### **Pasos para instalar el stack Grafana-Prometheus en Kubernetes**

**Prerrequisitos:**

Habilitar métricas:

kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml

kubectl get deployment metrics-server -n kube-system

/!\ Nota: Si sale algún error relacionado a validación TLS, por favor ejecute el siguiente procedimiento:

kubectl edit deployment metrics-server -n kube-system

Agregue los siguientes argumentos bajo la sección "args" el el bloque de código "container spec":

- --kubelet-insecure-tls

- --kubelet-preferred-address-types=InternalIP,Hostname,ExternalIP

Debería quedar similar al siguiente ejemplo:

---

spec:

containers:

- args:

- --cert-dir=/tmp

- --secure-port=10250

- --kubelet-use-node-status-port

- --metric-resolution=15s

- --kubelet-insecure-tls # <== Nueva Línea

- --kubelet-preferred-address-types=InternalIP,Hostname,ExternalIP # <=== Nueva Línea

---

#### **1. Preparar el entorno**

Asegúrese de tener instalado lo siguiente:

* Un clúster de Kubernetes funcional.
* kubectl configurado para conectarse al clúster.
* Helm instalado (compatible con la versión de su clúster).

Actualice los repositorios de Helm:

helm repo add prometheus-community https://prometheus-community.github.io/helm-charts

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

helm repo update

#### **2. Crear un namespace para el monitoreo**

Se recomienda crear un namespace dedicado para separar los recursos del stack.

kubectl create namespace monitoring

#### **3. Instalar Prometheus**

Utilice el chart de Helm kube-prometheus-stack, que incluye Prometheus, Alertmanager, Grafana y otros componentes preconfigurados.

helm install prometheus prometheus-community/kube-prometheus-stack \

--namespace monitoring

Esto instalará:

* Prometheus Server.
* Prometheus Operator.
* Grafana (con dashboards preconfigurados).
* Alertmanager.

#### **4. Verificar la instalación**

Verifique los pods en el namespace monitoring:

kubectl get pods -n monitoring

Espere a que todos los pods estén en estado Running.

Acceda al servicio de Grafana:

* + Localice el servicio de Grafana:
    - kubectl get svc -n monitoring | grep grafana
  + Use el comando kubectl port-forward para acceder localmente:
    - kubectl port-forward -n monitoring svc/prometheus-grafana 3000:80
  + Abra un navegador y visite http://localhost:3000.

**Credenciales predeterminadas**:

* + - Usuario: admin
    - Contraseña: prom-operator (puede verificarse en el secreto generado por Grafana).

#### **5.** Configurar Loki

helm install loki grafana/loki-stack --namespace monitoring \

--set promtail.enabled=true

Configure el datasource en Grafana con el siguiente URL:

<http://loki.<namespace>.svc.cluster.local:3100>

**Ejemplo:**

http://loki.monitoring.svc.cluster.local:3100

#### **Configurar almacenamiento persistente para Loki (opcional)**

Para evitar la pérdida de logs, configure almacenamiento persistente en el despliegue de Loki. Cree un archivo values-loki.yaml:

loki:

persistence:

enabled: true

size: 50Gi

storageClassName: standard

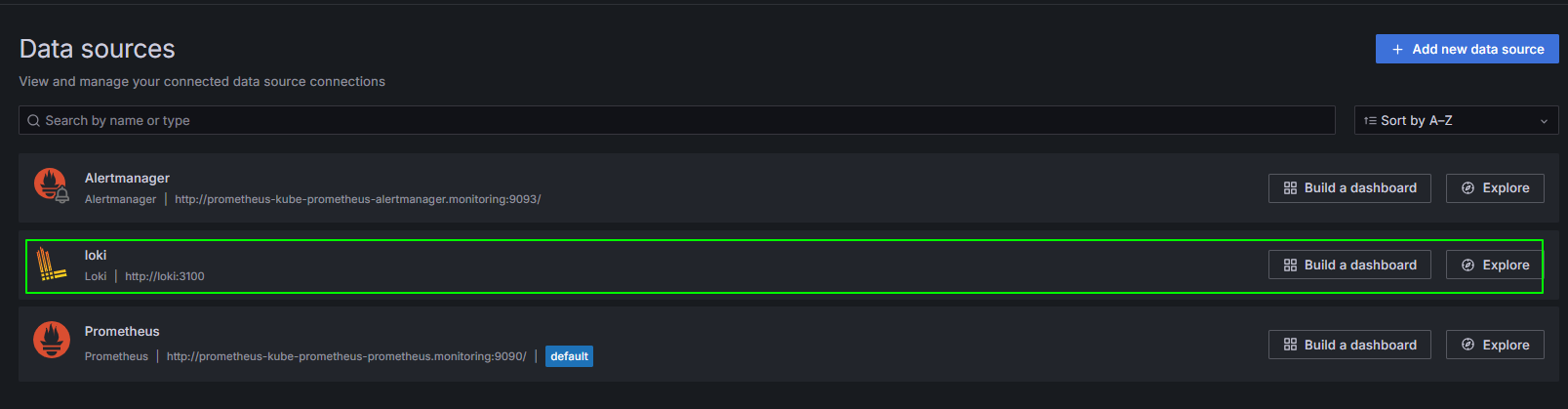
Actualice la instalación de Loki usando este archivo:

helm upgrade --install loki grafana/loki-stack \

--namespace monitoring \

-f values-loki.yaml

El datasource se configurará automáticamente y quedará dela siguiente manera:



#### **6. Configurar dashboards adicionales**

* + Puede agregar dashboards personalizados o importar dashboards desde el sitio oficial de Grafana (Grafana Dashboards).
  + Configure Prometheus como fuente de datos en Grafana (aunque suele estar preconfigurado con este stack).

#### **6. Configurar almacenamiento persistente (opcional)**

Para asegurarse de que los datos de Prometheus y Grafana sean persistentes, agregue almacenamiento persistente configurando valores personalizados en un archivo YAML.

Ejemplo de archivo values.yaml para el chart de Helm:

grafana:

persistence:

enabled: true

size: 10Gi

prometheus:

server:

persistentVolume:

enabled: true

size: 50Gi

Instale o actualice el stack con este archivo:

helm upgrade --install prometheus prometheus-community/kube-prometheus-stack \

--namespace monitoring \

-f values.yaml

#### **7. Configurar alertas (opcional)**

El stack incluye Alertmanager para manejar alertas. Configure reglas de alerta modificando los ConfigMaps o añadiendo archivos YAML específicos.

#### **Referencias**

* [Helm Chart de kube-prometheus-stack](https://github.com/prometheus-community/helm-charts/tree/main/charts/kube-prometheus-stack)
* Documentación oficial de Grafana