**DEPLOYMENT OF MACHINE LEARNING MODEL ON DOCKER, KUBERNETES AND JENKINS**

**PROJECT REPORT**  **ON**

**MLOP’s PROJECT**



***Submitted by:***

**DEVANSH PRASHAR (2110993775)**

**ISHITA JINDAL (2110993799)**

**SHUBHANK SHARMA(2110993837)**

**ADVITIYA BHARTI GUPTA(2110993858)**

**BE CSE (ARTIFICIAL INTELLIGENCE)**

**GUIDED BY**

**Mr. Chaitanya Soni**

**CHITKARA UNIVERSITY INSTITUTE OF ENGINEERING &**

**TECHNOLOGY**

**CHITKARA UNIVERSITY, RAJPURA**

**APRIL 2024**

# Acknowledgement

With immense pleasure **we, Mr. Devansh Prashar, Ms. Ishita Jindal, Mr. Shubhank Sharma, Mr Advitiya Bharti Gupta** presenting Machine learning Project Deployment on **“Docker, Kubernetes and Jenkins”** project report as part of the curriculum of ‘BE-CSE(AI)’.

We would like to express my sincere thanks to **Mr. Chaitanya Soni** and our mentor **Dr. Harshvardhan** for their valuable guidance and support in completing my project.

We would also like to express my gratitude towards our dean **Dr. Sushil Kumar Narang** for giving us this great opportunity to do a project on **Docker, Kubernetes and Jenkins Project Deployment**. Without their support and suggestions, this project would not have been completed.

Name: Devansh Prashar (2110993775)

Signatures: ............................

Name: Ishita Jindal (2110993799)

Signatures: …............................

Name: Shubhank Sharma (2110993837)

Signatures: ............................

Name: Advitiya Bharti Gupta(2110993858)

Signatures: …............................

**ABSTRACT**

In today's software development landscape, efficiency and reliability are paramount for ensuring seamless integration and rapid iteration across diverse domains. This report delves into the synergistic use of Docker, Kubernetes, and Jenkins to optimize project deployments within the distribution pipeline. Docker containers offer lightweight, portable environments ideal for packaging applications, while Kubernetes orchestrates these containers, automating deployment and scaling for enhanced efficiency and fault tolerance. Jenkins further streamlines the process by standardizing automated testing, building, and deployment through continuous integration and delivery (CI/CD).

The report explores various deployment strategies such as blue-green, canary, and rolling updates, assessing their suitability across different scenarios. It also highlights packaging techniques like Docker File optimization and image caching to expedite deployment. Integration of monitoring tools within pipelines ensures visibility into performance metrics and enables proactive issue resolution. Security considerations, including containerization best practices and vulnerability analysis, are addressed to fortify infrastructure against potential threats.

Through real-life case studies and practical examples, the report illustrates the benefits of leveraging Docker, Kubernetes, and Jenkins for agile, scalable, and reliable deployments. It concludes with recommendations for optimizing the delivery pipeline, addressing diverse challenges, and adapting to evolving technological landscapes, empowering organizations to implement delivery processes effectively in today's dynamic software development ecosystem.

**INTRODUCTION**

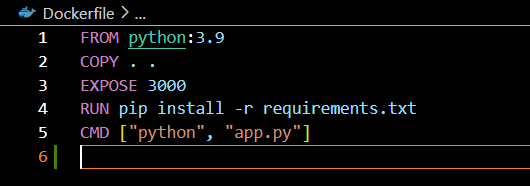
## DOCKER

The introduction of Docker has revolutionized software development and deployment practices by providing a standardized method to package, distribute, and execute applications within lightweight, portable containers. It effectively addresses the complexities associated with managing diverse dependencies and ensures uniformity across various environments. Through containerization, Docker encapsulates applications and their dependencies, resolving compatibility issues and simplifying deployment workflows. Developers utilize Docker files to configure applications, generating Docker images that serve as portable units for container execution. This approach fosters consistency across development, testing, and production environments, streamlining deployment processes.

Furthermore, Docker aligns with infrastructure as code (IaC) principles, enabling the programmable definition and management of infrastructure components. As organizations increasingly embrace hybrid and multi-cloud setups, Docker's portability and scalability emerge as foundational elements in contemporary software development pipelines. Its agility, efficiency, and reliability empower teams to adapt swiftly to evolving demands, solidifying its status as a critical technology in modern software development.

**Steps to deploy on Docker:**

1. Docker Installation:
   * Visit the official Docker website
   * [www.docker.com/products/docker-desktop](http://www.docker.com/products/docker-desktop)
   * Follow installation steps
2. Make a separate folder: A.py file containing a flask model and a docker file with no extension and write the following in it.



