i –t <image:tag> // *to directly open the interactive terminal session without exec command.*
4. docker restart python-c3 *//to restart an existing container*
5. docker rm <container-name>
6. docker rmi <image-id>
7. docker container prune //*delete all stopped containers from engine*
8. *EXPOSING AND PUBLISHING PORT*

docker run - -name nginx-c1 –d –p 8085:80 nginx //*binding host machine port to nginx server port(nginx runs in port 80)*

The same port can be used by apache web server if you run different container for a different server

docker run - -name apache-c1 –d –p 8000:80 httpd

1. docker run -d ubuntu sleep 100 // to keep a container for 100 secs

docker exec <container-id> cat /etc/\*release\* //to run a command in a running container

1. docker stop <container-name> //to stop a running container
2. docker start <container-name> //to start the same container and start working again
3. *PUSHING IMAGE TO DOCKERHUB*

docker tag <image-id> username/repository

docker login //press enter and provide both username and password of your dockerhub account.

docker push username/repository

1. *USING ENVIRONMENT VARIABLE AND ACCESS MYSQL SHELL*

docker pull mysql:latest

docker run - -name mysql-c1 -e MYSQL\_ROOT\_PASSWORD=<your-password> -d mysql:latest

docker exec -it mysql-c1 mysql -u root -p

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.

2. The Docker daemon pulled the "hello-world" image from the Docker Hub.

(amd64)

3. The Docker daemon created a new container from that image which runs the

executable that produces the output you are currently reading.

4. The Docker daemon streamed that output to the Docker client, which sent it

to your terminal.