# System Requirements Specification Index

**For**

# Docker Usecase

## Http webserver in docker

## 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

**Usecase on Automating Apache HTTP Server Deployment and Testing using docker**

**Introduction:**

ShopEase, a mid-sized e-commerce company, is preparing for its annual seasonal sale. With a dramatic increase in website traffic expected, the company must ensure its web servers can scale seamlessly and operate reliably. The DevOps team is tasked with automating the deployment process for Apache HTTP servers and verifying their functionality through an efficient, repeatable pipeline. This automation is essential to support the anticipated surge and to guarantee zero downtime during the critical sales period.

Background:

In previous seasonal sales, the lack of a standardized deployment process led to inconsistencies across servers, resulting in avoidable outages and loss of revenue. To resolve this, the DevOps team decided to leverage Docker containers to create a consistent, portable server environment. Using automated testing scripts, the team aims to ensure all servers are operational before being deployed to handle live traffic.

This use case illustrates how the automation of web server deployment and testing transforms the reliability and efficiency of ShopEase's infrastructure, providing a robust solution to meet high traffic demands during peak periods. By integrating Docker, Python-based tests, and version control via GitHub, the team ensures a smooth and collaborative development process.

**Objective:**

To automate the deployment of an Apache HTTP Server using Docker containers and verify its functionality through Python-based test scripts.

**Pre-requisites:**

1. **Docker** installed on your machine.
2. **Python 3.x** installed along with docker and requests libraries.
3. **Git** installed and configured.
4. Access to a GitHub repository for version control.

---

Steps to execute the usescase

1 . Open cmd check if  docker is installed on the system docker –version

2 docker pull httpd:latest

- **Run a Docker Container:**
- docker run -d --name httpd-container -p 8081:80 httpd
  - docker ps is used to check the container is running in the environment
  - Verify the container by visiting http://localhost:8081.

