Deploying Application with Domain Pointing on Ingress and Google Managed Certificate Attachment

Introduction

This document outlines the steps to deploy an application on Kubernetes with a domain pointing to an Ingress resource, and attaching a Google Managed Certificate for HTTPS access.

Prerequisites

- Access to Google Cloud Platform (GCP) console
- gcloud CLI installed and configured
- Kubernetes cluster set up on GCP
- Domain registered and managed in Cloud DNS

Step-by-Step Deployment

1. Reserve Static IP Address

First, reserve a static IP in GCP to use with the Ingress resource.

```
gcloud compute addresses create nginxexample --global
```

Verify the creation and obtain details of the reserved static IP.

```
gcloud compute addresses describe nginxexample --global
```

2. Create Kubernetes Manifest File

Create a Kubernetes manifest file (nginx-deployment.yaml) containing the deployment, service, Ingress, and ManagedCertificate sections. Replace placeholders with actual configurations and your reserved static IP.

```
# Insert provided YAML here

apiVersion: apps/v1

kind: Deployment

metadata:
  name: nginx
  namespace: flowise
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: nginx-svc
  namespace: flowise
```

```yaml
spec:
  ports:
  - port: 80
    targetPort: 80
    protocol: TCP
    name: http
  selector:
    app: nginx
  type: LoadBalancer
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx-ingress
  namespace: flowise
  annotations:
    kubernetes.io/ingress.class: "gce"
    kubernetes.io/ingress.global-static-ip-name: nginxexample
    networking.gke.io/managed-certificates: managed-cert
spec:
  rules:
  - host: dev.flow.disearch.ai
    http:
      paths:
      - path: /*
        pathType: ImplementationSpecific
        backend:
          service:
            name: nginx-svc
            port:
              number: 80
---
apiVersion: networking.gke.io/v1
kind: ManagedCertificate
metadata:
  name: managed-cert
  namespace: flowise
spec:
  domains:
  - dev.flow.disearch.ai
```

3. Deployment and Access

Deploy the application using the created manifest file.

```
kubectl apply -f nginx-deployment.yaml
```

For verfication access the application using the Kubernetes Service external IP, which should be your LoadBalancer IP.

4. Configure Ingress and Domain

Get the Ingress external IP and update Cloud DNS in the public zone to point your domain to this IP.

```
kubectl get ingress -n flowise nginx-ingress
```

5. Verification and Troubleshooting

Check the deployed resources in Kubernetes and GCP for verification and troubleshooting.

```
kubectl get svc -n flowise nginx-svc
```

```
kubectl get ingress -n flowise nginx-ingress
```

kubectl get managedcertificates -n flowise managed-cert --- > it should show ACTIVE

Also, verify resources in GCP console for more clarity.

Remember

- Ensure proper naming and labeling of resources.
- Replace placeholders with actual values in the manifest file.
- Verify DNS propagation after updating Cloud DNS settings.
- Troubleshoot any issues using Kubernetes and GCP resources.
- GCP Managed SSL Certificate will take 10 to 20 mints minimum for successfully provioning.
- Once SSL Certificate will provision successfully. you can access your application through browser with HTTPS as well.. Sometime it take few minuts for getting up..