**Loki Prometheus and Grafana on File Upload server app:**  
  
  
Build App:

docker build -t deepakk2212/upload-server:v3 .

docker push deepakk2212/upload-server:v3

kubectl apply -f deployment.yaml

Then install promtail to forward logs to loki:

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

helm upgrade --install promtail grafana/promtail \

--namespace monitoring --create-namespace \

--set "loki.serviceName=loki" \

--set "loki.servicePort=3100"

Upload server:  
  
kubectl port-forward svc/upload-server-service 8080:80

Loki Grafana Prometheus DevOps

Installing Loki:

kubectl create namespace monitoring

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

helm search repo grafana

helm install loki grafana/loki --namespace monitoring --version 6.29.0 --values manifests/loki-values.yaml

kubectl port-forward --namespace monitoring svc/loki-gateway 3100:80

kubectl get pods -n monitoring

kubectl get svc -n monitoring

Installing:

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

helm search repo prometheus-community

helm install prometheus prometheus-community/kube-prometheus-stack --namespace monitoring --version 70.7.0 --values manifests/prometheus-values.yaml

kubectl get svc -n monitoring

kubectl port-forward -n monitoring svc/prometheus-grafana 3000:80

Pometheus port-forwarding:

kubectl port-forward svc/prometheus-kube-prometheus-prometheus -n monitoring 9090:9090

Loki Queries:

{app="upload-server"}

{app="upload-server"} |= "upload"

{app="upload-server"} != "health"

**Get Log Counts (for alerting or dashboards):**

count\_over\_time({app="upload-server"}[5m])

rate({app="upload-server"}[1m])

**PromQL Queries for Kubernetes & upload-server**

**(Tested & Working)**

1. CPU Usage of upload-server Pod

Measures the CPU usage rate over the past 5 minutes.

rate(container\_cpu\_usage\_seconds\_total{pod=~"upload-server.\*", container!="POD"}[5m])

2. Memory Usage of upload-server Pod

Shows current memory usage in bytes.

container\_memory\_usage\_bytes{pod=~"upload-server.\*", container!="POD"}

3. Pod Uptime

Calculates how long the pod has been running.

time() - kube\_pod\_start\_time{pod=~"upload-server.\*"}

4. Pod Status (Running)

Checks if the pod is currently in the "Running" state.

kube\_pod\_status\_phase{pod=~"upload-server.\*", phase="Running"}

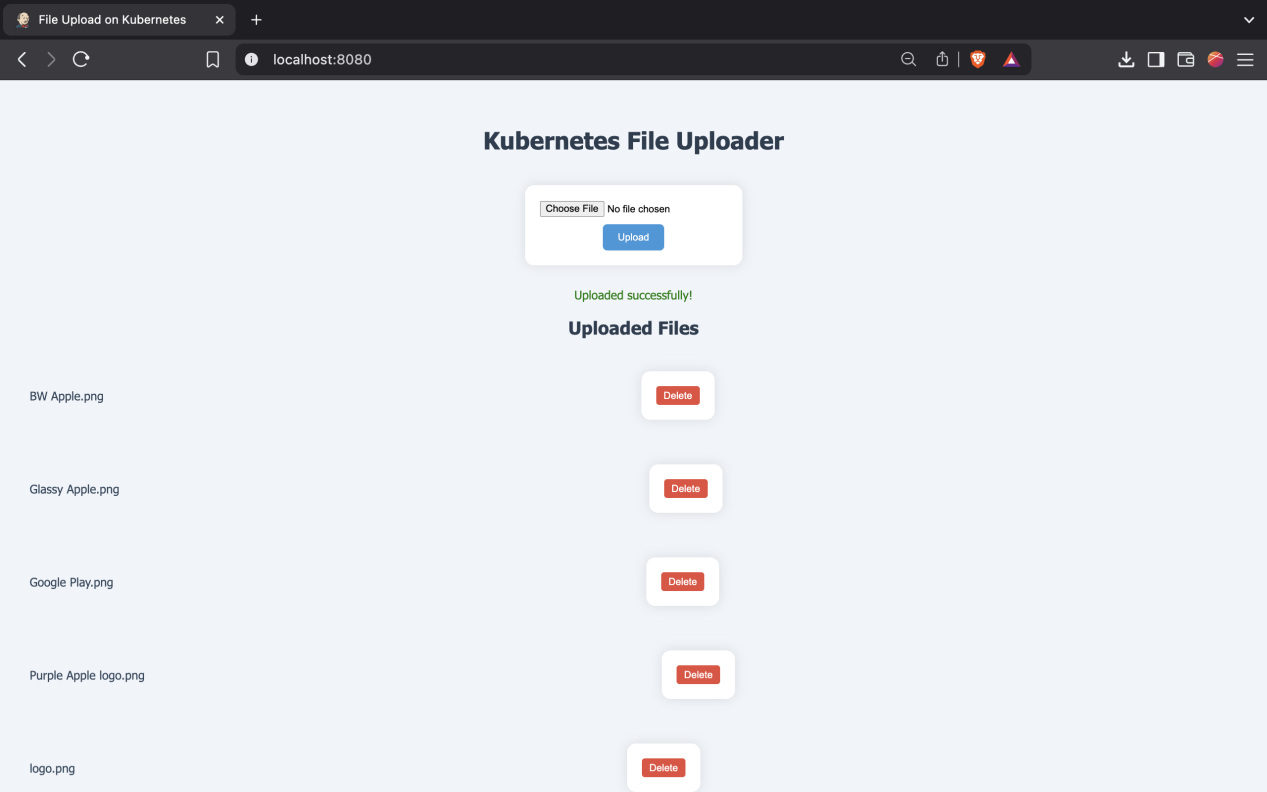
5. Disk I/O (Read + Write)

Shows the rate of disk reads and writes in bytes per second.

rate(container\_fs\_reads\_bytes\_total{pod=~"upload-server.\*"}[5m]) +

rate(container\_fs\_writes\_bytes\_total{pod=~"upload-server.\*"}[5m])

File Upload WebApp:



Screenshots:  
  
