**Student Name: Lanka Venkata Sai sreeharsha**

**Student ID: 20000272**

B9MG119

SME: Copita Tapas Bar

**Cloud Strategy and Sample Deployment**

**Cloud Technology for Business**

Table of Contents

[1 Introduction 2](#_Toc139921260)

[2 Enterprise background 3](#_Toc139921261)

[3 Current IT Setup 4](#_Toc139921262)

[4 Deployment Procedure 5](#_Toc139921263)

[5 Recommendations 10](#_Toc139921264)

[5.1 Cloud vs non-cloud solutions 10](#_Toc139921265)

[5.2 Justification for Recommendation 11](#_Toc139921266)

[6 Conclusion 12](#_Toc139921267)

[7 References 13](#_Toc139921268)

# Introduction

The study contains the exploration of cloud-based technology for the solution of small and medium-scale businesses. There are three main platforms used for the cloud-based solution which are Microsoft Azure, Docker, and GitHub. We push the code of the web application in the form of images on all the above platforms that we created for the Copita tapa bar company. This document is the reflection of the procedure and recommendation.

# Enterprise background

Copita is a part of tapa bar that culture is motivated by Spain. The reason behind the consideration of this company is to motivate the food business with the potential of innovative technology. The Copita is located in London, United Kingdom and they operate in the Takeaway part. The bar was opened in the year of 2011 by delivering tasty tapas. They change their menu every week on the basis of the seasonal change strategy. The company is associated with top-notch category ingredients because they aim for superb taste (Copita 2020).