---

**Part 2: Customize the http websever**

**Place a custom index.html file in the same directory.**

**To edit the container file use the command docker exec -it http_container bash**

- Navigate to htdocs folder and edit the index.html
- In the index html add

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Webserver Launch</title>
    <script>
        window.onload = function() {
            document.getElementById("message").innerText = "The webserver launched!";
        }
    </script>
</head>
<body>
    <h1 id="message">Loading...</h1>
</body>
</html>
```

Save and exit the container and run the testcase to run the testcase check the information below

**Expected Outcome:**

1. Successful deployment of an Apache HTTP Server in a Docker container.
2. Verification of server functionality through automated tests.
3. Code and results are version-controlled and shared on GitHub.

---

**Task List to be completed**

1. Check if docker is installed in the virtual machine ?
2. Create docker images in the virtual environment ?
3. Create webserver using apache server and create a sample website

4. Check whether the application is hosted in the correct ports
5. Launch the application
6. Run the testcase to check if the deployment in successful

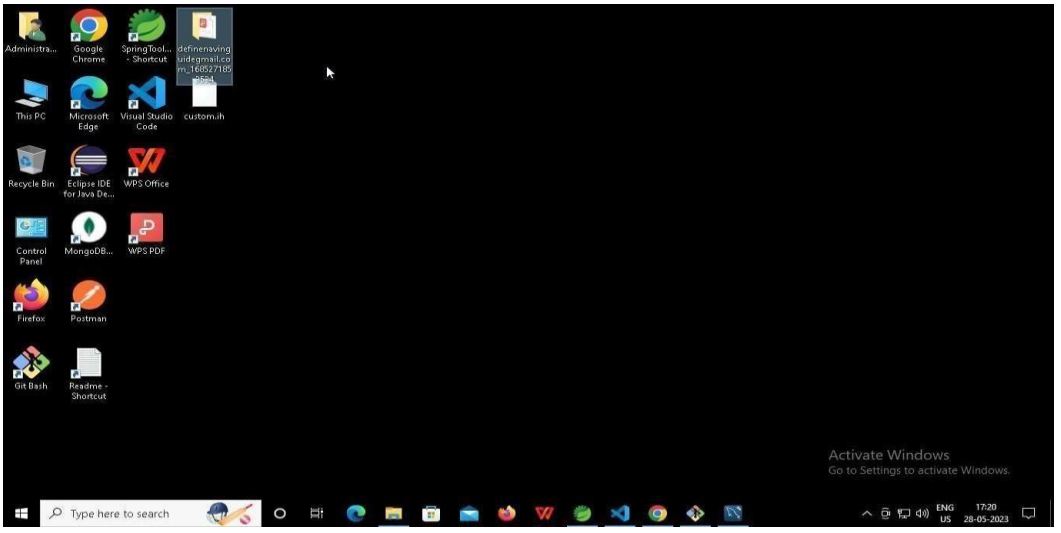
## Run the code and submit the code

- **Goto the project folder and in URL open cmd then run the python testcase code**
- **To Run the testcase run the python file in the project folder using the command using the command python run_tests.py**
- After the testcase is completed you push the code to the repository

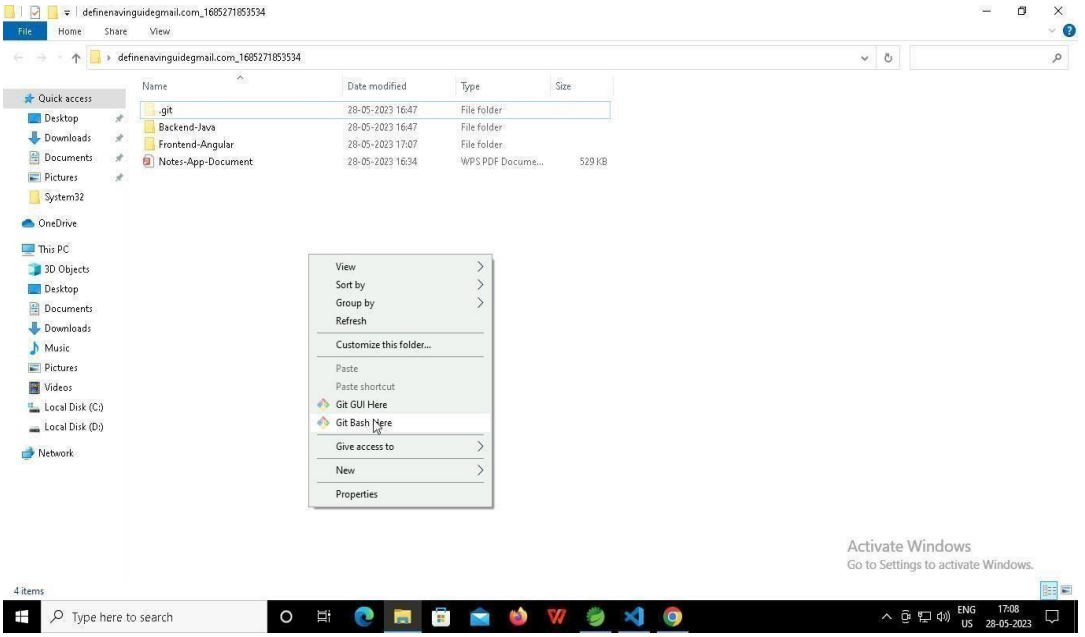
## Push the code the repository

**You can run test cases as many numbers of times and at any stage of Development, to check how many test cases are passed/failed and accordingly refactor your code.**

1. **Make sure before final submission you commit all changes to git**. For that open the project folder available on desktop

