National College of Ireland

Configuration Manual

MSc Research Project
MSc in Cloud Computing

Shubham Sanjay Tendulkar
Student ID: x20224753

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shubham Sanjay Tendulkar
Student ID:	x20224753
Programme:	MSc in Cloud Computing
Year:	2022
Module:	MSc Research Project
Supervisor:	Vikas Sahni
Submission Due Date:	15/08/2022
Project Title:	Configuration Manual
Word Count:	844
Page Count:	7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	19th September 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Shubham Sanjay Tendulkar
x20224753

1 Introduction

The following documents describes the steps of implementing the automated dockerization application proposed in the research paper. Python's ability to develop graphical user interfaces using Tkinter and Python's OS modules, which provide functions for interacting with the operating system, were utilized to create this application. This program aims to containerize web apps developed in popular languages such as ReactJS and ExpressJS. Additionally, this application simplifies the Docker-based WordPress instance creation process.

1.1 Pre-Requisites

To begin using the automated dockerization application, we must first configure the local environment. To create a proper environment for an application to run, we must first check the installed Python version using the command below. Preferably the machine must be running python version 3.10.* and above as shown.

```
python3 -V
Python 3.10.6
```

After checking the installed python version we must download and install Docker desktop on the machine. Docker desktop can be downloaded from `https://www.docker.com/products/docker-desktop/`. Before running the application, it has to be checked whether NumPy and Tkinter are installed. If they are not already installed, they can be installed easily with the following commands.

```
pip install numpy
pip install tk
```

2 Execution steps to launch application

Following are the steps to be followed for executing the application.

- Start Docker desktop.
- Extract the automated dockerization app's zip file in a folder.
- Browse to that folder.

- Open terminal in that directory and execute command "python main.py".
- This will provide a GUI of Automated Dockerization application as shown in figure1.
- In order to close the application user can just click on "close" button or press escape button on keyboard.

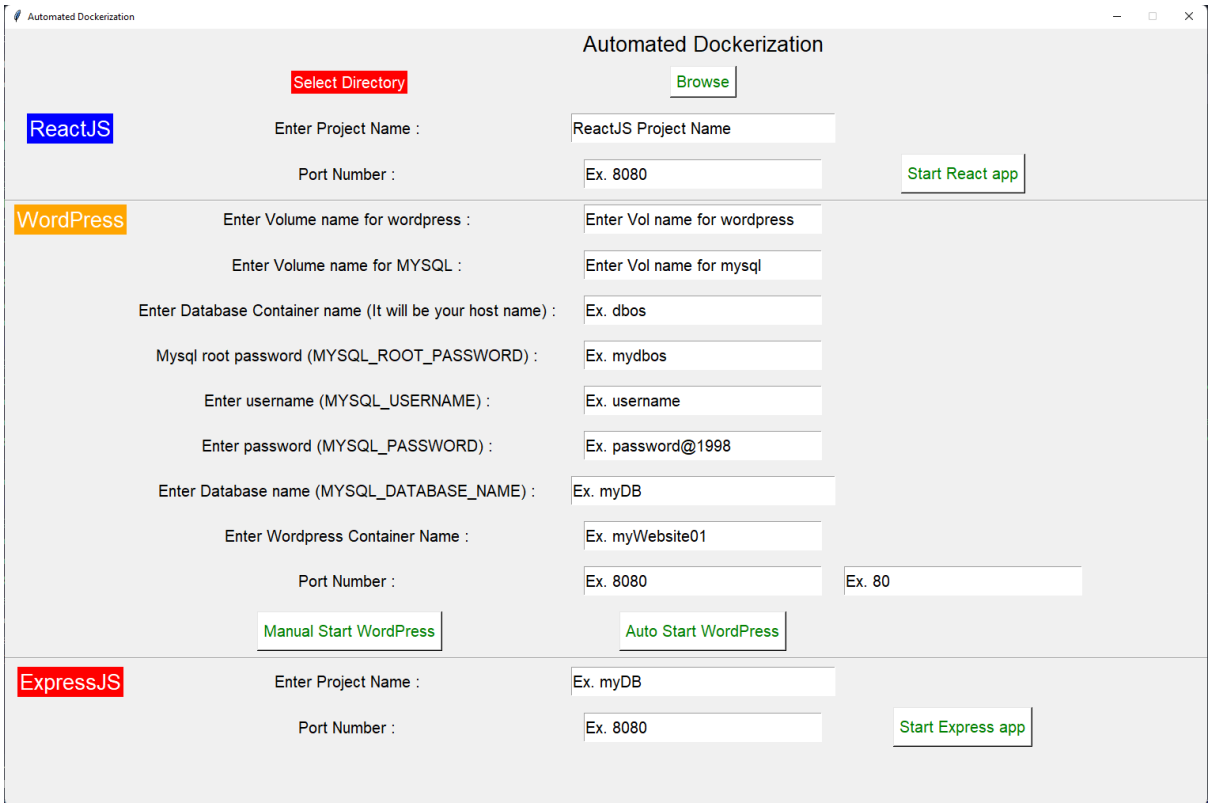

Figure 1: GUI of Automated Dockerization App

3 Execution steps to Dockerize ReactJS application

This section is a continuation of the second section, so it is expected that the application's graphical user interface is already opened.