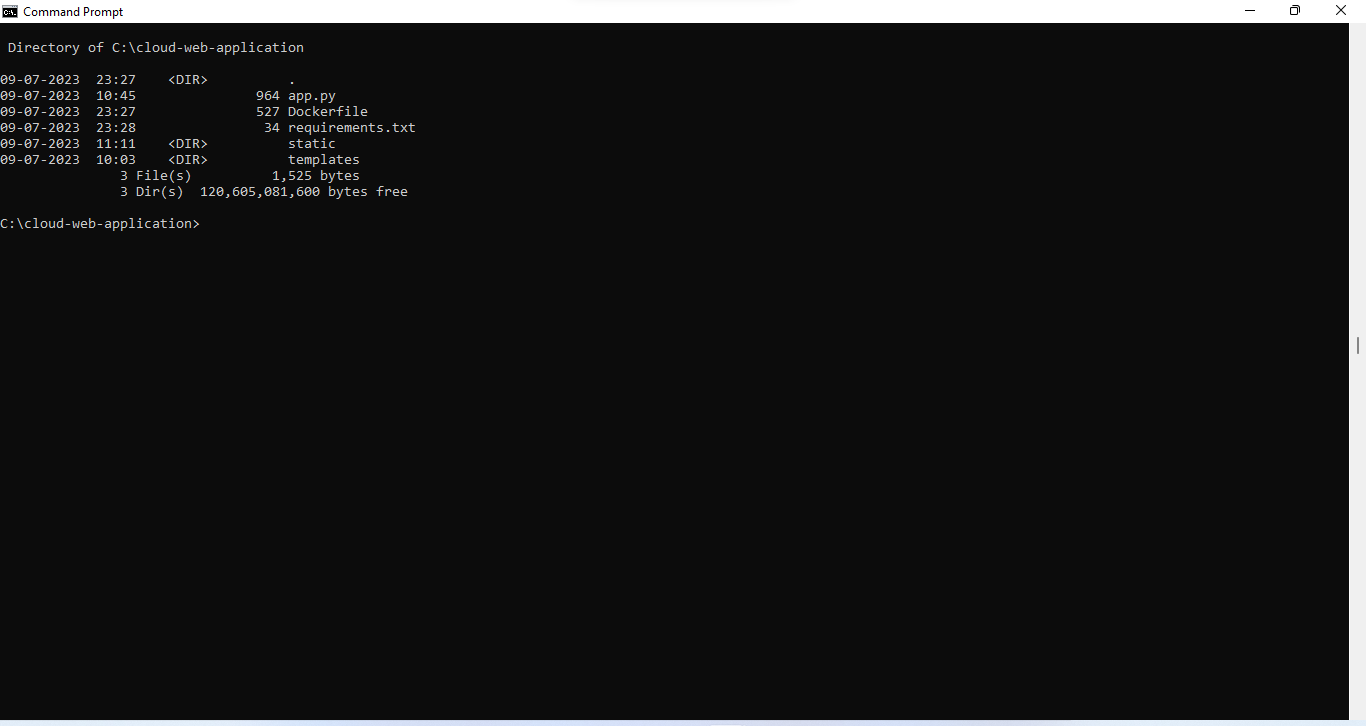
# Current IT Setup

The company is associated with different IT setup and the major part is its website. The company also works on digital marketing in order to promote its products and business. With the help of the aspects of digital marketing, the company can enable a wider range of clients because the use of the Internet is spreading widely. As the company engages with more customers than the data comes into action. The data can be used for various purposes, especially for the effective strategy of the company. with the help of data analysis, companies can identify their positive and negative points which is helpful for identifying target customers. For the development of the backend of the website, the company is associated with the Java script. The company can integrate the program with advanced technology so various options can be enabled in the form of API. Currently, the company is associated with multiple APIs. After the integration of Django or Flask, all the functionalities can be performed with a single code (GeeksforGeeks 2020). As the code length decreases then the speed and durability of the website increases.

# Deployment Procedure

**Application Directory structure**:

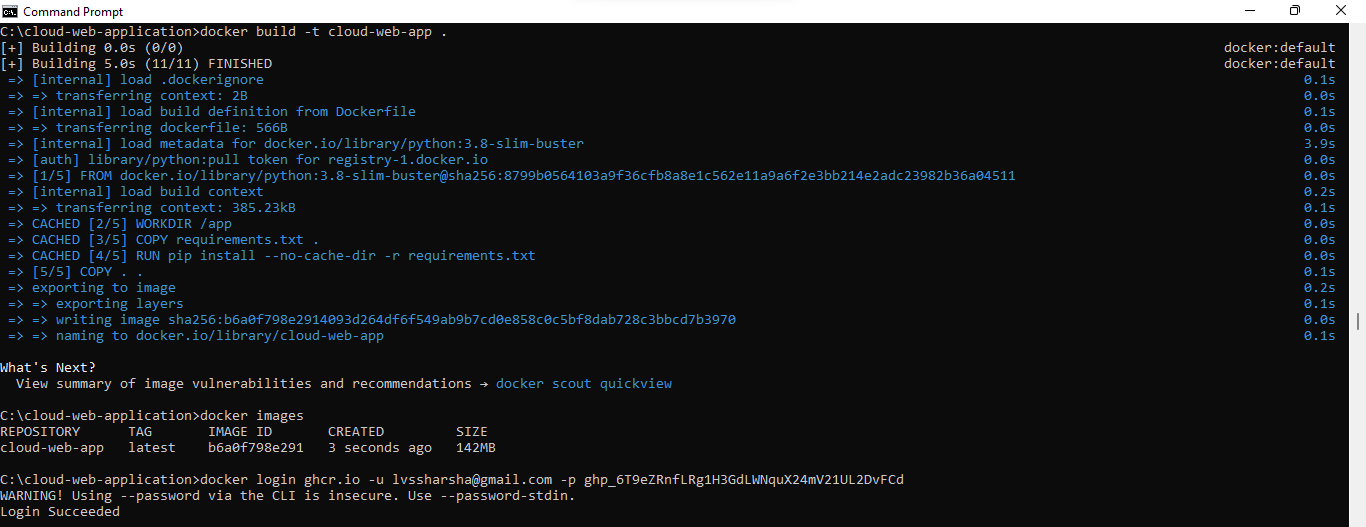
The directory structure should be setup as per according to the given image which includes the source code and all the required files along with the Docker file to create and containerize the image.



