

# PROJECT DOCUMENTATION

## 1. TITLE :

CI/CD Pipeline for Zomato Application using Node.js, Docker, Kubernetes(kops cluster), and GitHub Actions .

## 2. INTRODUCTION :

Modern application development demands automated pipelines to ensure continuous integration and delivery. This project demonstrates how to containerize a Node.js Zomato application and automate its build-test-deploy lifecycle using GitHub Actions for CI/CD, Docker for containerize the application and Deployment is handled by using kubernetes on kops cluster .

## 3. ABSTRACT :

This project demonstrates the implementation of a CI/CD pipeline for the Zomato application using Docker containers, GitHub Actions, and Kubernetes(Kops) Pods.The pipeline automatically builds Docker images, to build the code, runs unit tests, pushes images to Docker Hub, and deploys them using kubernetes(Kops). Here am using KOPS Cluster because of high availability my cluster. In this project a zomato application am deploying in two ways and the first one is docker-compose am creating docker containers to run the application and the second one is kubernetes (kops cluster) am creating Pods to run the application. The pipeline is ensures consistent, automated and making it suitable for local environments and learning devops practices.

## 4. TOOLS AND TECHNOLOGIES USED :

Git, GitHub, Node.js Jenkins, GitHub Actins for CI/CD Automation, Docker, Docker-Compose, DockerHub, Kubernetes(Kops Cluster), Kubectl .

## 5. STEPS INVOLVED IN BUILDING THE PROJECT :

#STEP-1 : First of all taking a New Server with Server configurations are server\_name : project, AMI : Amazon linux kernel 5.10, Instance\_type : t2.medium, Key\_Pair : sai-kp, Security\_Group : All-Traffic, EBS : 20GIB . After connecting to Server with SSH .

#STEP-2 : Now To Set-Up the Tools Git, Docker, Docker-Compose, Jenkins, Kubernetes(Kops) by using Commands .

#STEP-3 : Now am creating a New Repository in GitHub and collect the source code via browser . And the Repository Name is project-4 .

#STEP-4 : After am accessing the Jenkins dash board with public-ip:8080 . 8080 it is a port number . Now am creating a New job for the pipeline running purpose .

#Step-5 : Now am installed some plugins for my zomato application purpose .

- PLUGIN\_NAME : pipeline stage view , node.js, docker pipeline, eclipse temurin installer . After configured all the plugins to my Jenkins job.
- For my Zomato Application am using here two ways of deployments and the first one am deploying in docker-compose and second one is am deploying in kubernetes kops cluster then pods are created .
- Now writing and implementing CI/CD pipeline with the stages AND CI/CD WorkFlow – Results .

stage-1 : It is a clean workspace .

stage-2 : To get the source code from github to ci server .

stage-3 : To build the Source Code .

stage-4 : To build the dockerfile .then a New docker-image has to come .

stage-5 : To Rename the docker-image and that docker-image push to Docker-Hub .

stage-6 : To deploy the application in containers .

#STEP-6 : After my CI/CD pipeline is successfully executed . Then a New Docker Image and Docker Containers are also created .

NOTE : am taking the those screenshots of CI/CD pipeline , docker image and docker containers are pasted in my GitHub Repository .

- There are some Docker Commands are executed in my project and the commands are :
- COMMAND : docker build -t image\_name - To build the dockerfile .
- COMMAND : docker-compose up -d - To create a containers .
- COMMAND : docker images - To check the list of docker-images .
- COMMAND : docker ps - To check the list of docker-containers .
- COMMAND : docker push image\_name - To push the image to dockerhub .

#STEP-7 : Now am deploying my application in second way that is kubernetes kops cluster . So Here am creating KOPS cluster because of my cluster is high availability . Now to set-up a Kubernetes Kops Cluster in server by using ,

- awscli commands . (commands are collecting from via browser) .
  - kubectl commands . (commands are collecting from via browser) .
  - kops cluster commands . (commands are collecting from via browser) .
  - Now am creating a kops cluster creation by using command .
- COMMAND : kops create cluster --name=sai.k8s.local --zones=us-east-1a,us-east-1b --master-size=t2.medium --master-count=1 --master-volume-size=30 --node-size=t2.medium --node-count=3 --node-volume-size=30 .

- COMMAND : kops get cluster - To check the list of clusters .
- After KOPS cluster set-up is completed . Now am creating two files for depoying my Zomato Application .
- 1. one is deployment YAML file and the name is deployment.yaml. 2. second is service YAML file and the name is svc.yaml .

```

root@ip-172-31-80-112:~/folder
[root@ip-172-31-80-112 folder]# cat deployment.yaml
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: zomato
  template:
    metadata:
      labels:
        app: zomato
    spec:
      containers:
        - name: app-container
          image: saikumar1817/project-4:mydockerimage
          ports:
            - containerPort: 3000
[root@ip-172-31-80-112 folder]#
[root@ip-172-31-80-112 folder]#
[root@ip-172-31-80-112 folder]# cat service.yaml
---
apiVersion: v1
kind: Service
metadata:
  name: app-service
spec:
  type: LoadBalancer
  selector:
    app: zomato
  ports:
    - port: 3000
      targetPort: 3000
[root@ip-172-31-80-112 folder]#
[root@ip-172-31-80-112 folder]#
[root@ip-172-31-80-112 folder]#

```

This is the SCREENSHOTS of deployment files deployment.yaml and svc.yaml files .

- Some Commands are there to Apply & Access .
  - COMMAND : kubectl apply -f deployment .yaml - To Executed the deployment file .
  - COMMAND : kubectl apply -f svc.yaml - To Executed the service file .
- There are some commands executed in my project .
  - COMMAND : kubectl get po - To Check the list of pods .
  - COMMAND : kubectl get deploy - To Check the list of Deployments .
  - COMMAND : kubectl get svc - To check the list of services .

NOTE : Here am taking the screenshots of list of pods , deployments , services are pasted in GitHub Repository .

#STEP-8 : overall my Zomato Application deploying is completed .

- Now am accessing the Zomato application with public\_ip:31482 .

NOTE : am taking the SCREENSHOT of deployed Zomato Application and sending to GitHub Repository .

## 7. CONCLUSION :

This project successfully implements a CI/CD pipeline for a Node.js Zomato Application. By using Docker and Kubernetes to set-up KOPS Cluster it replicates real-world deployment workflows. This method provides a foundational understanding of DevOps practices and prepares the application for production-grade infrastructure in the future.

NOTE : am taking all the SCREENSHOTS and sending to my GITHUB REPOSITORY into Deliverables folder .