1. Image building:
   * Open docker desktop
   * Open CMD in the folder, and run the following commands.
   * Docker build -t image name.
   * The built image will be seen on docker desktop
2. Docker hub: Log in in to your docker hub online. Click on create repository and fill in the credentials
3. Image pushing to Docker Hub:
   * Run the following commands on the cmd to push the image to the docker hub.
   * Docker tag local-image:tagname new-repo:tagname
   * Docker push new-repo:tagname
4. Now our image in pushed to docker hub

## Kubernetes

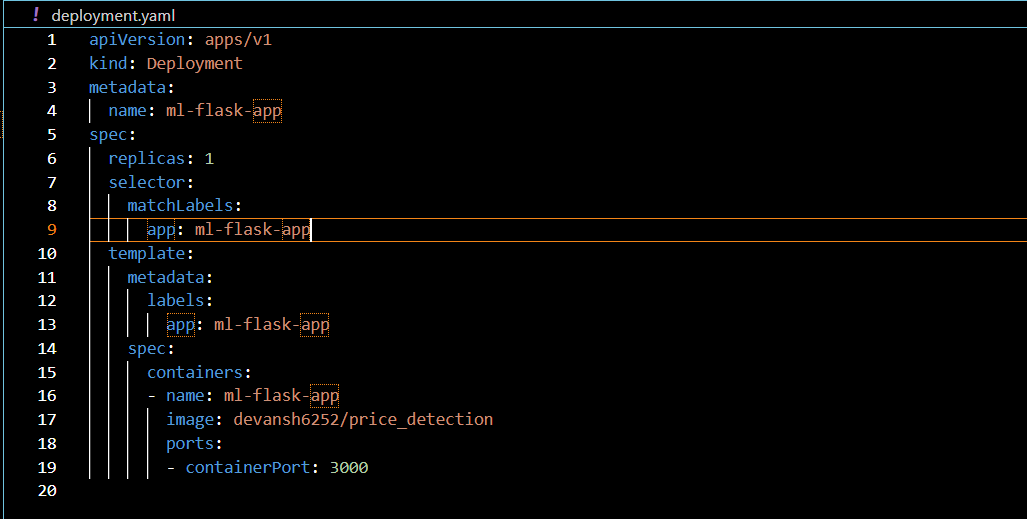
Kubernetes stands out as a pivotal technology reshaping the landscape of deploying and managing containerized applications. Originally developed by Google and made open source in 2014, Kubernetes serves as a robust orchestration platform automating the deployment, scaling, and operation of application containers across machine clusters. With the advent of microservices architectures and containerization, Kubernetes tackles the complexities of managing distributed systems by abstracting away infrastructure intricacies and offering a declarative model to define desired states.

Equipped with a comprehensive suite of features including service discovery, load balancing, and automated rollouts and rollbacks, Kubernetes streamlines the management of containerized workloads. This empowers organizations to achieve heightened levels of agility, scalability, and reliability. Through Kubernetes, teams can effortlessly deploy and scale applications, optimize resource allocation, and ensure uninterrupted availability in dynamic and diverse environments.

As businesses increasingly embrace cloud-native technologies, Kubernetes has evolved into an indispensable tool for orchestrating containerized infrastructure and catalyzing innovation in modern software development methodologies..

**Steps to deploy on Kubernetes:**

1. Kubernetes Installation:
   * Run the following command in CMD
   * Curl.exe -LO <https://dl.k8s.io/release/v1.29.2/bin/windows.amd64/kubectl.exe>
   * Check the version using the following command
   * kubectl version -client
2. Docker Settings:
   * Open Docker Hub
   * Click on settings
   * Choose container terminal as integrated
   * Tick on (Expose daemon on tcp://localhost:2375 without TLS)
   * Click on apply & restart
3. Kubernetes Commands:
   * kubectl config view
   * kubectl get nodes
4. Yaml File: Build a deployment.yaml in the same folder using the following code.



