

# 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.

<u>ALL</u> 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<br/>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 figure 1.
- In order to close the application user can just click on "close" button or press escape button on keyboard.

Automated Dockerization	n			- 🗆 x
		Automated Dockerization		
	Select Directory	Browse		
ReactJS	Enter Project Name :	ReactJS Project Name		
	Port Number :	Ex. 8080	Start React app	
WordPress	Enter Volume name for wordpress :	Enter Vol name for wordpress		
	Enter Volume name for MYSQL :	Enter Vol name for mysql		
	Enter Database Container name (It will be your host name) :	Ex. dbos		
	Mysql root password (MYSQL_ROOT_PASSWORD) :	Ex. mydbos		
	Enter username (MYSQL_USERNAME) :	Ex. username		
	Enter password (MYSQL_PASSWORD) :	Ex. password@1998		
	Enter Database name (MYSQL_DATABASE_NAME) :	Ex. myDB		
	Enter Wordpress Container Name :	Ex. myWebsite01		
	Port Number :	Ex. 8080	Ex. 80	
	Manual Start WordPress	Auto Start WordPress		
ExpressJS	Enter Project Name :	Ex. myDB		
	Port Number :	Ex. 8080	Start Express app	

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 figure 2.

# 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 Figure 3 we can see the WordPress running on port entered manually.

WordPress - Installation X +			v – a x
← → C ☆ ③ localhost:8081/wp-admin/install.php?step=1			eੇ ☆ 👍 🖬 📌 🖬 🦣 i
	Welcome		
	welcome		
		s five-minute WordPress installation process! Just fill in the information below and you'll g the most extendable and powerful personal publishing platform in the world.	
	Information n	eeded	
	Please provide the follo	wing information. Don't worry, you can always change these settings later.	
	Site Title		
	Username	Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the © symbol.	
	Password	KQ4\$SbJ)H3*5KaaaU8 Ø Hide Strong	
		Important: You will need this password to log in. Please store it in a secure location.	
	Your Email		
		Double-check your email address before continuing.	
	Search Engine Visibility	Discourage search engines from indexing this site It is up to search engines to honor this request.	
	Install WordPress		
Jif's 16°C U Raining now		# 🖻 🖻 刘 🥥 💁 🐵 🗔	∧ ອຸd× 100 03-20 MM 15/08/2022 ປ

Figure 3: WordPress running on port set manually 8081

## 6.3 Result of creating Wordpress instance on Docker automatically

In this Figure 4 we can see the WordPress running on default port.

New Tab     X     WordPress - Installation	× +		v – 🖬 X
← → C ☆ ③ localhost:8080/wp-admin/install.php?step=1			🕶 🔄 🎓 👍 🗉 ı 🛊 🖬 👘 🗄
			_
	Welcome		
		s five-minute WordPress installation process! Just fill in the information below and you'll g the most extendable and powerful personal publishing platform in the world.	
	Information n	eeded	
	Please provide the follo	wing information. Do not worry, you can always change these settings later.	
	Site Title		
	Username	Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the $\circledast$ symbol.	
	Password	(PRGAborJDgKocLynp Ø Hide Strong	
	Your Email	Important: You will need this password to log in. Please store it in a secure location.           Double-check your email address before continuing.	
	Search engine visibility	<ul> <li>Discourage search engines from indexing this site</li> <li>It is up to search engines to honor this request.</li> </ul>	
Raining now		🏭 🗮 🛱 刘 🧕 💁 🤗 🍯 🖶	へ

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 Figure 5 we can see the ExpressJS running on port 5050.



Figure 5: ExpressJS app running on port 5050

#### 6.5 Docker Desktop Results

Following figures 67 are the screenshots of Docker Desktop after completing all the mentioned tasks which shows all the containers and volumes created.

Docker Desktop Upgrade plan					🛎 🌣 Sig	in in 🔒		-	o x
Containers  Images	Containers Give Feedback 🖳 A container packages up code and its dependencies so								
Volumes     Dev Environments	Showing 5 items				Q Search				
		IMAGE	STATUS	PORT(S)	STARTED				
Extensions BETA	magical_robinson cba423d19fe8 @		Exited (1)	6060			•		Î
Add Extensions	Containername bcf00b2a2eb3 🗇		Running		4 minutes agc		l ¢		Î
	□		Running	8081	4 minutes agc 🖄		I ¢		Î
	□		Running	5050	1 minute ago [/	2	I ¢		Î
	$\Box \longrightarrow \bigotimes \begin{array}{c} \text{automated-dockerization-} \\ 2 \text{ containers} \end{array}$		Exited				•		Î
	RAM 4.33GB CPU 0.33% 🐧 Not connected to H								1.10.1 Q <sup>*</sup>

Figure 6: Docker Container

Docker Desktop Upgrade plan			👪 🌣 Sign in	θ -	o x`
<ul><li>Containers</li><li>Images</li></ul>	Volumes Give Feedback  Q Volumes are the preferred mechanism for persisting data generated by and used by Docker containers. Learn r			Create	:
Volumes     Dev Environments	Showing 4 items		Q Search		:
Extensions BETA		STATUS	CREATED	SIZE	
<ul> <li>⊕ Add Extensions</li> </ul>	automated-dockerizationmysql_storage	in use	6 minutes ago	162.4 MB	Î
	automated-dockerizationwordpress_storage  volmysql	in use in use	22 minutes ago 5 minutes ago	67.1 MB	Î 
	volwordpress	in use	5 minutes ago	42.5 MB	•
۵	RAM 4.33GB CPU 0.00% 🕅 Not connected to Hub			v4	.10.1 Q <sup>°.</sup>

Figure 7: Docker Volumes