RAMAIAH
Institute of Technology

OPERATING SYSTEM PROCEDURE REPORT FOR DOCKERIZING A PROJECT

Non CIE Component Report

Sept 2022 – Dec 2022

Semester 5

Submitted by

Shane George Shibu	1MS20CS108
Tejas BN Shetty	1MS20CS128
V.Aditya Raj	1MS20CS132
Monish Raman	1MS20CS133
Vishweshwara N	1MS20CS143

SUPERVISED BY

Dr. Dayananda R. B

ASSOCIATE PROFESSOR

DEPARTMENT OF CSE

RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute, Affiliated to VTU)

BANGALORE-560054

www.msrit.edu, December 2022

RAMAIAH
Institute of Technology

STEPS TO DOCKERIZE A WEB PROJECT:

1. We have used Linux software for our implementation and the commands used will be with respect to its terminal.

2. Firstly, we create a website folder on the desktop and import all the PHP, SQL, CSS files and other attachments required in the Database management project that was previously implemented.

3. We add a text file called “Dockerfile” with the following content within:
   FROM httpd
   COPY . /usr/local/apache2/htdocs/
   The first line of this Apache Dockerfile instructs Docker to pull the official httpd image from DockerHub.
   The second line instructs Docker to copy all of the files from a subfolder named /website/ and move them into the htdocs folder of the httpd Docker Image.

4. Then we go ahead by opening the terminal and enter the following commands in order:
   1. *sudo su:* We can execute instructions as another user using sudo without revealing your true identity. To use these restricted rights, you must have a line in /etc/sudoers. You enter an interactive session as root using sudo -i. Su stands for "switch to a specific user." Su changes to the root user immediately.

   2.*sudo root:* The true name of the administrator account is "root." Using the command "Sudo," regular users can carry out administrator functions. A user is not "Sudo." Long answer: The system administrator account is called "root" (sometimes known as "superuser").

   3. *cd Desktop/website/* :The command goes to website directory that we have created in the Desktop .

4. *snap install docker*: The command installs docker into the system so as to run all the docker commands and operations that need to be carried from here on. They update automatically and roll back gracefully. Snaps are discoverable and installable from the Snap Store, an app store with an audience of millions

5. *docker build -t website . :* A Dockerfile and a "context" are used by the docker build command to create Docker images. The collection of files in the PATH or URL that has been specified serves as a build's context. Any of the files in the context may be referred to by the build process.

6. *docker images:* A file called a Docker image is used to run programmes within a Docker container. Docker images serve as a template or collection of instructions for creating a Docker container. When utilising Docker, the starting point is also a set of images. A snapshot and an image are similar concepts in virtual machine (VM) settings.

7. *docker run -itd -p 8080:80 --name website website :* The command runs the runs the docker file having port '8080' to '80'and returns the token id of the created container .

8. *docker ps :* The command display the following details CONTAINER ID, IMAGE, COMMAND, CREATED, STATUS, PORTS and NAMES of all the containers present in the docker.

9. *docker login:* The command authenticates with the existing login credentials. The password will be stored unencrypted in /root/snap/docker/2285/.docker/config.json. Configure a credential helper to remove this warning.

10. *docker tag website <username>/website:latest:* An image name is made up of slash-separated name components, optionally prefixed by a registry hostname. The hostname must comply with standard DNS rules but may not

contain underscores. If a hostname is present, it may optionally be followed by a port number in the format:8080.

11. *docker login:* The command authenticates with the existing login credentials. The password will be stored unencrypted in /root/snap/docker/2285/.docker/config.json. Configure a credential helper to remove this warning. We use the command again as we now start pushing the project into the cloud from the next command.

12. *docker push <username>/website:latest:* We use docker image push to share your images to the Docker Hub registry or to a self-hosted one. It is in this command that we push our project into the docker's cloud.

5. Now got to the "Docker HUB" website and log in. Once, the login is complete we have to go to the repositories where we can find the project that we have pushed . Enter the public view from where we take the Docker Pull command :

   docker pull <username>/website

6. We take another system to pull the commands. Therefore we enter another the terminal and enter following commands:
   1. *sudo su*
   2. sudo *docker pull <username>/website*
   3. docker run -d -p 8080:80 <username>/website

7. Now we have to go to a browser and enter "localhost:8080" and we can see that the web project can now be run in the new system as well just by pulling the code from the docker cloud and how seamlessly the web project can be shared across devices.

RAMAIAH
Institute of Technology