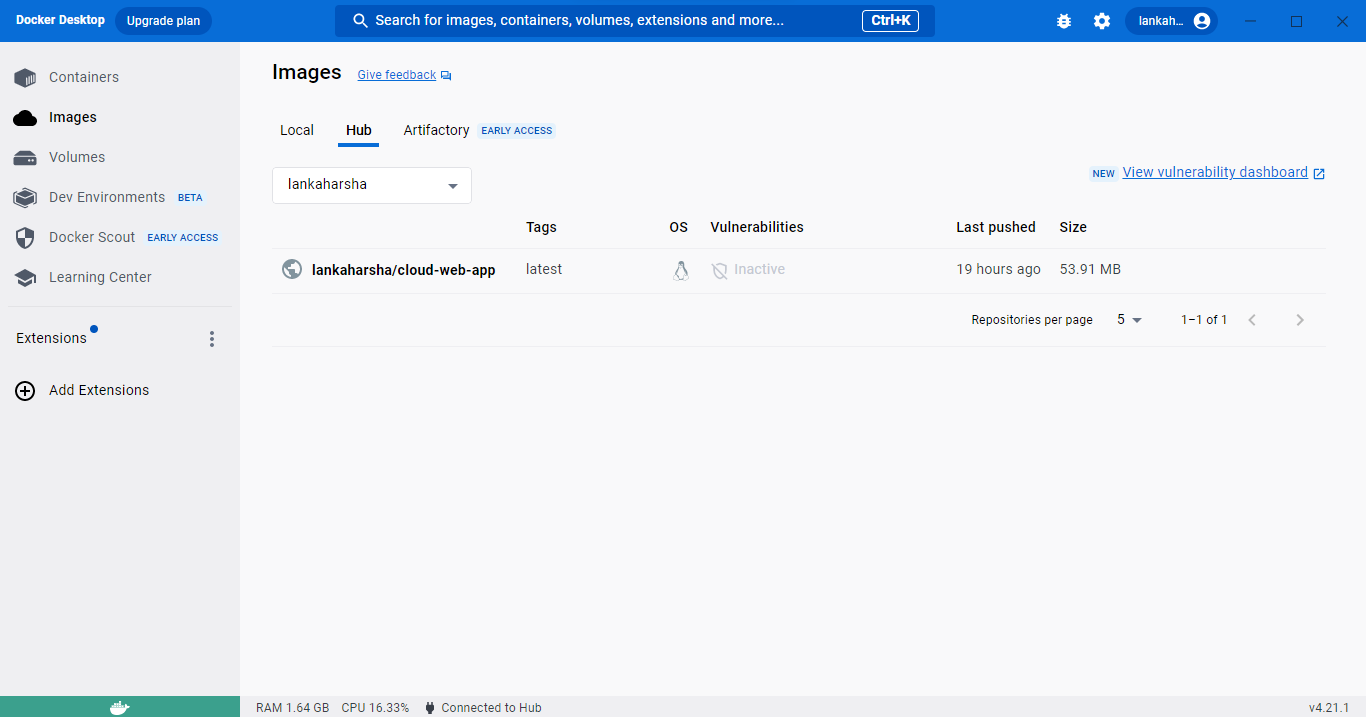
**Creation of Docker image:**

In order to containerize the application, we have to create a docker image using the docker commands as shown in the image below.

Once the image is built, we can check the images present in the local (Docker 2022).



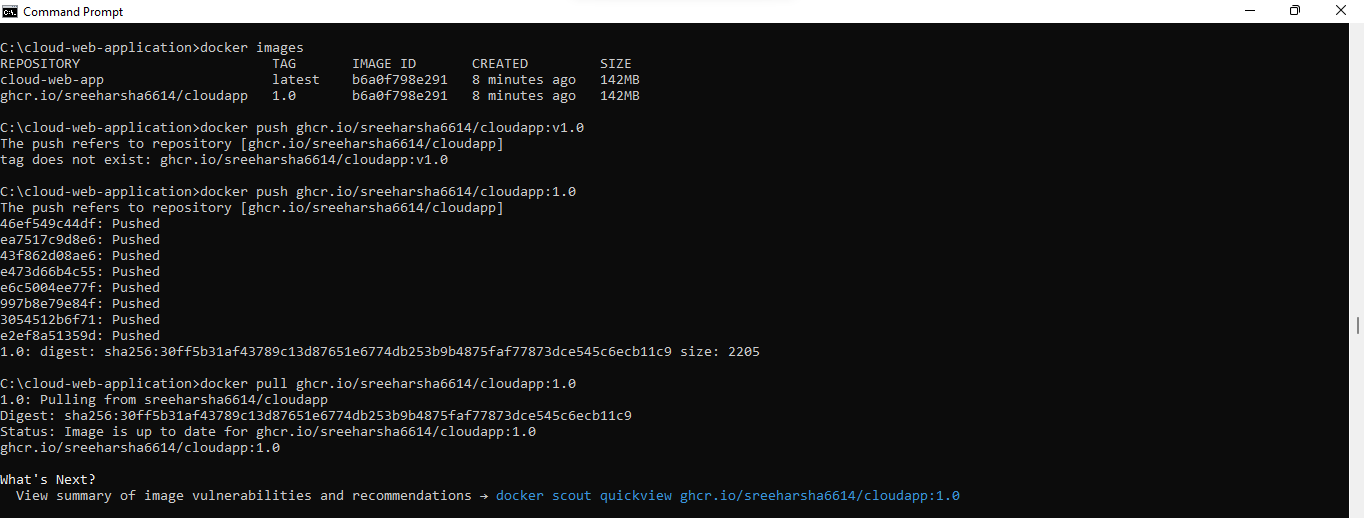
The image is also pushed to docker hub in order to run the web application through the docker desktop.