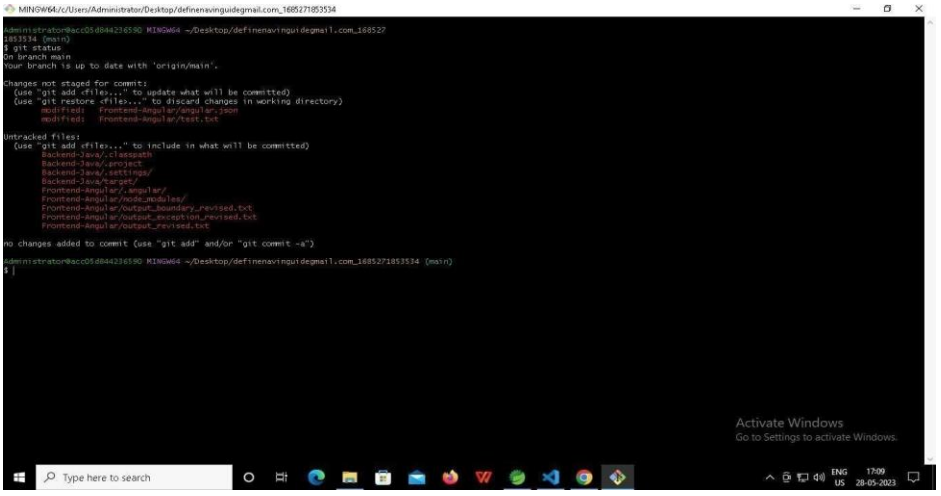



   a. Right click in folder and open Git Bash



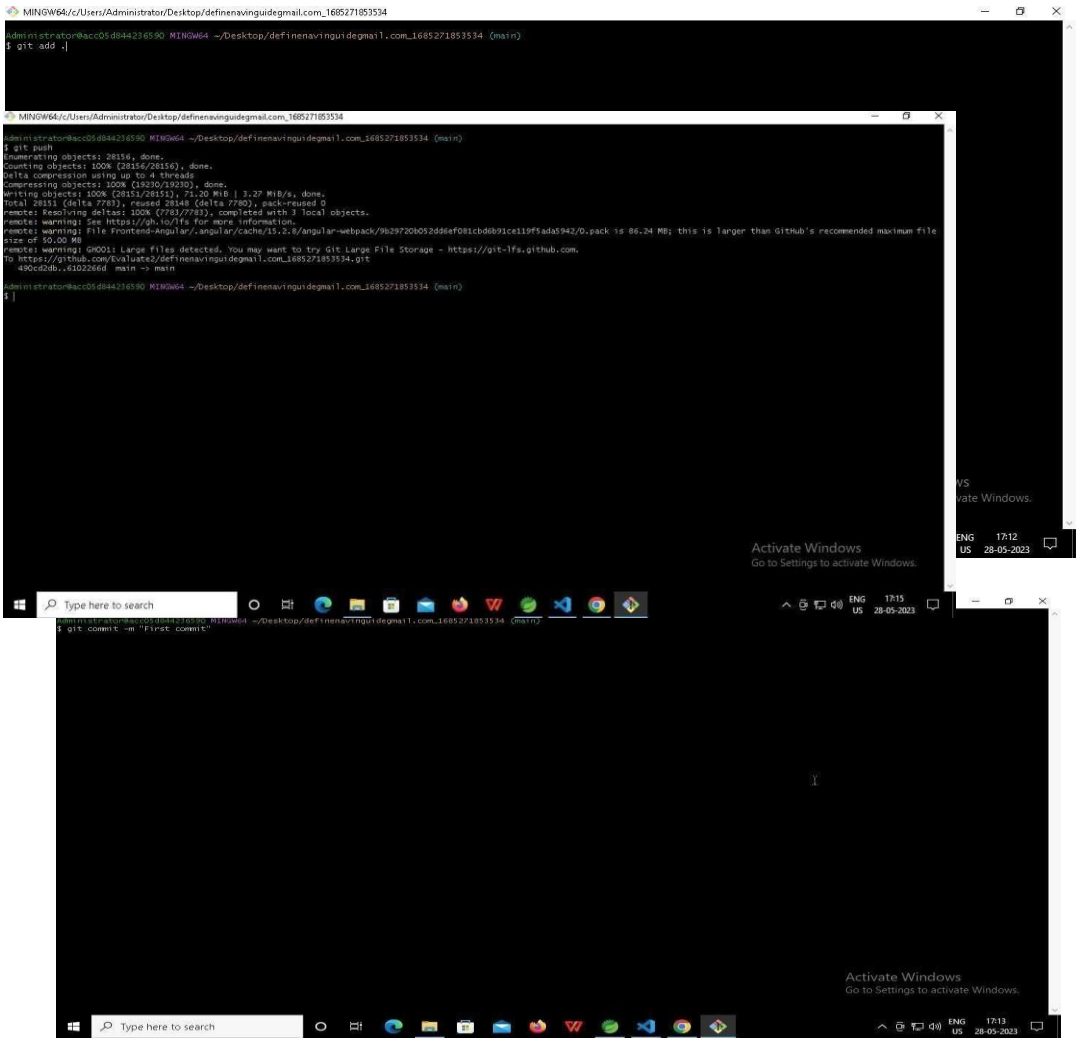

b. In Git bash terminal, run following commands

c. git status




d. git add .

e. git commit -m “First commit”
(You can provide any message every time you commit)


MINGW64:/c/Users/Administrator/Desktop/definenavinguidegmail.com_1685271853534
Administrator@acc05d844236590 MINGW64 ~/Desktop/definenavinguidegmail.com_1685271853534 (main)
$ git add .
MINGW64:/c/Users/Administrator/Desktop/definenavinguidegmail.com_1685271853534
Administrator@acc05d844236590 MINGW64 ~/Desktop/definenavinguidegmail.com_1685271853534 (main)
$ git push
Enumerating objects: 28156, done.
Counting objects: 100% (28156/28156), done.
Delta compression using up to 4 threads
Compressing objects: 100% (19230/19230), done.
Writing objects: 100% (28151/28151), 71.20 MiB | 3.27 MiB/s, done.
Total 28151 (delta 7781), reused 28148 (delta 7780), pack-reused 0
remote: Resolving deltas: 100% (7783/7783), completed with 3 local objects.
remote: warning: See https://gh.io/lfs for more information.
remote: warning: File Frontend-Angular/.angular/cache/15.2.8/angular-webpack/9b29720b052dd6ef081cbd6b91ce119f5ada5942/0.pack is 86.24 MB; this is larger than GitHub's recommended maximum file size of 50.00 MB
remote: warning: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.
To https://github.com/Evaluate2/definenavinguidegmail.com_1685271853534.git
490cd2db..6102266d main -> main
Administrator@acc05d844236590 MINGW64 ~/Desktop/definenavinguidegmail.com_1685271853534 (main)
$
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG US 17:15 28-05-2023
MINGW64:/c/Users/Administrator/Desktop/definenavinguidegmail.com_1685271853534
Administrator@acc05d844236590 MINGW64 ~/Desktop/definenavinguidegmail.com_1685271853534 (main)
$ git commit -m "First commit"
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG US 17:13 28-05-2023

F .git push

-----x-----