**Steps to configure Docker**

OPERATING SYSTEMS PROJECT REPORT

**SUBMITTED BY**

|  |  |
| --- | --- |
| **Name: S D Hrudhay** | **USN: 1MS20CS098** |
| **Name: Safwan G A**  **Name: Sanskar R Gondkar** | **USN: 1MS20CS099**  **USN: 1MS20CS107** |
| **Name: Vivek J** | **USN: 1MS20CS144** |

As part of the Course **Operating systems– CS51**

SUPERVISED BY

Faculty

Dr. Dayananda R B

ASSOCIATE PROFESSOR

DEPARTMENT OF CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAMAIAH INSTITUTE OF TECHNOLOGY Sept 2022 – Dec 2022

Department of Computer Science and Engineering

Ramaiah Institute of Technology

(Autonomous Institute, Affiliated to VTU)

Bangalore – 54

**CERTIFICATE**

This is to certify that S D Hrudhay (1MS20CS098), Safwan G A (1MS20CS099), Sanskar R Gondkar (1MS20CS144), Vivek J(1MS20CS144) have completed the dockerization of our dbms project as part of Database Project. We declare that the entire content embodied in this B.E. 5th Semester report contents are not plagiarized.

Submitted by Guided by

Dr. Dayananda R B

(Dept of CSE, RIT)

(Associate Professor, Dept. of CSE, RIT)

Department of Computer Science and Engineering

|  |  |
| --- | --- |
| Name: S D Hrudhay | USN: 1MS20CS098 |
| Name: Safwan G A | USN: 1MS20CS099 |
| Name: Sanskar R Gondkar | USN: 1MS20CS107 |
| Name: Vivek J | USN: 1MS20CS144 |

Ramaiah Institute of Technology

(Autonomous Institute, Affiliated to VTU)

Bangalore – 54

**Procedure to dockerize the dbms project:**

Steps to dockerize your php and sql project:

* open a new folder 'project',
* create a folder called 'src' inside 'project' and copy all the contents of your project to src.
* in 'project' create 'Dockerfile' and paste the following:

FROM php:7.4-apache

RUN docker-php-ext-install mysqli

* in 'project' create 'docker-compose.yml' and paste the following:

version: "3.1"

services:

php:

build:

context: .

dockerfile: Dockerfile

ports:

- 80:80

volumes:

- ./src:/var/www/html/

db:

image: mysql

# NOTE: use of "mysql\_native\_password" is not recommended: <https://dev.mysql.com/doc/refman/8.0/en/upgrading-from-previous-> series.html#upgrade-caching-sha2-password

# (this is just an example, not intended to be a production configuration)

command: --default-authentication-plugin=mysql\_native\_password

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: example

adminer:

image: adminer

restart: always

ports:

- 8080:8080

* run it using following command:

$ docker-compose up -d

* run following command to see your container names and ports:

$ docker ps

* run following command to see your images and their tags:

$ docker ps -a

**Procedure to push images in docker hub:**

* save all your container instances into a new image using the below command:

$ docker commit <containerid> <newimagename>

* now to push images into your docker hub user login to your docker hub using the command:

$ docker login

Then enter your username and password.

If you are logged in successfully then the response status message will be shown as login succeeded.

* Then tag the image you want to push using the below command:

$docker tag <imagename>:<imageversion> <namespacename>/<imagename>

* Then push the image into your repository by using the command:

$ docker push <namespacename>/<imagename>

Follow the above commands mentioned to push all the images into your dockerhub account

**Procedure to pull the images from your docker hub:**

Make sure to login to your docker hub and stop all running instances of your containers in your environment and remove them and also remove the images so that they don’t collude with the images you want to pull. This can be satisfied using the commands mentioned below:

* command to stop all running containers:

$ docker stop $(docker ps -aq)

* command to remove all running containers:

$ docker rm $(docker ps -aq)

* command to remove all images:

$ docker rmi $(docker images -q) --force

* Then you can start pulling the images from your namespace of your docker hub by using the command:

$ docker pull <namespace>/<imagename>

Use the above command to pull all the images from your docker hub.

* Then run the container instances in your images using the command:

$ docker-compose up -d