# Grafana + Loki + Prometheus + cAdvisor – Aplicación

Repositorio

<https://github.com/jvicmar95/ALL/tree/main/APLICACION>

## Despliegues

### Desplegar Loki

kubectl create namespace monitoring

kubectl apply -f pvc-loki.yaml -n monitoring

kubectl apply -f loki-config.yaml -n monitoring

kubectl apply -f loki-service.yaml -n monitoring

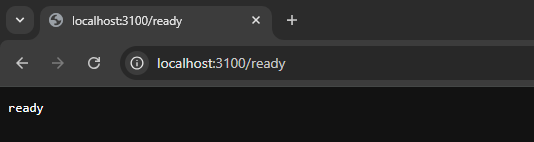
kubectl apply -f loki-deployment.yaml -n monitoring

#### Verificacion

kubectl get pods -n monitoring -l app=loki

kubectl port-forward svc/loki 3100:3100 -n monitoring

<http://localhost:3100/ready>



### Desplegar Promtail

kubectl apply -f promtail-config.yaml -n monitoring

kubectl apply -f promtail-serviceaccount.yaml -n monitoring

kubectl apply -f promtail-daemonset.yaml -n monitoring

### Desplegar Prometheus

kubectl apply -f pvc-prometheus.yaml -n monitoring

kubectl apply -f pvc-alertmanager.yaml -n monitoring

kubectl apply -f prometheus-config.yaml -n monitoring

kubectl apply -f prometheus-service.yaml -n monitoring

kubectl apply -f prometheus-deployment.yaml -n monitoring

