

# 03 - Monitoreo de aplicación

Tipo	Area Temática	Duración
Taller	Introducción a sistemas de monitoreo	1 hora

## Tema:

Taller de Monitoreo Básico con Prometheus y Grafana en Kubernetes

## Objetivo:

El objetivo de este taller es que los estudiantes configuren un sistema de monitoreo básico para una aplicación desplegada en Kubernetes. Usarán Prometheus para recopilar métricas y Grafana para visualizarlas. Al finalizar, deberán ser capaces de interpretar las métricas básicas y generar visualizaciones útiles para el monitoreo de la aplicación.

## Instrucciones

1. Sigue cada uno de los pasos indicados para instalar y configurar Prometheus y Grafana en tu clúster de Kubernetes.
2. Configura la recolección de métricas para la aplicación previamente instalada.
3. Crea dashboards en Grafana que permitan visualizar el estado y rendimiento de la aplicación.
4. Documenta cada paso y justifica las configuraciones realizadas.

# Paso 1: Configuración de Prometheus

## 1. Instalar Prometheus en el clúster de Kubernetes:

- Puedes instalar Prometheus utilizando Helm, un gestor de paquetes para Kubernetes:

```
helm repo add prometheus-community <https://prometheus-com
munity.github.io/helm-charts>
helm repo update
helm install prometheus prometheus-community/prometheus
```

## 2. Verificar la instalación:

- Asegúrate de que Prometheus esté corriendo:

```
kubectl get pods -l "app=prometheus"
```

## 3. Acceder a la interfaz de Prometheus:

- Encuentra el servicio de Prometheus y expónlo localmente para acceder a la interfaz web:

```
kubectl port-forward <prometheus-pod-name> 9090
```

- Accede a <http://localhost:9090> desde tu navegador.

## 4. Configurar Prometheus para recolectar métricas:

- Revisa el archivo de configuración de Prometheus ( `prometheus.yaml` ) y asegúrate de que esté configurado para recolectar métricas de tus aplicaciones y nodos del clúster.
- Añade tu aplicación si no está incluida:

```
scrape_configs:
  - job_name: 'web-app'
```

```
static_configs:
  - targets: ['<web-app-service>:8080']
```

## 5. Aplicar la configuración:

- Si hiciste cambios, actualiza la configuración de Prometheus y reinicia el pod:

```
kubectl delete pod <prometheus-pod-name>
```

# Paso 2: Configuración de Grafana

## 1. Instalar Grafana en el clúster de Kubernetes:

- Al igual que con Prometheus, usa Helm para instalar Grafana:

```
helm install grafana grafana/grafana
```

## 2. Acceder a la interfaz de Grafana:

- Encuentra el servicio de Grafana y expónlo localmente:

```
kubectl port-forward <grafana-pod-name> 3000
```

- Accede a `http://localhost:3000` desde tu navegador.
- Las credenciales por defecto son `admin/admin`.

## 3. Agregar Prometheus como fuente de datos en Grafana:

- Navega a `Configuration > Data Sources` y selecciona Prometheus.
- Introduce la URL de Prometheus (por ejemplo, `http://prometheus-server.default.svc.cluster.local:9090`) y guarda la configuración.

## 4. Importar un dashboard prediseñado:

- Grafana ofrece dashboards preconfigurados para Kubernetes. Importa uno de estos para monitorear las métricas básicas del clúster y la aplicación.

- Ve a `Dashboards > Manage > Import` y usa el ID de un dashboard desde la comunidad de Grafana, por ejemplo, el ID `6417` (Kubernetes cluster monitoring) o `3119` (Kubernetes/Prometheus).

#### 5. Crear un dashboard personalizado:

- Crea un nuevo dashboard que muestre métricas clave para tu aplicación, como uso de CPU, memoria, número de requests, y latencia.
- Usa las consultas PromQL de Prometheus para definir los paneles.

## Paso 3: Monitoreo y Alerta

### 1. Establecer alertas básicas en Prometheus:

- Configura alertas en Prometheus para detectar umbrales críticos, como alto uso de CPU o memoria.
- Añade reglas de alerta en el archivo `prometheus.yaml`:

```
groups:
- name: example
  rules:
  - alert: HighCPUUsage
    expr: 100 - (avg by (instance) (rate(node_cpu_seconds_total{job="node_exporter",mode="idle"}[5m])) * 100) > 80
    for: 5m
    labels:
      severity: warning
    annotations:
      summary: "High CPU usage detected"
```

### 2. Visualizar y analizar las métricas:

- Usa Grafana para monitorear en tiempo real las métricas de la aplicación.
- Detecta cualquier anomalía o problema de rendimiento y documenta tus hallazgos.

### 3. Probar el sistema de alertas:

- Genera carga en la aplicación para activar las alertas configuradas y verifica que se activen correctamente en Prometheus.
- 

## Conclusión

Al finalizar este taller, los estudiantes deberán ser capaces de configurar un entorno de monitoreo básico en Kubernetes utilizando Prometheus y Grafana. Deberán también interpretar las métricas obtenidas y utilizar las visualizaciones para detectar y resolver posibles problemas en la infraestructura.