- The initial step is to store a React application locally on the system.
- The second step is to locate the ReactJS project directory by choosing "Browse" button and selecting the directory.
- The user is required to provide the ReactJS application a name.
- Then the user will be required to provide the port number that should be exposed for the ReactJS application.
- Then the user should click on button "Start React app"
- This will dockerize the react app and open browser to the exposed port provided by user as shown in figure2.

4 Execution steps to create Wordpress instance on Docker

This section assumes that GUI of automated docekrization is already executed.

4.1 Manual Configuration of Wordpress

Following are the steps to create Wordpress instance on Docker manually

- Enter Volume name for wordpress
- Enter Volume name for MYSQL
- Enter Database Container name and it will also be your host name
- MYSQL root password
- Enter MYSQL username
- Enter MYSQL password
- Enter MYSQL Database name
- Enter Wordpress Container Name
- Enter Port Number to be exposed
- Click on button labelled "Manual Start WordPress"
- This will dockerize the WordPress and open browser to the exposed port provided by user as shown in figure3.

After filling in all details your input fields should like figure.

4.2 Automatic Configuration of Wordpress

Following are the steps to create Wordpress instance on Docker automatically by press of one button.

- The only step in this task is to click on button labelled "Auto Start WordPress"

After doing this step Wordpress instance on Docker will be created by using default given values as shown in figure.

5 Execution steps to Dockerize ExpressJS application

Since this section is a continuation of the second section, it is assumed that the graphical user interface of the application has already been launched before continuing.

- First step is to have a ExpressJS app stored in machine locally.

- Second step is to locate the directory of ExpressJS project by clicking on "browse" button and selecting directory.
- The user is required to specify a name for the ExpressJS application.
- Then the user will be required to provide the port number that should be exposed for the ExpressJS application.
- Then the user should click on button "Start Express app"
- This will dockerize the ExpressJS app and open browser to the exposed port provided by user.

6 Results

This section describes the many outcomes that can be produced by following the aforementioned processes.

6.1 Result of dockerizing ReactJS Application

In this Figure2 we can see the ReactJS running.