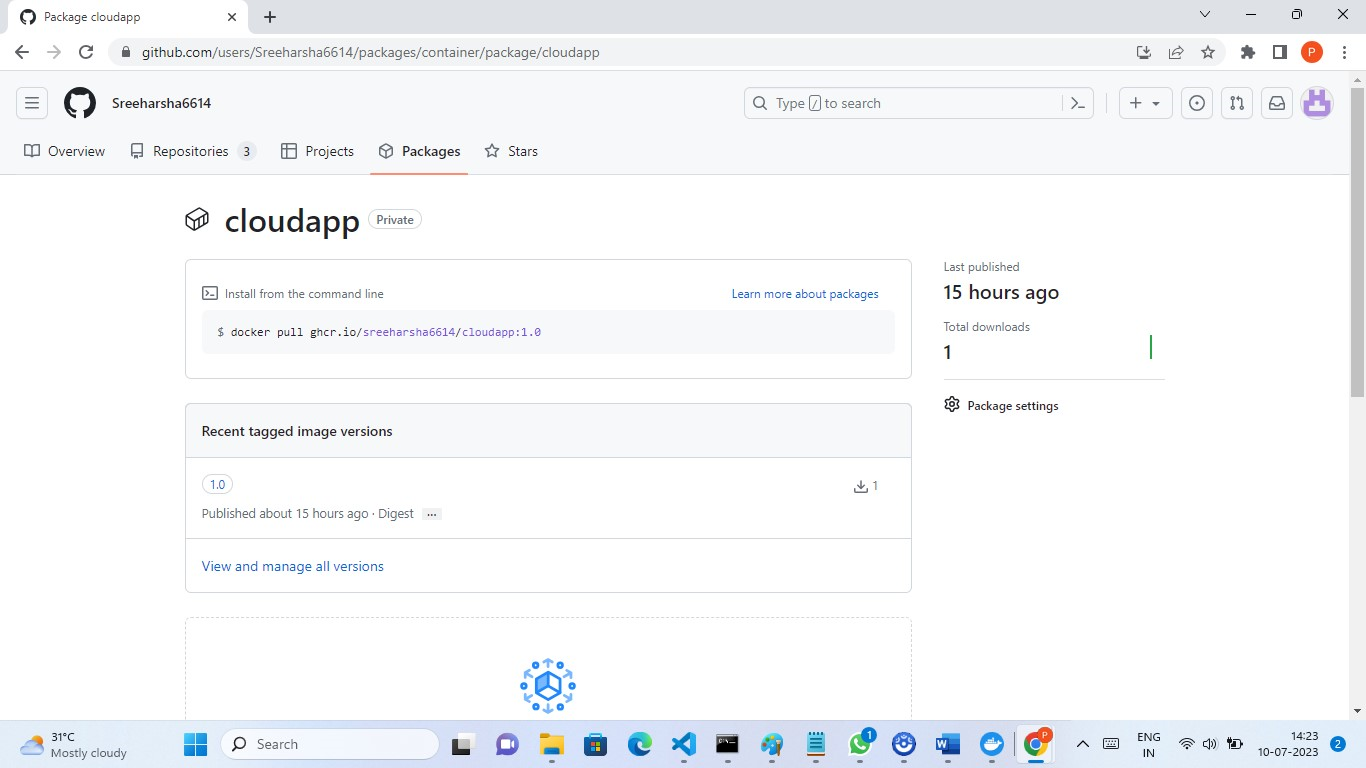


**Pushing the Image to GitHub:**

Now the created docker image is pushed to GitHub registry using the below steps:

* Tag the docker image to the GitHub registry
* Push the docker image to the GitHub as shown in the image below.
* Verify the image pushed in the GitHub using the GitHub portal.