kubectl apply -f kube-state-metrics-serviceaccount.yaml -n monitoring

kubectl apply -f kube-state-metrics-clusterrole.yaml

kubectl apply -f kube-state-metrics-clusterrolebinding.yaml

kubectl apply -f kube-state-metrics-service.yaml -n monitoring

kubectl apply -f kube-state-metrics-deployment.yaml -n Monitoring

### Desplegar cAdvisor

kubectl apply -f cadvisor-service.yaml

kubectl apply -f cadvisor-daemonset.yaml

### Desplegar grafana

kubectl apply -f pvc-grafana.yaml -n monitoring

kubectl apply -f grafana-config.yaml -n monitoring

kubectl apply -f grafana-datasources.yaml -n monitoring

kubectl apply -f grafana-service.yaml -n monitoring

kubectl apply -f grafana-deployment.yaml -n monitoring

Probar

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

http://localhost:3000

### Desplegar aplicación

kubectl create namespace aplicación

kubectl apply -f pvc-aplicacion.yaml -n aplicacion

kubectl apply -f service-aplicacion.yaml -n aplicacion

kubectl apply -f deployment-aplicacion.yaml -n aplicación

Probar

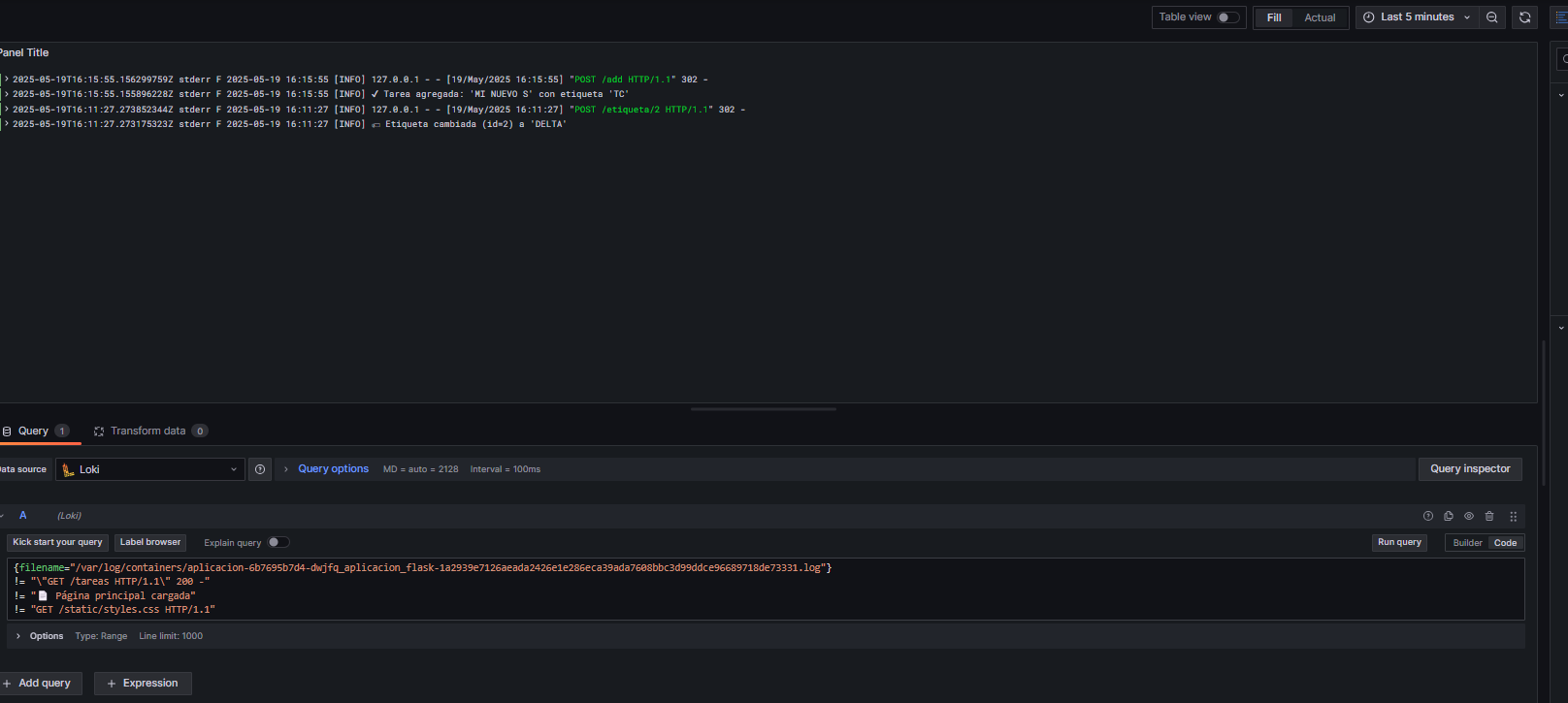
kubectl port-forward svc/aplicacion-service 8080:80 -n aplicación