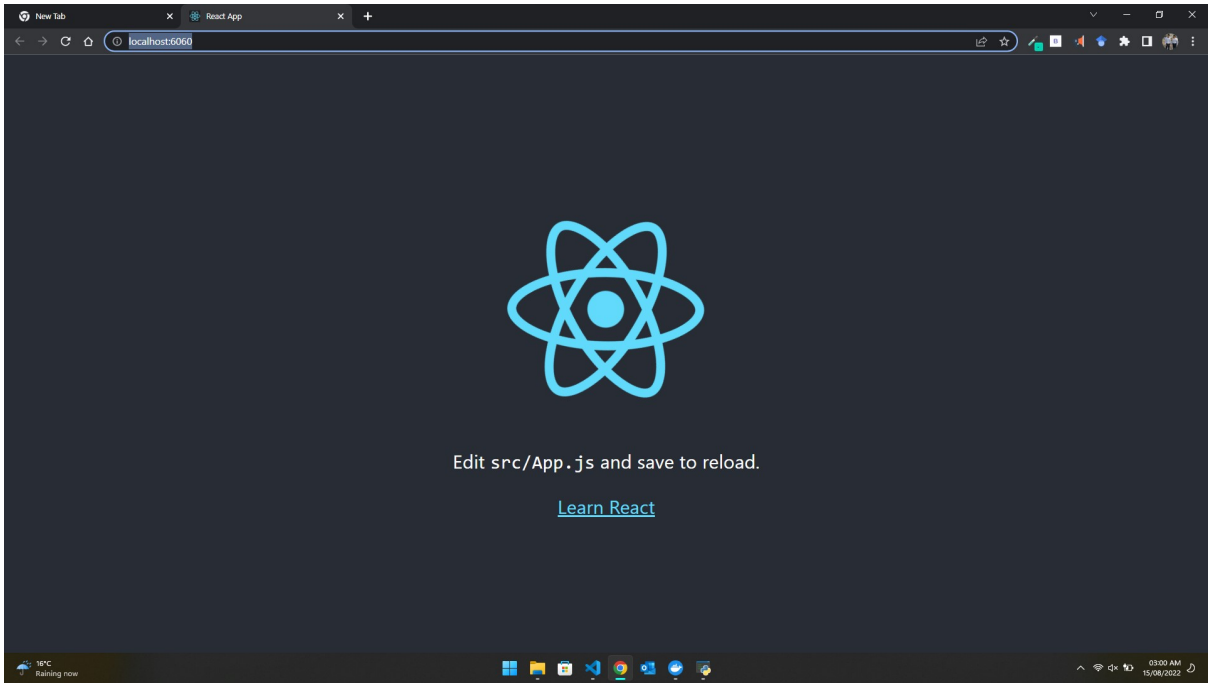

Figure 2: ReactJS app running on port 6060

6.2 Result of creating Wordpress instance on Docker manually

In this Figure3 we can see the WordPress running on port entered manually.

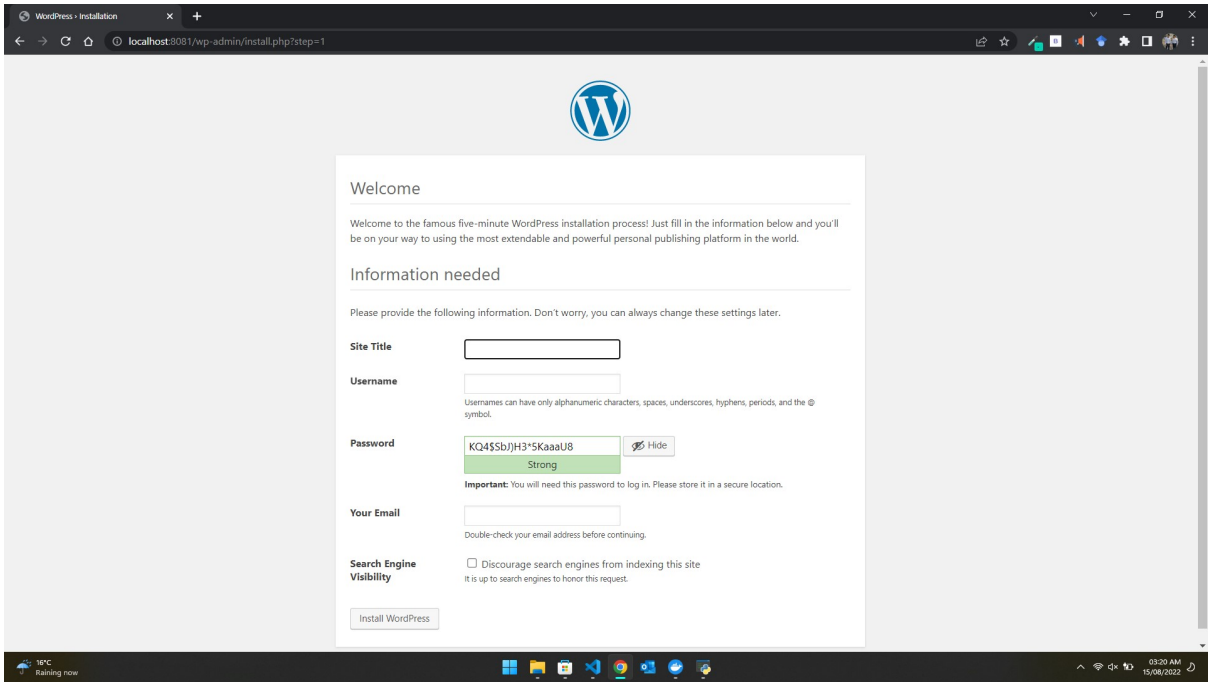

Figure 3: WordPress running on port set manually 8081

6.3 Result of creating Wordpress instance on Docker automatically

In this Figure4 we can see the WordPress running on default port.