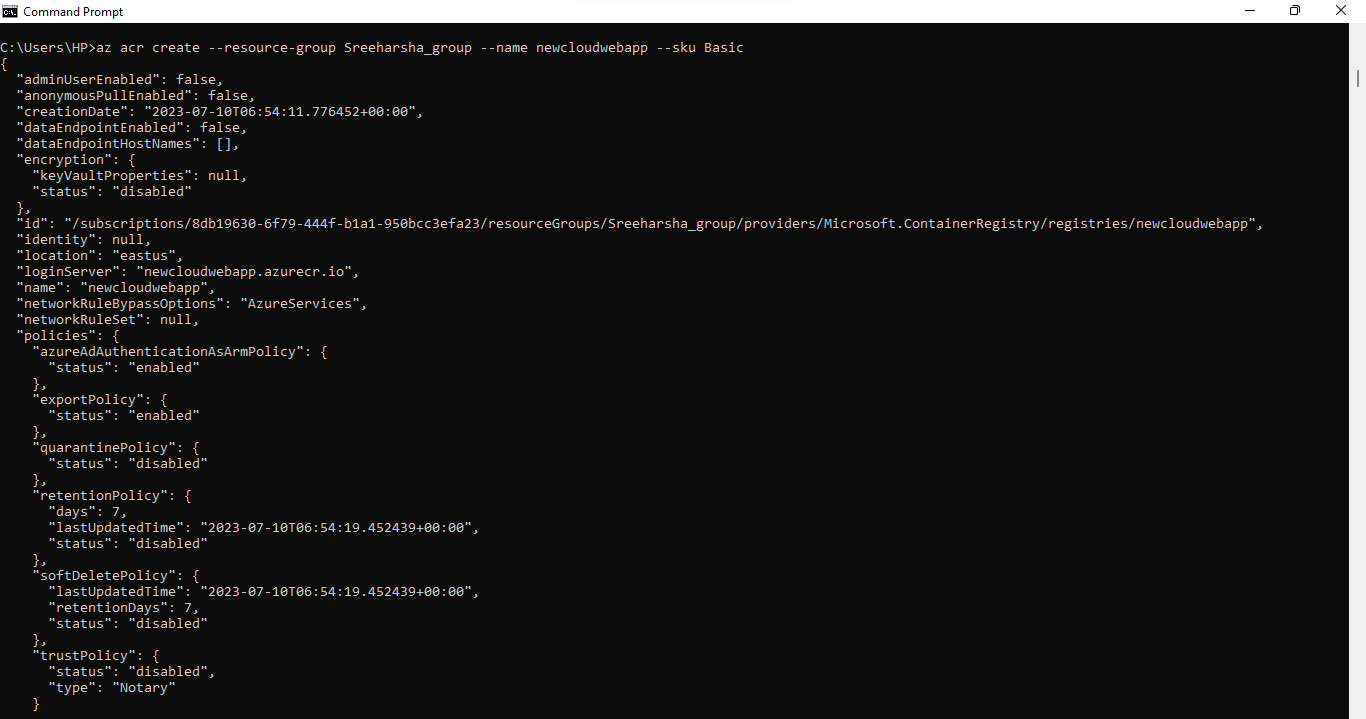


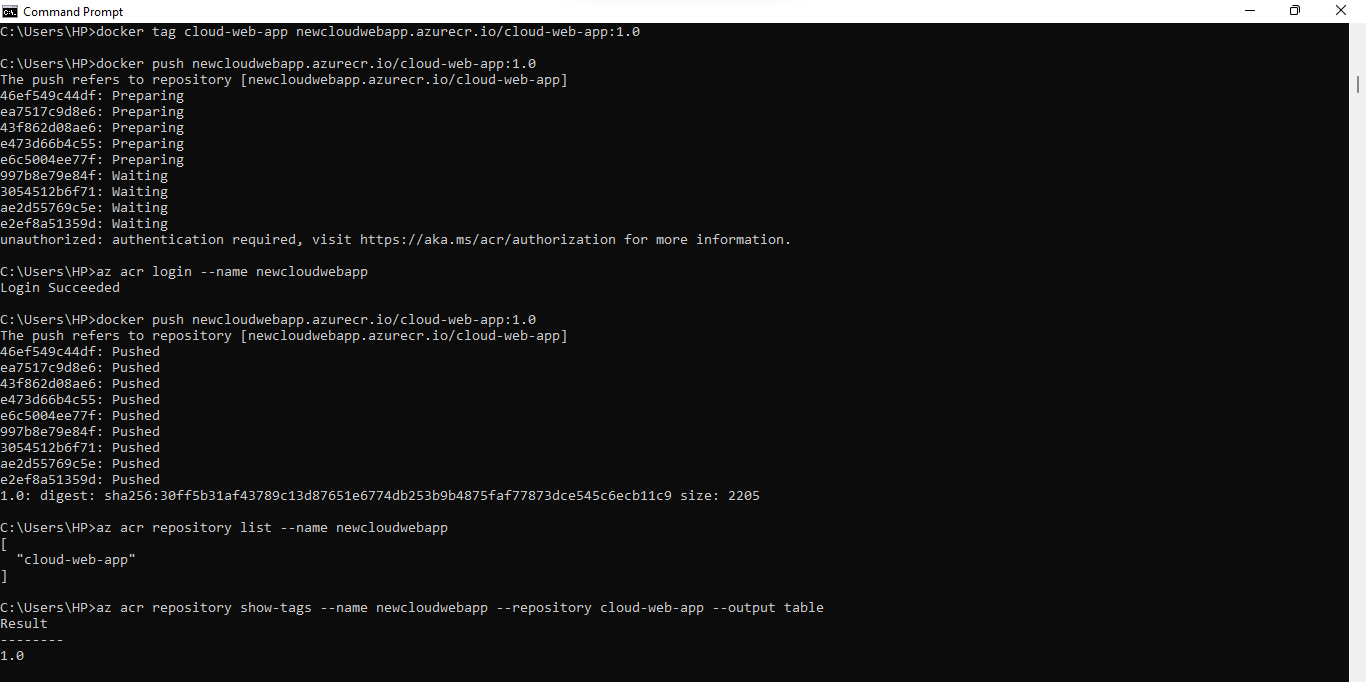


**Uploading to azure as container:**

The image is uploaded to azure as follows:

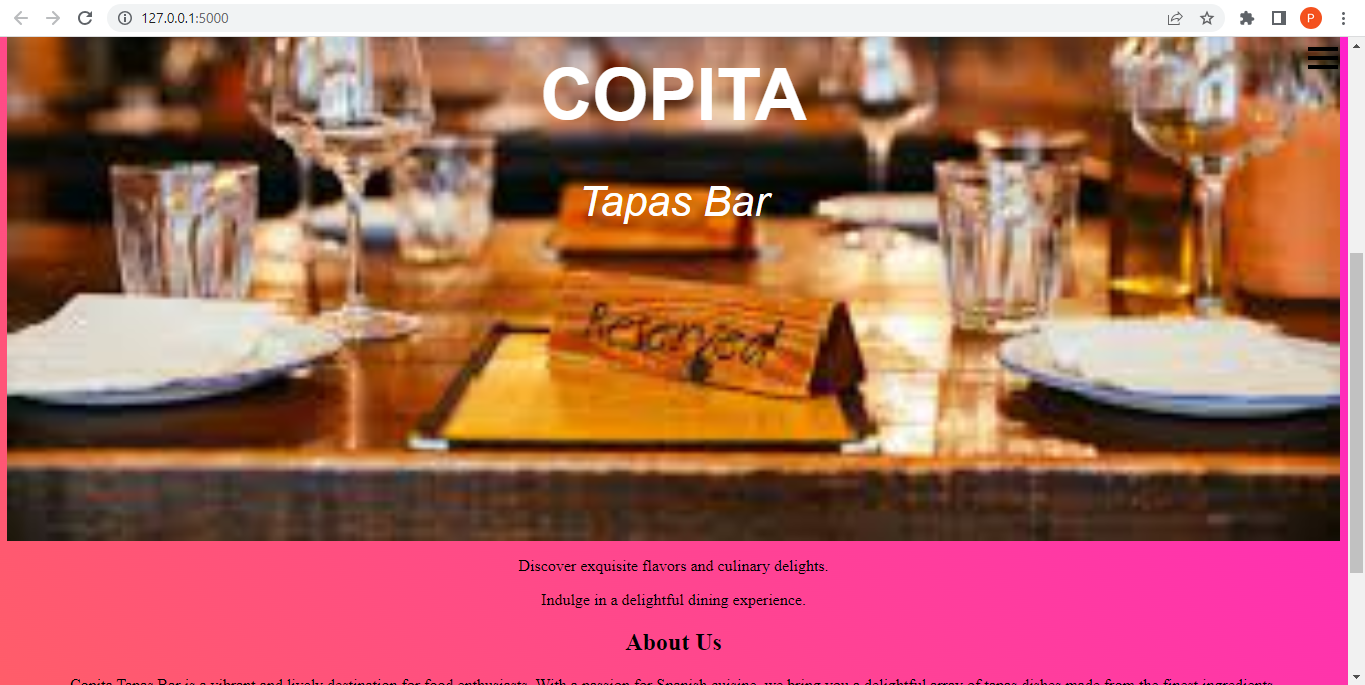
* Create the azure container registry using the resource group
* Tag the docker image to azure container registry.
* Login to azure container registry.
* Push the image to azure container registry.