http://localhost:8080

## Grafana

### Crear grafico logs

Filtramos solo los que queremos, quitamos todo lo que sea actualizar pagina



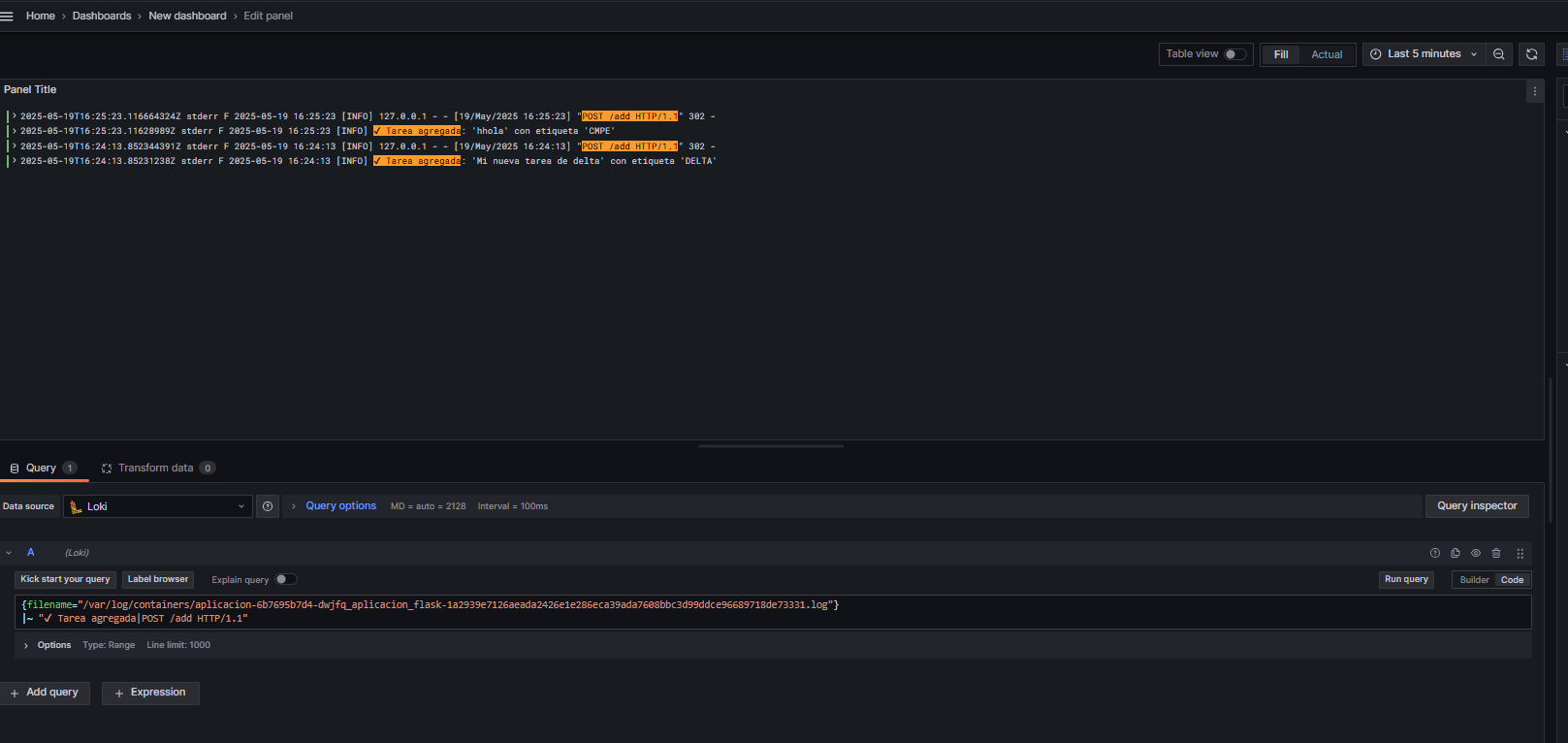
{filename="/var/log/containers/aplicacion-6b7695b7d4-dwjfq\_aplicacion\_flask-1a2939e7126aeada2426e1e286eca39ada7608bbc3d99ddce96689718de73331.log"}

!= "\"GET /tareas HTTP/1.1\" 200 -"

!= "📄 Página principal cargada"

!= "GET /static/styles.css HTTP/1.1"

### Grafico Alta de notas



{filename="/var/log/containers/aplicacion-6b7695b7d4-dwjfq\_aplicacion\_flask-1a2939e7126aeada2426e1e286eca39ada7608bbc3d99ddce96689718de73331.log"}

|~ "✔ Tarea agregada|POST /add HTTP/1.1"

{filename="/var/log/containers/aplicacion-6b7695b7d4-dwjfq\_aplicacion\_flask-1a2939e7126aeada2426e1e286eca39ada7608bbc3d99ddce96689718de73331.log"}

|~ "✔ Tarea agregada"

count\_over\_time({filename="/var/log/containers/aplicacion-6b7695b7d4-dwjfq\_aplicacion\_flask-1a2939e7126aeada2426e1e286eca39ada7608bbc3d99ddce96689718de73331.log"}

|~ "✔ Tarea agregada" [24h])

