

REMLA Application Stack Deployment Guide

This guide provides step-by-step instructions to deploy the complete REMLA application stack, including the frontend, backend services, and a full monitoring suite (Prometheus & Grafana), to a local Kubernetes cluster using Minikube and Helm.

Prerequisites

Before you begin, ensure you have the following command-line tools installed and configured on your system:

- * **Minikube:** For running a local Kubernetes cluster.
- * **kubectl:** The Kubernetes command-line tool.
- * **Helm:** The package manager for Kubernetes.

Deployment Instructions

Step 1: Start Your Minikube Cluster

First, start your local Minikube cluster and enable the NGINX Ingress addon, which is required to route external traffic to your services.

```
```bash
Start the Minikube cluster
minikube start

Enable the NGINX Ingress addon
minikube addons enable ingress
```

The Ingress controller may take a moment to initialize. You can monitor its status with `kubectl get pods -n ingress-nginx --watch` and wait for it to be ready.

### Step 2: Install the Application with Helm

The entire application stack is packaged as a Helm chart. You will install this chart into a dedicated monitoring namespace.

1. **Navigate to the Helm chart directory:**  
`cd project/operation/k8s/remla-chart`
2. Create the monitoring namespace:  
This namespace will contain all your application and monitoring components.  
`kubectl create namespace monitoring`
3. Install the Helm chart:  
This command deploys all resources (services, deployments, monitoring stack, etc.) into

the monitoring namespace. We will give our deployment a release name of remla-app.  
helm install remla-app . -n monitoring

### Step 3: Configure Host and Access the Application

To access the main application, you need to map the application's hostname (remla.local) to your Minikube cluster's IP address.

1. **Get your Minikube IP address:**

```
minikube ip
```

2. Add an entry to your /etc/hosts file:

You will need administrative privileges to edit this file. Open it with your favorite editor (e.g., `sudo nano /etc/hosts`) and add the following line, replacing `<your-minikube-ip>` with the IP address from the previous command.

```
<your-minikube-ip> remla.local
```

3. Access the application:

Open your web browser and navigate to `http://remla.local`. You should see the Sentiment Analysis application frontend.

### Step 4: Access Monitoring Dashboards (Prometheus & Grafana)

The Prometheus and Grafana services are exposed directly via NodePort services. You can get their unique URLs using the minikube service command.

1. **Get the Grafana URL:**

```
minikube service remla-app-grafana -n monitoring --url
```

Open the output URL in your browser to access the Grafana dashboard. You can find the "REMLA App Service" dashboard in the "Dashboards" section.

2. **Get the Prometheus URL:**

```
minikube service remla-app-kube-prometheus-prometheus -n monitoring --url
```

Open the output URL to access the Prometheus UI, where you can explore and query all collected metrics.

### Step 5: Test the Application

Now that everything is running, you can test the full functionality:

1. Go to **`http://remla.local`**.
2. Enter a review (e.g., "The food was amazing!") and click "Analyze Sentiment".
3. The application should display a prediction (e.g., "Positive") and a confidence score.
4. Go to your Grafana dashboard. You should see the panels populating with data reflecting your interactions.

# Cleaning Up

Once you are finished, you can remove all the deployed resources and shut down your local cluster.

1. Uninstall the Helm release:

This will delete all the Kubernetes resources created by the chart.

```
helm uninstall remla-app -n monitoring
```

2. Delete the Minikube cluster:

This will completely stop and delete the local Kubernetes cluster.

```
minikube delete
```