**🚀 Full Deployment Guide — Frontend (React + Tomcat) with Helm & Kubernetes**

**1️⃣ Prerequisites**

Make sure you have:

* **Docker Desktop** installed with **Kubernetes enabled**
* **Helm** installed
* **kubectl** installed
* Your **frontend Docker repo**: https://github.com/srithars/s207-ecomm-frontend-docker.git

Check versions:

kubectl version --client

helm version

docker --version

kubectl get nodes

**2️⃣ Dockerfile for Frontend**

Create Dockerfile.frontend:

# Step 1: Build stage (Node.js for React/Vite)

FROM node:18 AS build-stage

WORKDIR /app

# Install git to allow repo cloning

RUN apt-get update && apt-get install -y git

# Clone your frontend repo

RUN git clone https://github.com/srithars/s207-ecomm-frontend-docker.git .

# Install dependencies and build

RUN npm install

RUN npm run build

# Step 2: Runtime stage (Tomcat for serving)

FROM tomcat:9-jdk17

RUN rm -rf /usr/local/tomcat/webapps/\*

# Copy React build output into Tomcat

COPY --from=build-stage /app/dist /usr/local/tomcat/webapps/ecommerce

EXPOSE 8085

CMD ["catalina.sh", "run"]

**3️⃣ Build Docker Image**

docker build -t srithar123/skill14:v1 -f Dockerfile.frontend .

Optional: run locally to test:

docker run -p 8085:8085 srithar123/skill14:v1

**4️⃣ Create Helm Chart**

helm create frontend

This creates a folder frontend/ with Chart.yaml, values.yaml, and templates/.

**5️⃣ Edit values.yaml**

Update frontend/values.yaml with:

replicaCount: 2

image:

repository: srithar123/skill14

tag: v1

pullPolicy: IfNotPresent

service:

type: NodePort

port: 8085

targetPort: 8080

nodePort: 30090 # Changed to avoid conflicts

ingress:

enabled: true

className: "nginx"

hosts:

- host: frontend.local

paths:

- path: /

pathType: Prefix

tls: []

resources:

requests:

cpu: 100m

memory: 128Mi

limits:

cpu: 300m

memory: 256Mi

autoscaling:

enabled: true

minReplicas: 2

maxReplicas: 5

targetCPUUtilizationPercentage: 70

serviceAccount:

create: false

httpRoute:

enabled: false

**6️⃣ Edit Templates**

**deployment.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: {{ include "frontend.fullname" . }}

labels:

app: {{ include "frontend.name" . }}

spec:

replicas: {{ .Values.replicaCount }}

selector:

matchLabels:

app: {{ include "frontend.name" . }}

template:

metadata:

labels:

app: {{ include "frontend.name" . }}

spec:

containers:

- name: frontend

image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"

ports:

- containerPort: {{ .Values.service.targetPort }}

resources:

{{- toYaml .Values.resources | nindent 12 }}

**service.yaml**

apiVersion: v1

kind: Service

metadata:

name: {{ include "frontend.fullname" . }}

spec:

type: {{ .Values.service.type }}

selector:

app: {{ include "frontend.name" . }}

ports:

- port: {{ .Values.service.port }}

targetPort: {{ .Values.service.targetPort }}

nodePort: {{ .Values.service.nodePort }}

**ingress.yaml**

{{- if .Values.ingress.enabled }}

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: {{ include "frontend.fullname" . }}

spec:

ingressClassName: {{ .Values.ingress.className }}

rules:

{{- range .Values.ingress.hosts }}

- host: {{ .host }}

http:

paths:

{{- range .paths }}

- path: {{ .path }}

pathType: {{ .pathType }}

backend:

service:

name: {{ include "frontend.fullname" $ }}

port:

number: {{ $.Values.service.port }}

{{- end }}

{{- end }}

{{- end }}

**hpa.yaml**

{{- if .Values.autoscaling.enabled }}

apiVersion: autoscaling/v2

kind: HorizontalPodAutoscaler

metadata:

name: {{ include "frontend.fullname" . }}

spec:

scaleTargetRef:

apiVersion: apps/v1

kind: Deployment

name: {{ include "frontend.fullname" . }}

minReplicas: {{ .Values.autoscaling.minReplicas }}

maxReplicas: {{ .Values.autoscaling.maxReplicas }}

metrics:

- type: Resource

resource:

name: cpu

target:

type: Utilization

averageUtilization: {{ .Values.autoscaling.targetCPUUtilizationPercentage }}

{{- end }}

✅ Delete unnecessary files: templates/serviceaccount.yaml and templates/httproute.yaml to avoid errors.

**7️⃣ Install Ingress Controller**

Docker Desktop doesn’t have it by default:

helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx

helm repo update

helm install nginx-ingress ingress-nginx/ingress-nginx

**8️⃣ Clean Up Old Releases**

If you tried multiple installs:

helm uninstall frontend --ignore-not-found

kubectl delete all -l app=frontend --ignore-not-found

**9️⃣ Install Frontend Helm Chart**

helm install frontend ./frontend

Check status:

kubectl get all

kubectl get ingress

kubectl get hpa

**🔟 Edit Hosts File (Windows)**

1. Open Notepad **as Administrator**
2. Open:

C:\Windows\System32\drivers\etc\hosts

1. Add line at the end:

127.0.0.1 frontend.local

1. Save and close

Test:

ping frontend.local

Open browser:  
[**http://frontend.local/**](http://frontend.local/) → React/Tomcat frontend should load.

**1️⃣1️⃣ Verify Autoscaling**

kubectl get hpa

Example:

NAME REFERENCE TARGETS MINPODS MAXPODS REPLICAS AGE

frontend Deployment/frontend 10%/70% 2 5 2 2m

**1️⃣2️⃣ Clean Redeploy (Optional)**

helm uninstall frontend

kubectl delete all -l app=frontend

helm install frontend ./frontend

✅ **All issues fixed:**

* serviceAccount error → set create: false
* httpRoute error → set enabled: false
* NodePort conflicts → changed to 30090
* Helm release conflicts → uninstalled old release
* Ingress annotation warning → replaced with spec.ingressClassName