{filename="/var/log/containers/aplicacion-6b7695b7d4-dwjfq\_aplicacion\_flask-1a2939e7126aeada2426e1e286eca39ada7608bbc3d99ddce96689718de73331.log"}

|~ "❌ Tarea eliminada"

count\_over\_time({filename="/var/log/containers/aplicacion-6b7695b7d4-dwjfq\_aplicacion\_flask-1a2939e7126aeada2426e1e286eca39ada7608bbc3d99ddce96689718de73331.log"}

|~ "❌ Tarea eliminada " [24h])

{filename="/var/log/containers/aplicacion-6b7695b7d4-dwjfq\_aplicacion\_flask-1a2939e7126aeada2426e1e286eca39ada7608bbc3d99ddce96689718de73331.log"}

|~ "🗂 Backup creado"

count\_over\_time({filename="/var/log/containers/aplicacion-6b7695b7d4-dwjfq\_aplicacion\_flask-1a2939e7126aeada2426e1e286eca39ada7608bbc3d99ddce96689718de73331.log"}

|~ "✔ Tarea agregada" [24h])

kubectl create configmap dash-aplicacion-logs --from-file=logs-aplicacion.json="GRAFANALL/Dashboard/Logs aplicacion-1747672903732.json" -n monitoring --dry-run=client -o yaml > dash-aplicacion-logs-configmap.yaml

## cAdvisor

📊 Tabla resumen de métricas clave de cAdvisor

| **Categoría** | **Métrica Prometheus** | **Descripción** | **Consulta útil** |
| --- | --- | --- | --- |
| **CPU** | container\_cpu\_usage\_seconds\_total | Tiempo total de CPU usado (en segundos acumulados) | rate(container\_cpu\_usage\_seconds\_total{name!~".\*POD.\*"}[5m]) |
|  | container\_cpu\_cfs\_throttled\_seconds\_total | Tiempo que el contenedor fue limitado por CPU (throttled) | rate(container\_cpu\_cfs\_throttled\_seconds\_total[5m]) |
|  | container\_cpu\_cfs\_periods\_total | Número de periodos en los que se evaluó el uso de CPU | rate(container\_cpu\_cfs\_periods\_total[5m]) |
| **Memoria** | container\_memory\_usage\_bytes | Total de memoria usada por el contenedor | container\_memory\_usage\_bytes{name!~".\*POD.\*"} |
|  | container\_memory\_working\_set\_bytes | Memoria en uso activo (excluye caché liberable) | container\_memory\_working\_set\_bytes{name!~".\*POD.\*"} |
|  | container\_memory\_rss | Memoria residente física (RAM real) | container\_memory\_rss{name!~".\*POD.\*"} |
|  | container\_memory\_failcnt | Fallos al asignar memoria (OOM events) | container\_memory\_failcnt{name!~".\*POD.\*"} |
| **Disco** | container\_fs\_usage\_bytes | Bytes usados en el sistema de archivos del contenedor | container\_fs\_usage\_bytes{name!~".\*POD.\*"} |
|  | container\_fs\_reads\_bytes\_total | Bytes leídos del disco por el contenedor | rate(container\_fs\_reads\_bytes\_total[5m]) |
|  | container\_fs\_writes\_bytes\_total | Bytes escritos al disco por el contenedor | rate(container\_fs\_writes\_bytes\_total[5m]) |
| **Red** | container\_network\_receive\_bytes\_total | Total de bytes recibidos por red | rate(container\_network\_receive\_bytes\_total[5m]) |
|  | container\_network\_transmit\_bytes\_total | Total de bytes enviados por red | rate(container\_network\_transmit\_bytes\_total[5m]) |
|  | container\_network\_receive\_errors\_total | Errores al recibir tráfico de red | rate(container\_network\_receive\_errors\_total[5m]) |
|  | container\_network\_transmit\_errors\_total | Errores al transmitir tráfico de red | rate(container\_network\_transmit\_errors\_total[5m]) |