### Deployment of Boston House Pricing Using Docker, Kubernetes, and Jenkins

#### Submitted by

***Asmi Garg***

***2110993769***

**Semester: 6**

Logo

Description automatically generated

**BE-CSE (Artificial Intelligence)**

***Guided by***

**Chaitanya Soni**

**CHITKARA UNIVERSITY INSITUTE OF ENGINEERING & TECHNOLOGY**

**CHITKARA UNIVERSITY, RAJPURA**

**April,2024**

**Table of Contents**

1. Acknowledgment
2. Abstract
3. Introduction
4. Objectives
5. Project Overview
6. Docker
7. Kubernetes
8. Jenkins
9. Conclusion

ACKNOWLEDGEMENT

With immense please I, Ms Asmi Garg presenting deployment of Boston house pricing on Docker , Kubernetes and Jenkins project report as part of the curriculum of ‘ BE-CSE (AI) ’.

I would like to express my sincere thanks to Mr Chaitanya Soni for his valuable guidance and support in completing my project.

I would also like to express my gratitude towards our dean Dr. Sushil Narang for giving me this great opportunity to do this project. Without their support and suggestions, this project would not have been completed.

Name: Asmi Garg

Roll no. :2110993769

### ABSTRACT

This report provides a comprehensive overview of the deployment process for the "Boston House Pricing" project utilizing Docker, Kubernetes, and Jenkins. Beginning with an introduction to containerization and continuous integration/continuous deployment (CI/CD), the report delves into the specific methodologies employed for each platform. It details the steps involved in containerizing the application with Docker, orchestrating it within a Kubernetes cluster, and implementing an automated deployment pipeline using Jenkins.

Throughout the report, key concepts such as Docker images, Kubernetes pods, Jenkins pipelines, and deployment strategies are discussed in depth. Practical insights are provided to address common challenges faced during implementation.

Furthermore, the report includes a comparative analysis of Docker, Kubernetes, and Jenkins, evaluating their respective strengths, limitations, and suitability for different deployment scenarios.

In summary, this report serves as a comprehensive guide to deploying modern applications using Docker, Kubernetes, and Jenkins, offering valuable insights for developers, DevOps engineers, and technology enthusiasts alike.

### Introduction

The advent of modern software development practices has revolutionized the way applications are built, deployed, and managed. In today's rapidly evolving technological landscape, the demand for scalable, reliable, and efficient deployment solutions has never been greater. This report explores the deployment process for the "Boston House Pricing" project using three popular DevOps tools: Docker, Kubernetes, and Jenkins.

The "Boston House Pricing" project aims to develop a predictive pricing model for real estate properties in the Boston area. Leveraging machine learning algorithms and statistical analysis, the project seeks to provide accurate estimates of housing prices based on various features such as location, size, and amenities.

### Objectives

The primary objective of this report is to demonstrate the deployment workflow for the "Boston House Pricing" project using Docker, Kubernetes, and Jenkins. Specifically, the report aims to:

* Containerize the application using Docker to create lightweight, portable environments.
* Orchestrate the application within a Kubernetes cluster to achieve scalability, resilience, and resource optimization.
* Implement a CI/CD pipeline using Jenkins to automate the deployment process and ensure consistency and reliability.

This report focuses on the deployment aspect of the "Boston House Pricing" project, with an emphasis on the utilization of Docker, Kubernetes, and Jenkins. While the project development and data analysis phases are crucial, they are beyond the scope of this report. By examining the deployment process through the lens of these three DevOps tools, this report aims to provide insights into modern deployment practices and empower readers to leverage containerization, orchestration, and automation for their own projects.

### PROJECT OVERVIEW

The "Boston House Pricing" project aimed at developing a predictive pricing model for residential properties in the Boston metropolitan area. The project leverages machine learning techniques and statistical analysis to estimate housing prices based on a variety of factors such as location, size, age, and amenities.

The primary goal of the project is to provide accurate and reliable estimates of housing prices to assist prospective buyers, sellers, and real estate professionals in making informed decisions. By analyzing historical sales data and property attributes, the project seeks to uncover underlying patterns and trends that influence housing prices in the Boston market.

### Technologies Used

The project utilizes a range of technologies and tools to achieve its objectives:

* Python: The primary programming language used for data preprocessing, model development, and analysis.
* Scikit-learn: A machine learning library in Python used for building and evaluating predictive models.
* Pandas: A data manipulation library in Python used for data cleaning, transformation, and analysis.
* Jupyter Notebook: An interactive computing environment used for exploratory data analysis and model prototyping.

The "Boston House Pricing" project addresses a significant real-world problem faced by homebuyers, sellers, and industry professionals in the housing market. Accurate pricing information is crucial for making informed decisions and maximizing the value of real estate transactions. By providing a data-driven approach to pricing estimation, the project aims to contribute valuable insights and tools to the real estate industry.

### DOCKER

Docker serves as a pivotal tool in modern software development, harnessing the power of containerization technology. This innovative approach enables applications and their dependencies to be encapsulated into lightweight, portable containers. These containers create isolated environments, ensuring the consistency and reliability of applications across various computing environments, from development through to production stages.

Below are the steps to deploy "Boston House Pricing" project using Docker:

* **Installation of Docker Desktop:**

1. Visit the official Docker website at www.docker.com/products/docker-desktop.
2. Download Docker Desktop suitable for your operating system.
3. Follow the installation instructions to set up Docker on machine.