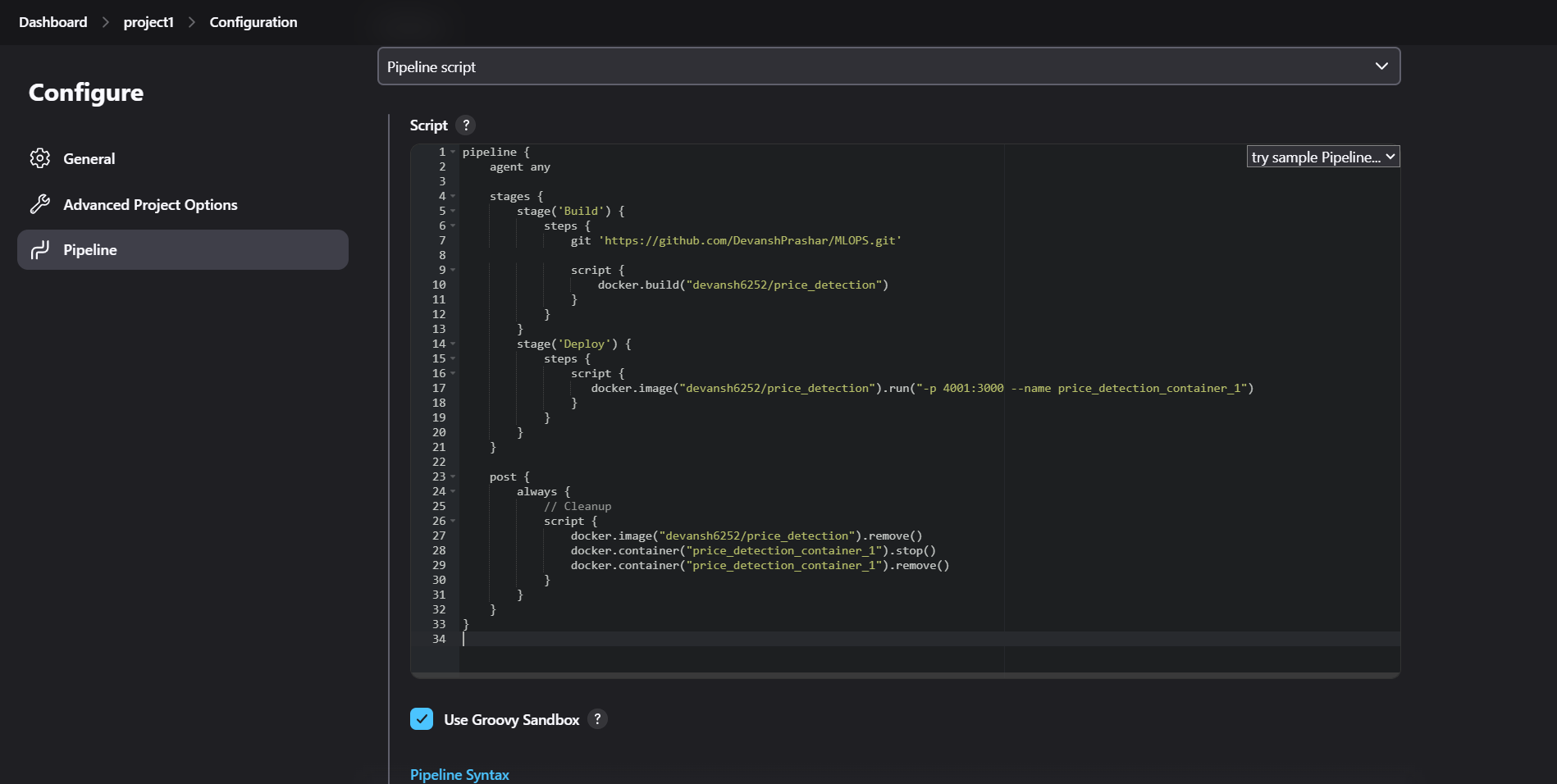
1. Deployments:
   * Use the following command to deploy the model
   * Kubectl apply -f deployments.yaml

## Jenkins

Jenkins plays a central role in modern software development as an automation server that facilitates continuous integration and continuous delivery (CI/CD) pipelines. Originating as Hudson and later rebranded as Jenkins, it empowers developers to automate essential tasks such as building, testing, and deploying applications. With its vast array of plugins and seamless integration capabilities, Jenkins streamlines workflows and promotes collaboration among development teams. By automating repetitive tasks and fostering an agile culture, Jenkins enables organizations to accelerate release cycles, enhance code quality, and embrace DevOps principles. Consequently, Jenkins has become synonymous with efficient software delivery and serves as a catalyst for innovation across various industries.

**Steps to deploy on Jenkins**:

1. Log in to Jenkins using your local host.
2. Click on the new item & enter the item name to create a new pipeline.
3. Click on pipeline
4. Click on the “GitHub hook trigger for GITScm polling” and enable it.
5. Now write the pipeline script



1. Enable use Groovy Sandbox.
2. Now build the project