**Text Case Converter Application Deployment Documentation**

This documentation covers the complete setup, deployment, and troubleshooting of the Text Case Converter application using Docker, Jenkins, and Kubernetes (Minikube).

**Table of Contents**

1. Application Overview
2. Development Environment
3. Docker Configuration
4. CI/CD Pipeline with Jenkins
5. Kubernetes Deployment
6. Troubleshooting Common Issues

**1. Application Overview**

**Text Case Converter** is a web application that transforms text into various case formats:

Uppercase, Lowercase, CamelCase, Title Case, Alt Case, Pascal Case, Sentence case, Initial Case, Swap case

The application provides a simple UI with text input field and conversion buttons, allowing users to copy results to clipboard or download as text files.

**2. Development Environment**

**Prerequisites**

* JDK 21
* Git
* Docker
* Jenkins
* Kubernetes (Minikube)
* kubectl CLI

**Local Development**

For local development and testing:

# Clone the repository

git clone https://github.com/GiridharanS1729/text-case-converter.git

cd text-case-converter

index.html

**3. Docker Configuration**

**Dockerfile**

FROM nginx:alpine

COPY . /usr/share/nginx/html

EXPOSE 80

**Building and Testing Docker Image Locally**

# Build the Docker image

docker build -t giridharans1729/text-case-converter:latest .

# Run the container locally

docker run -d -p 8080:80 giridharans1729/text-case-converter:latest

# Verify the application is working

# Visit http://localhost:8080 in a browser

**4. CI/CD Pipeline with Jenkins**

**Jenkins Setup**

1. Install required Jenkins plugins:
   * Git plugin
   * Docker plugin
   * Pipeline plugin
   * Credentials plugin
2. Configure credentials in Jenkins:
   * github\_seccred: GitHub credentials
   * docker: Docker Hub credentials

**Jenkinsfile (Pipeline Definition)**

pipeline {

agent any

tools { jdk 'jdk21' }

stages {

stage('Clean Workspace') {

steps {

script {

echo "Cleaning workspace..."

deleteDir()

}

}

}

stage('Git Checkout') {

steps {

script {

git branch: 'main',

credentialsId: 'github\_seccred',

url: 'https://github.com/GiridharanS1729/text-case-converter.git'

}

}

}

stage('Docker Build & Push') {

steps {

script {

withDockerRegistry(credentialsId: 'docker', toolName: 'docker') {

def imageName = "giridharans1729/text-case-converter"

def tag = "latest"

sh "docker build -t ${imageName} ."

sh "docker tag ${imageName} ${imageName}:${tag}"

sh "docker push ${imageName}:${tag}"

}

}

}

}

}

}

**Setting Up Jenkins Job**

1. Create a new Pipeline job
2. Configure it to use SCM for pipeline definition
3. Point to your repository and specify the Jenkinsfile path
4. Set up webhook triggers for automatic builds

**5. Kubernetes Deployment**

**Minikube Setup**

# Start Minikube

minikube start

# Enable ingress addon (optional)

minikube addons enable ingress

**Deployment YAML (deployment.yaml)**

apiVersion: apps/v1

kind: Deployment

metadata:

name: static-website

spec:

replicas: 1

selector:

matchLabels:

app: static-website

template:

metadata:

labels:

app: static-website

spec:

containers:

- name: static-website

image: giridharans1729/text-case-converter:latest

imagePullPolicy: Always

ports:

- containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

name: static-website-service

spec:

selector:

app: static-website

ports:

- protocol: TCP

port: 80

targetPort: 80

type: LoadBalancer

**Deploy to Kubernetes**

# Apply the deployment configuration

kubectl apply -f deployment.yaml

# Verify deployment

kubectl get deployments

kubectl get pods

kubectl get services

# Access the application

minikube service static-website-service –url

**6. Troubleshooting Common Issues**

**Service Unreachable Error**

If you encounter "SVC\_UNREACHABLE: service not available: no running pod for service" error:

1. Check pod status:
2. kubectl get pods -l app=static-website
3. Check pod logs for errors:
4. kubectl logs <pod-name>
5. Verify image pull status:
6. kubectl describe pod <pod-name>
7. Check service configuration:
8. kubectl describe service static-website-service
9. Get detailed logs:
10. minikube logs --file=logs.txt

**Pod Crashes**

If pods are crashing after starting:

1. Check application logs:
2. kubectl logs <pod-name> --previous
3. Verify container port configuration in both Dockerfile and deployment YAML
4. Test the Docker image locally before deployment

**LoadBalancer Pending State**

If the LoadBalancer stays in pending state:

1. Remember that Minikube doesn't support LoadBalancer by default:
2. # Use minikube tunnel in a separate terminal
3. minikube tunnel
4. Or access through NodePort:
5. # Change service type to NodePort
6. kubectl patch svc static-website-service -p '{"spec": {"type": "NodePort"}}'
7. minikube service static-website-service

**7. Maintenance Guidelines**

**Updating the Application**

1. Make changes to application code
2. Commit and push to GitHub
3. Jenkins pipeline will automatically trigger a new build and Docker image
4. Update Kubernetes deployment:
5. kubectl rollout restart deployment static-website

**Scaling**

# Scale the deployment

kubectl scale deployment static-website --replicas=3

**Health Monitoring**

Consider adding health checks to the deployment:

spec:

containers:

- name: static-website

livenessProbe:

httpGet:

path: /

port: 80

initialDelaySeconds: 30

periodSeconds: 10

readinessProbe:

httpGet:

path: /

port: 80

initialDelaySeconds: 5

periodSeconds: 5

This documentation provides a comprehensive guide for deploying and maintaining the Text Case Converter application. For specific issues not covered here, refer to the official documentation for Docker, Jenkins, and Kubernetes.

**SCREENSHOTS**