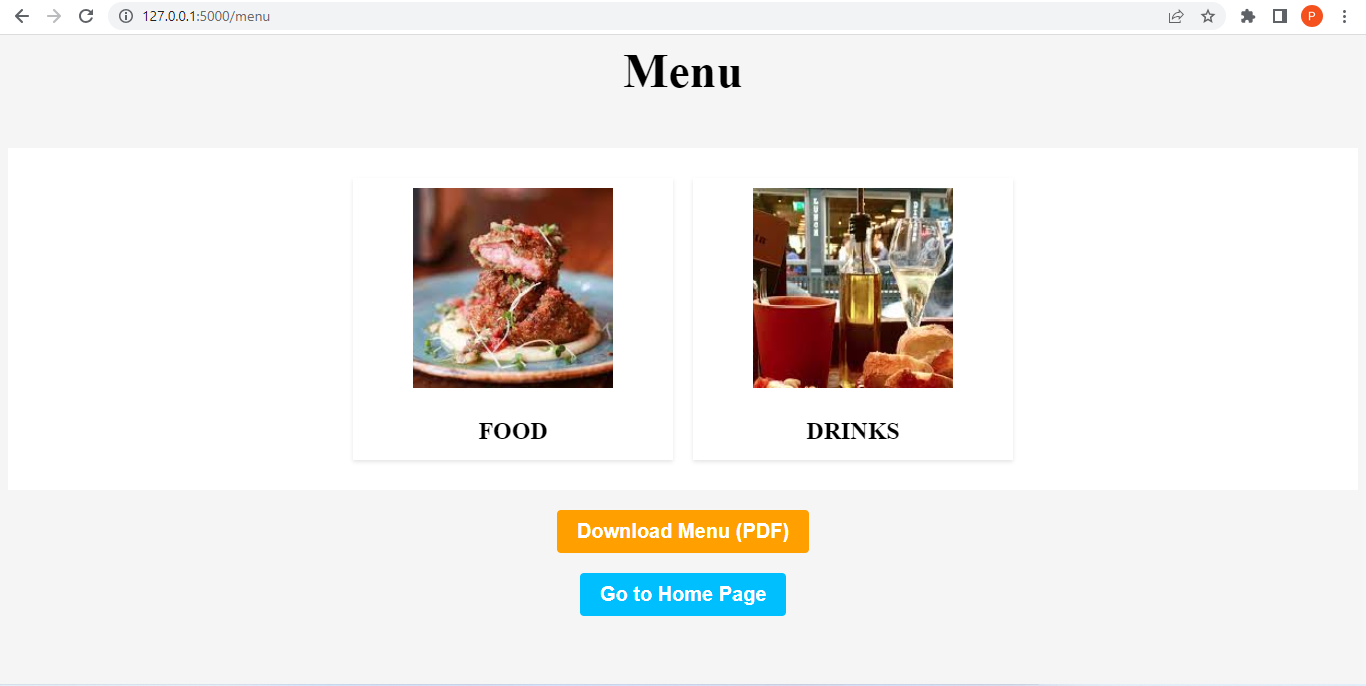


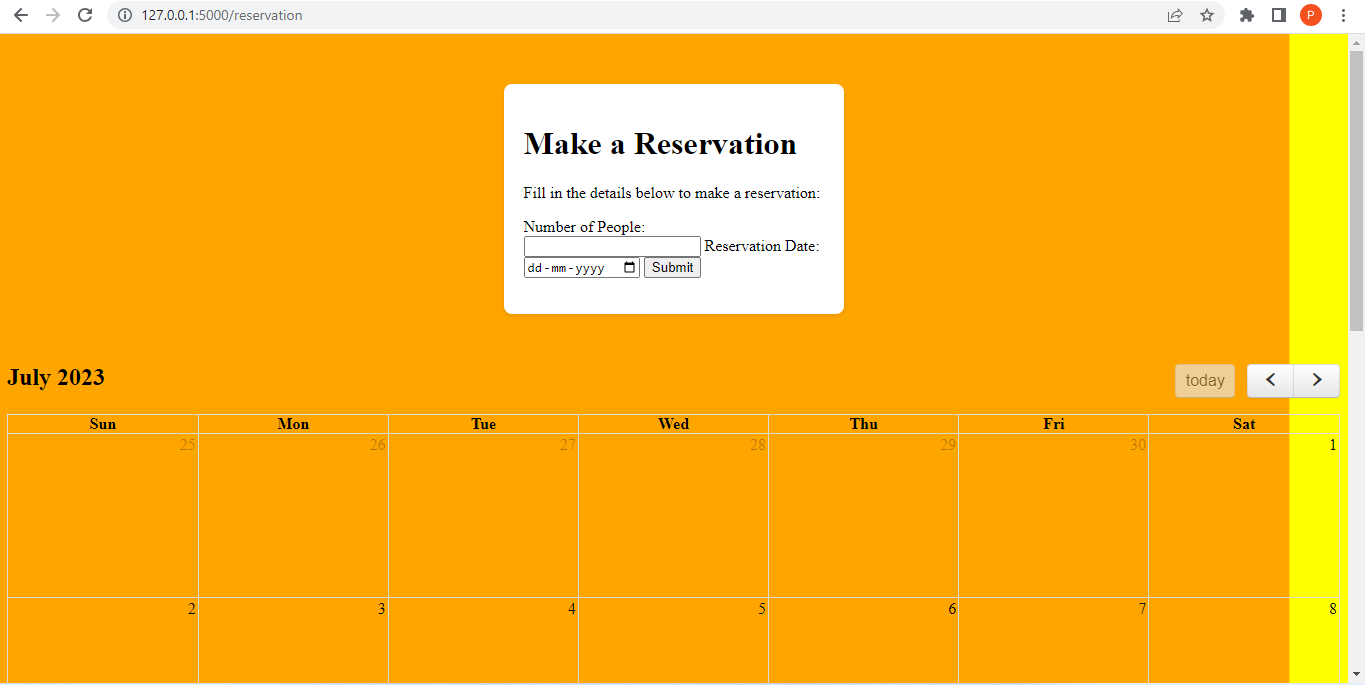


**Test/run the application:**

Run the application on the server through local or docker image or by pulling image through github or azure.







# Recommendations

# Cloud vs non-cloud solutions

It is recommended that the company should adopt cloud solutions as:

* Cloud solutions involve deploying, hosting the applications on the remote server, or using the remote database to store the data.
* Using the cloud, we can scale up or scale down the applications based on our requirements in order to save cost and maintain speed.
* Using the cloud anyone can access the remote server without much more dependency on the on-premise infrastructure.
* Cloud solution works on the pay-as-you-go model where we are charged only for the services or resources we used and for the amount of time we use that service. We are not charged for the infrastructure setup costs (Microsoft 2023).
* Cloud services involve updating software, ensuring that applications and underlying infrastructure are regularly patched and kept up to date. This reduces the burden on IT teams.
* Cloud solution maintains the infrastructure requirements including hardware updates, security patches, and fault tolerance.
* Cloud solutions provide quick provisioning and deployment of applications, reducing time-to-market for businesses.

Appropriate deployment types and service level

For deploying the cloud resources or services we needed.

* Docker instance
* Docker Container
* GitHub Container Registry
* Containerization of the application
* Updating the application.
* Multi-Container Apps

Services/Resources used:

* Azure Active registry
* Github Registry
* Docker Desktop
* Python/HTML/CSS/JS

