**Grafana, Loki and Promtail Documentation**

**Deploying MinIO, Grafana, Promtail, and Loki on Kubernetes**

This guide outlines the step-by-step process to deploy MinIO, Grafana, Promtail, and Loki on a Kubernetes cluster.

**Step 1: Create Kubernetes Namespaces**

Namespaces help organize resources.

**kubectl create ns test**

**kubectl create ns grafana**

**Step 2: Deploy MinIO**

Deploy MinIO using the provided YAML configuration.

**kubectl apply -f minio-deployment.yaml**

Expose MinIO for local access:

**kubectl port-forward svc/minio-service 9000:9000 9001:9001 -n test**

* MinIO Web UI will be accessible at http://localhost:9001
* MinIO S3 API will be available at http://localhost:9000

**Step 3: Deploy Grafana**

Add the Grafana Helm repository:

**helm repo add grafana https://grafana.github.io/helm-charts**

**helm repo update**

Apply the Grafana deployment manifest:

**kubectl apply -f grafana-deployment.yaml**

Expose Grafana for local access:

**kubectl port-forward -n grafana svc/grafana-service 3000:80 -n grafana**

* Access Grafana at http://localhost:3000

**Step 4: Install Promtail**

Install Promtail using Helm:

**helm upgrade --install promtail grafana/promtail -n grafana --create-namespace -f promtail-values.yaml**

**Step 5: Create a Service Account for Loki**

Apply the service account configuration for Loki to access MinIO:

**kubectl apply -f loki-serviceaccount.yaml**

**Step 6: Deploy Loki**

Install Loki using Helm:

**helm upgrade --install loki grafana/loki -n grafana --create-namespace -f loki-values.yaml --set loki.auth\_enabled=false**

**Step 7: Deploy Random Logger Application**

**1. Create dev namespace**

**kubectl create ns dev**

**2. Build the Docker Image**

Navigate to the **Random Logger** application directory and build the Docker image:

**docker build -t random-logger .**

**3. Create a New Helm Chart**

Run the following command to create a new Helm chart named random-logger:

**helm create random-logger**

This command generates the basic Helm chart structure in a new directory called random-logger.

**4. Modify values.yaml**

Edit the default values.yaml with following:

**image:**

**repository: random-logger  # Use the local image name**

**tag: latest**

**pullPolicy: IfNotPresent  # Use local image**

**5. Install the Random Logger using Helm**

Navigate to the Helm chart directory and install the application in dev namespace:

**helm install random-logger . -n dev**

**Summarys**

* MinIO stores logs in S3-csompatible storage.
* Promtail collects logs and sends them to Loki.
* Loki indexes and stores logs for querying in Grafana.
* Grafana visualizes logs from Loki.
* **Random Logger Application** generates logs to test the pipeline.

This setup ensures an efficient logging pipeline within the Kubernetes cluster.