Let's provide a step-by-step solution for creating, containerizing, and deploying a web-based calculator application to Kubernetes.

**1. Set up your Development Environment:**

* **Install Docker:** Download and install Docker Desktop (for Windows or macOS) or Docker Engine (for Linux).
* **Install kubectl:** Download and install the Kubernetes command-line tool (kubectl).
* **Set up a Kubernetes Cluster:**
  + **Minikube (Recommended for local development):** Install Minikube and start it:

Bash

minikube start

* + **Cloud Provider (GKE, EKS, AKS):** Create a Kubernetes cluster on your chosen cloud platform.

**2. Create the Calculator Application:**

Create a new directory (e.g., calculator-app) and create the following files inside it:

* **index.html:**

HTML

<!DOCTYPE html>

<html>

<head>

<title>Simple Calculator</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="calculator">

<input type="text" id="display" readonly>

<div class="buttons">

<button onclick="appendToDisplay('7')">7</button>

<button onclick="appendToDisplay('8')">8</button>

<button onclick="appendToDisplay('9')">9</button>

<button onclick="setOperator('+')">+</button><br>

<button onclick="appendToDisplay('4')">4</button>

<button onclick="appendToDisplay('5')">5</button>

<button onclick="appendToDisplay('6')">6</button>

<button onclick="setOperator('-')">-</button><br>

<button onclick="appendToDisplay('1')">1</button>

<button onclick="appendToDisplay('2')">2</button>

<button onclick="appendToDisplay('3')">3</button>

<button onclick="setOperator('\*')">\*</button><br>

<button onclick="appendToDisplay('0')">0</button>

<button onclick="appendToDisplay('.')">.</button>

<button onclick="calculate()">=</button>

<button onclick="setOperator('/')">/</button><br>

<button onclick="clearDisplay()">C</button>

</div>

</div>

<script src="script.js"></script>

</body>

</html>

* **style.css:**

CSS

.calculator {

width: 250px;

margin: 0 auto;

border: 1px solid #ccc;

padding: 10px;

}

#display {

width: 100%;

margin-bottom: 10px;

padding: 5px;

font-size: 18px;

text-align: right;

}

.buttons button {

width: 50px;

height: 50px;

margin: 2px;

font-size: 16px;

}

* **script.js:**

JavaScript

let display = document.getElementById('display');

let currentInput = '';

let operator = '';

let previousInput = '';

function appendToDisplay(value) {

currentInput += value;

display.value = currentInput;

}

function setOperator(op) {

if (currentInput !== '') {

previousInput = currentInput;

operator = op;

currentInput = '';

}

}

function calculate() {

let result;

if (previousInput !== '' && currentInput !== '' && operator !== '') {

try {

result = eval(previousInput + operator + currentInput);

display.value = result;

currentInput = result.toString();

previousInput = '';

operator = '';

} catch (error) {

display.value = 'Error';

currentInput = '';

previousInput = '';

operator = '';

}

}

}

function clearDisplay() {

display.value = '';

currentInput = '';

previousInput = '';

operator = '';

}

**3. Create the Dockerfile:**

In the same calculator-app directory, create a file named Dockerfile (no extension):

Dockerfile

FROM httpd:2.4-alpine

COPY . /usr/local/apache2/htdocs/

EXPOSE 80

**4. Build the Docker Image:**

Open a terminal in the calculator-app directory and run:

Bash

docker build -t calculator-app .

**5. Create Kubernetes Deployment and Service YAML Files:**

* **calculator-deployment.yaml:**

YAML

apiVersion: apps/v1

kind: Deployment

metadata:

name: calculator-deployment

spec:

replicas: 3

selector:

matchLabels:

app: calculator

template:

metadata:

labels:

app: calculator

spec:

containers:

- name: calculator

image: calculator-app

ports:

- containerPort: 80

* **calculator-service.yaml:**

YAML

apiVersion: v1

kind: Service

metadata:

name: calculator-service

spec:

type: LoadBalancer # For external access (Minikube uses minikube tunnel)

ports:

- protocol: TCP

port: 80 # Port exposed by the service

targetPort: 80 # Port the container is listening on

selector:

app: calculator

**6. Deploy to Kubernetes:**

Open a terminal and navigate to the directory containing the YAML files. Then run:

Bash

kubectl apply -f calculator-deployment.yaml

kubectl apply -f calculator-service.yaml

**7. Access the Application:**

* **Minikube:**

Bash

minikube service calculator-service

This command opens your default web browser and forwards the service port to your local machine.

* **Cloud Provider (GKE, EKS, AKS):**

Bash

kubectl get service calculator-service

Look for the EXTERNAL-IP or HOSTNAME in the output. Use this address in your browser.

**Complete Step-by-Step Summary:**

1. **Environment Setup:** Docker, kubectl, Minikube (or cloud cluster).
2. **Application Code:** Create index.html, style.css, and script.js.
3. **Dockerfile:** Create the Dockerfile to containerize the application.
4. **Build Docker Image:** docker build -t calculator-app .
5. **Kubernetes YAML:** Create calculator-deployment.yaml and calculator-service.yaml.
6. **Deploy to Kubernetes:** kubectl apply -f calculator-deployment.yaml and kubectl apply -f calculator-service.yaml
7. **Access Application:** minikube service calculator-service (Minikube) or get the EXTERNAL-IP of the service (cloud).

This detailed, step-by-step solution should make it easy to deploy your calculator application to Kubernetes. If you face any issues, double-check the console output for error messages and ensure that all prerequisites are correctly installed.