## Justification for Recommendation

The traditional on-premise infrastructure basically known as non-Cloud solutions involves deploying and hosting the applications on the On-Premise infrastructure. We are not able to scale up or scale down based on the demands. We don’t get the flexibility to use the remote access of the servers. Also, it requires high infrastructure costs for the setup. The deployment of the resources and services takes more time and it’s not fully automated.

Considering all these conditions for the organization which requires scalability, less infrastructure costs, and remote access to the resources it is necessary to move from on-premise to the cloud environment which includes moving the on-premise system to the cloud resources and using the containerization to host the application on the remote server or by using the virtual machine on the cloud server.

# Conclusion

The report represents the successful deployment of the Web application of the Copita tapas bar that is useful for different purposes like better storage, data security, and efficient processing of Code. It also includes the recommendation for cloud and non-cloud-based solutions.

# References

Copita. (2020). *Copita Tapas bar*. Available at: https://copita.co.uk/ [Accessed 10 Jul. 2023].

Docker. (2022). *Docker: Accelerated, Containerized Application Development*. [online] Available at: https://www.docker.com/ [Accessed 10 Jul. 2023].

GeeksforGeeks. (2020). *Differences Between Django vs Flask*. [online] Available at: https://www.geeksforgeeks.org/differences-between-django-vs-flask/ [Accessed 10 Jul. 2023].

Microsoft.com. (2023). *Global Infrastructure.* Microsoft Azure. [online] Available at: https://azure.microsoft.com/en-us/explore/global-infrastructure [Accessed 10 Jul. 2023].

‌

‌

‌

‌