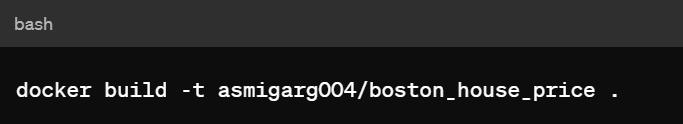
* **Project Setup:**

1. Create a dedicated folder for the project.
2. Place a Python file containing the Flask model within this folder.
3. Create a Dockerfile (without any extension) within the same folder to specify the project's configuration.

* **Building the Docker Image:**

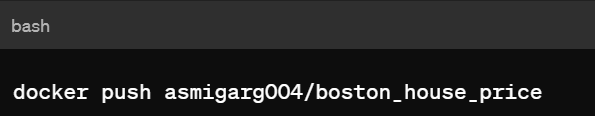
1. Launch Docker Desktop and ensure it's operational.
2. Open Command Prompt (CMD) and navigate to the directory where project files are located.

* **Execute the following command to build the Docker image:**

****

* **Integration with Docker Hub:**

1. Navigate to the Docker Hub website and log in using the credentials.
2. Create a new repository named "boston\_house\_price".
3. Push the tagged image to Docker Hub:



After completing these steps, "Boston House Pricing" project will be successfully built as a Docker image, ready for deployment, thus streamlining both development and deployment processes.

### KUBERNETES

Kubernetes, often abbreviated as K8s, is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. It offers a robust set of tools for managing complex workloads in production environments. Kubernetes, originally developed by Google, has emerged as the industry standard for container orchestration, offering a robust platform for managing containerized applications at scale. With its declarative approach to application deployment and management, Kubernetes simplifies the complexities of containerized environments, allowing developers to focus on building and deploying applications without worrying about the underlying infrastructure. By providing features such as automated scaling, rolling updates, and service discovery, Kubernetes enables organizations to achieve higher levels of agility, resilience, and efficiency in deploying modern applications in dynamic environments.

Below are the steps to deploy the "Boston House Pricing" project using Kubernetes:

* **Installation of Kubernetes:**

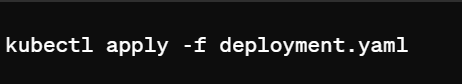
1. Install Kubernetes on your chosen platform, whether it's a local development environment, on-premises servers, or a cloud provider.
2. Refer to the Kubernetes documentation or your cloud provider's documentation for installation instructions.

* **Setting Up Kubernetes Resources**:

1. Define Kubernetes resource specifications to describe the desired state of your application.
2. This involves creating Deployments, Pods, Services, ConfigMaps, and Secrets necessary for your application to run.

* **Deploying the Application:**

1. Apply the Kubernetes resource specifications using the kubectl apply command to deploy the application to the Kubernetes cluster.



### JENKINS

Jenkins is an open-source automation server that streamlines the software development lifecycle by automating tasks such as building, testing, and deploying applications. It facilitates continuous integration and delivery (CI/CD) processes, enabling teams to deliver high-quality software more efficiently.

**Key Features of Jenkins**

* Continuous Integration (CI): Jenkins automates the process of integrating code changes into a shared repository, allowing teams to detect and resolve integration errors early in the development cycle.
* Continuous Delivery (CD): Jenkins automates the deployment process, enabling teams to deliver software updates rapidly and reliably to production or staging environments.
* Pipeline as Code: Jenkins Pipeline allows teams to define the entire software delivery process as code, enabling versioning, sharing, and reuse of pipelines across projects.

**Deploying with Jenkins**

To deploy the "Boston House Pricing" project using Jenkins:

**Put Project on GitHub**: Start by putting the "Boston House Pricing" project on GitHub. This involves creating a new repository on GitHub and pushing the project code to the repository.

**Set Up Jenkins Job**: Create a new Jenkins job to automate the deployment process of the project. Configure the job to pull the latest changes from the GitHub repository, build the project, and deploy it to the target environment.

**Write Jenkins Pipeline Script:** Utilize Jenkins Pipeline to define and visualize the deployment process as code. Write a pipeline script that includes stages for pulling the code from the GitHub repository, building the project, running tests, and deploying the application.

**Execute Pipeline**: Run the Jenkins pipeline to trigger the deployment process. Jenkins will automatically pull the code from the GitHub repository, execute the defined stages in the pipeline script, and deploy the application according to the specified configuration.

**Monitoring and Management:** Regularly monitor Jenkins jobs and pipelines to ensure smooth execution. Manage Jenkins resources and optimize performance as needed.

### CONCLUSION

In conclusion, the deployment of the "Boston House Pricing" application using Docker, Kubernetes, and Jenkins exemplifies the efficacy of modern software development practices. These technologies offer robust capabilities for automating and streamlining the software development lifecycle, from building and testing to deployment and delivery.

Docker facilitates the containerization of applications, ensuring consistency and reliability across diverse environments. Kubernetes provides powerful orchestration and management features, allowing for efficient scaling and management of applications in production environments. Jenkins automates the continuous integration and delivery process, enabling rapid and dependable deployment of software updates.

This project underscores the significance of automation and standardization in contemporary software development methodologies. By leveraging these technologies, deployment processes are simplified, and overall efficiency and reliability of the development workflow are enhanced.

Looking forward, the insights gained from this project will continue to inform the adoption of best practices and the exploration of emerging technologies to drive innovation and deliver value to stakeholders.