### Setting up Backstage on Google Kubernetes Engine (GKE)

**Prerequisites**

1. **Google Cloud Account**: Ensure you have a Google Cloud account and a project created.
2. **Billing Enabled**: Make sure billing is enabled for your project.
3. **Google Cloud SDK**: Install the [Google Cloud SDK](https://cloud.google.com/sdk/docs/install) if using the CLI.
4. **kubectl**: Install kubectl (included with the Google Cloud SDK).

**Step 1: Create a GKE Cluster**

**Option 1: Using Google Cloud Console**

1. Go to the [Google Cloud Console](https://console.cloud.google.com/).
2. Navigate to **Kubernetes Engine > Clusters**.
3. Click **Create Cluster**.
4. Choose the **Standard** cluster type.
5. Configure the cluster:
   * **Name**: backstage-cluster
   * **Location Type**: Zonal (choose a zone, e.g., us-central1-a).
   * **Node Pool**: Use the default settings (1 node, e2-medium machine type).
6. Click **Create**.

**Option 2: Using Google Cloud CLI**

Run the following command to create a GKE cluster:

gcloud container clusters create backstage-cluster \

--zone us-central1-a \

--num-nodes 1 \

--machine-type e2-medium

**Step 2: Connect to the GKE Cluster**

**Option 1: Using Google Cloud Console**

1. After the cluster is created, go to **Kubernetes Engine > Clusters**.
2. Click the **Connect** button next to your cluster.
3. Copy the gcloud command provided and run it in your terminal.

**Option 2: Using Google Cloud CLI**

Run the following command to configure kubectl to connect to your cluster:

gcloud container clusters get-credentials backstage-cluster \

--zone us-central1-a

**Step 3: Deploy Backstage**

**Option 1: Using Helm (Simplest Approach)**

1. Install Helm:

curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash

1. Add the Backstage Helm chart repository:

helm repo add backstage https://backstage.github.io/charts

helm repo update

1. Install Backstage:

helm install backstage backstage/backstage

1. Verify the deployment:

kubectl get pods

**Option 2: Using Kubernetes Manifests**

1. Create a backstage.yaml file with the following content:

apiVersion: apps/v1

kind: Deployment

metadata:

name: backstage

spec:

replicas: 1

selector:

matchLabels:

app: backstage

template:

metadata:

labels:

app: backstage

spec:

containers:

- name: backstage

image: backstage/backstage:latest

ports:

- containerPort: 3000

---

apiVersion: v1

kind: Service

metadata:

name: backstage-service

spec:

selector:

app: backstage

ports:

- protocol: TCP

port: 80

targetPort: 3000

type: LoadBalancer

1. Apply the manifest:

kubectl apply -f backstage.yaml

**Step 4: Access Backstage**

1. Get the external IP of the Backstage service:

kubectl get svc backstage-service

1. Open the external IP in your browser (e.g., http://<EXTERNAL\_IP>).

gcloud container clusters delete backstage-cluster \

--zone us-central1-a