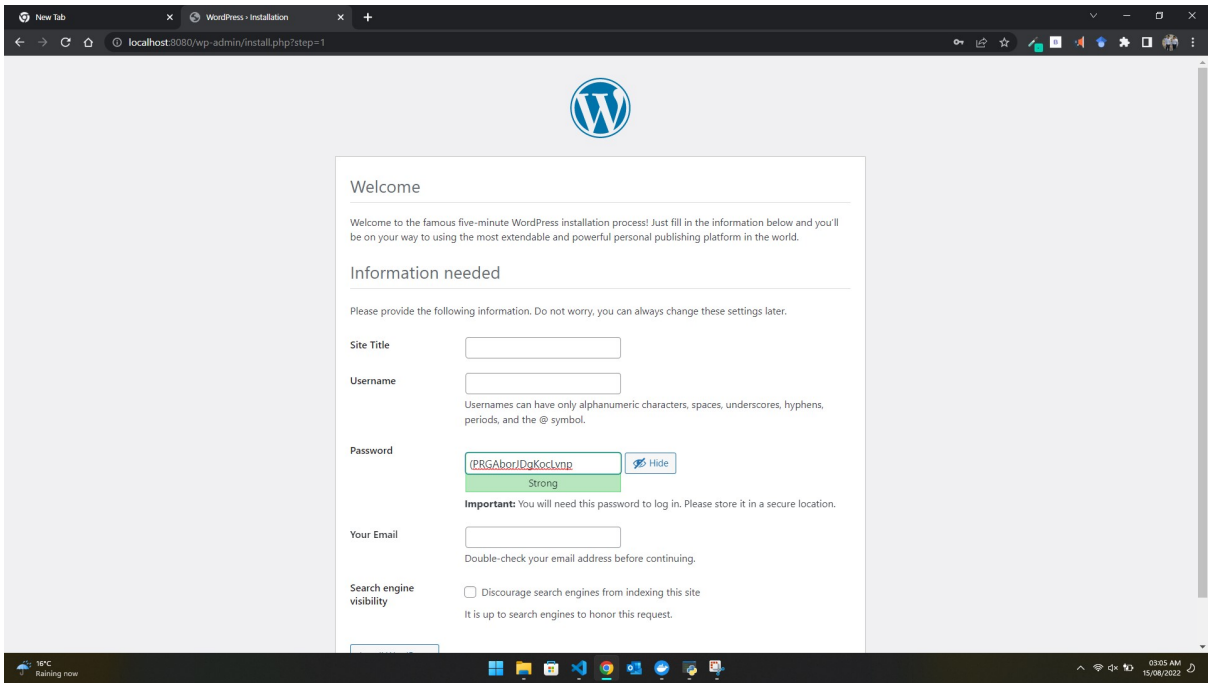

Figure 4: WordPress running on default port 8080

In this Figure we can see the WordPress running.

6.4 Result of dockerizing ExpressJS Application

In this Figure5 we can see the ExpressJS running on port 5050.

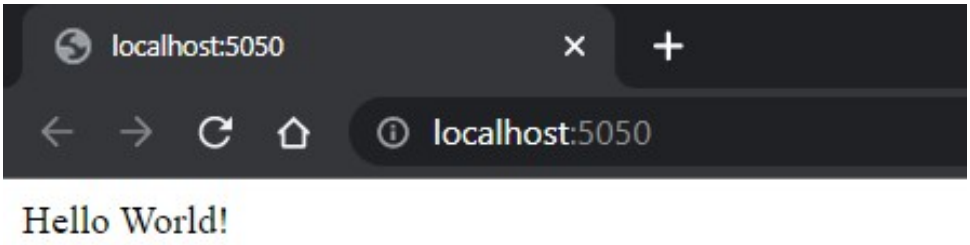

Figure 5: ExpressJS app running on port 5050

6.5 Docker Desktop Results

Following figures67 are the screenshots of Docker Desktop after completing all the mentioned tasks which shows all the containers and volumes created.

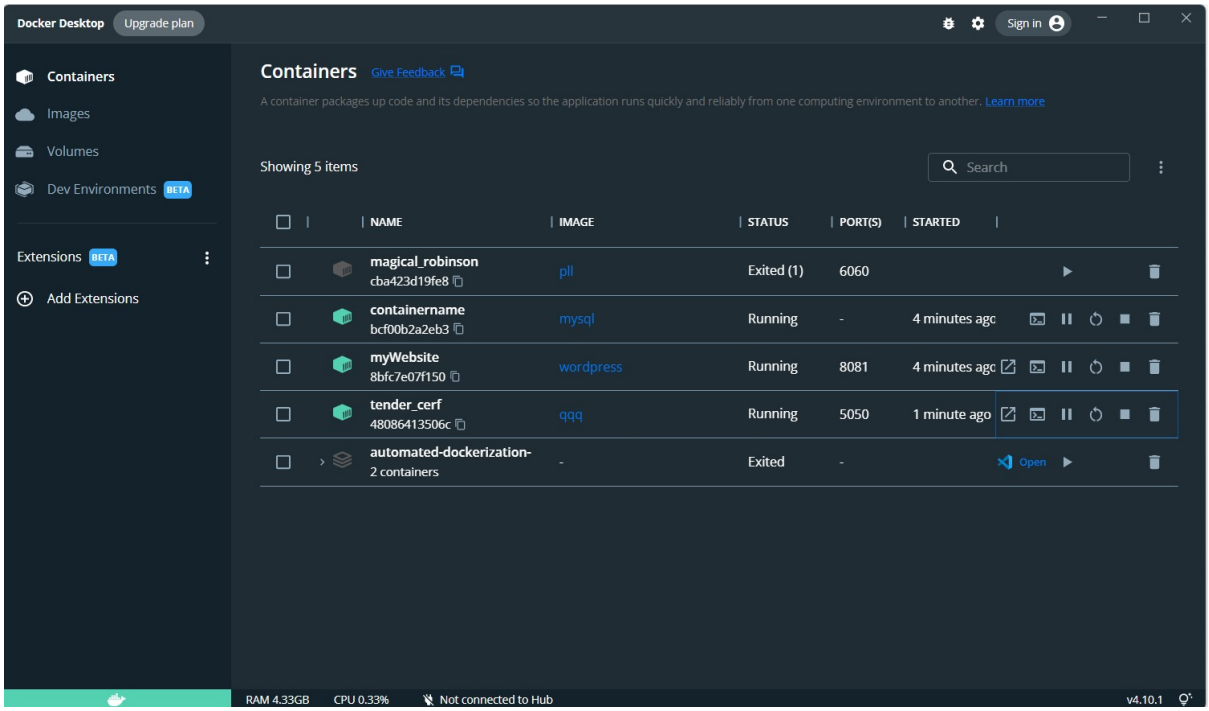

Figure 6: Docker Container

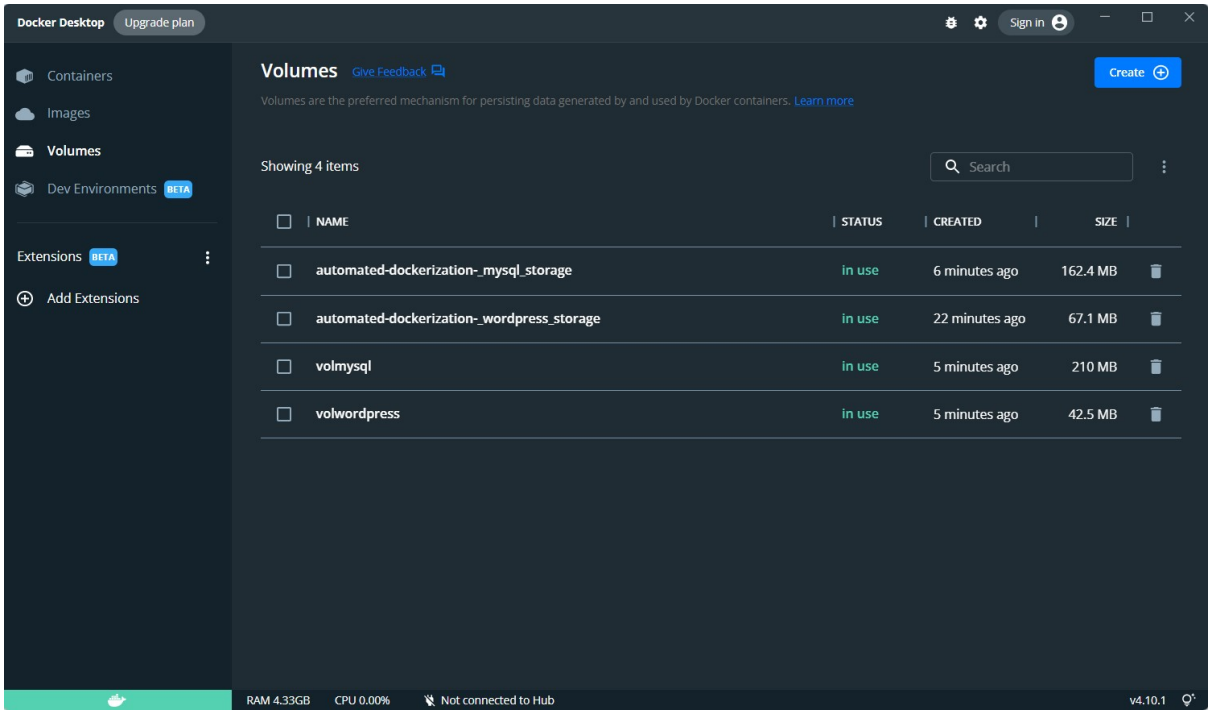

Figure 7: Docker Volumes