Kubernetes

Data Infrastructure Insights

NetApp
February 03, 2026

Tabla de contenidos

Kubernetes

Descripción general del clúster de Kubernetes

El Kubernetes Explorer de Data Infrastructure Insights es una herramienta poderosa para mostrar el estado general y el uso de sus clústeres de Kubernetes y le permite profundizar fácilmente en las áreas de investigación.

Al hacer clic en **Paneles > Explorador de Kubernetes**, se abre la página de lista de clústeres de Kubernetes. Esta página de descripción general contiene una tabla de los clústeres de Kubernetes en su inquilino.

[Página de lista de Kubernetes]

Lista de clústeres

La lista de clústeres muestra la siguiente información para cada clúster de su inquilino:

- Clúster **Nombre**. Al hacer clic en el nombre de un clúster se abrirá el**"página de detalles"** para ese grupo.
- Porcentajes de **saturación**. La saturación general es la más alta entre la saturación de CPU, memoria o almacenamiento.
- Número de **Nodos** en el clúster. Al hacer clic en este número, se abrirá la página de lista de nodos.
- Número de **Pods** en el clúster. Al hacer clic en este número, se abrirá la página de lista de pods.
- Número de **Espacios de nombres** en el clúster. Al hacer clic en este número, se abrirá la página de lista de espacios de nombres.
- Número de **cargas de trabajo** en el clúster. Al hacer clic en este número, se abrirá la página de lista de carga de trabajo.

Refinando el filtro

Cuando esté filtrando, a medida que comience a escribir se le presentará la opción de crear un **filtro comodín** basado en el texto actual. Al seleccionar esta opción se devolverán todos los resultados que coincidan con la expresión comodín. También puede crear **expresiones** usando NOT o AND, o puede seleccionar la opción "Ninguno" para filtrar valores nulos en el campo.

[Filtrado con comodines en K8S Explorer]

Los filtros basados en comodines o expresiones (por ejemplo, NO, Y, "Ninguno", etc.) se muestran en azul oscuro en el campo de filtro. Los elementos que seleccione directamente de la lista se muestran en azul claro.

[Filtro que muestra elementos comodín y seleccionados]

Los filtros de Kubernetes son contextuales, lo que significa, por ejemplo, que si estás en la página de un nodo específico, el filtro pod_name solo enumera los pods relacionados con ese nodo. Además, si aplica un filtro para un espacio de nombres específico, entonces el filtro pod_name listará solo los pods en ese nodo *y* en ese espacio de nombres.

Tenga en cuenta que el filtrado de comodines y expresiones funciona con texto o listas, pero no con números, fechas o valores booleanos.

Antes de instalar o actualizar el operador de monitoreo de Kubernetes de NetApp

Lea esta información antes de instalar o actualizar el"Operador de monitoreo de Kubernetes" .

Componente	Requisito
Versión de Kubernetes	Kubernetes v1.20 y superior.
Distribuciones de Kubernetes	AWS Elastic Kubernetes Service (EKS) Azure Kubernetes Service (AKS) Google Kubernetes Engine (GKE) Red Hat OpenShift Rancher Kubernetes Engine (RKE) VMware Tanzu
Sistema operativo Linux	Data Infrastructure Insights no admite nodos que se ejecuten con arquitectura Arm64. Monitoreo de red: debe ejecutar el kernel de Linux versión 4.18.0 o superior. Photon OS no es compatible.
Etiquetas	Data Infrastructure Insights admite la monitorización de los nodos de Kubernetes que ejecutan Linux, al especificar un selector de nodos de Kubernetes que busca las siguientes etiquetas de Kubernetes en estas plataformas: Kubernetes v1.20 y superior: Kubernetes.io/os = linux Rancher + cattle.io como plataforma de orquestación/Kubernetes: cattle.io/os = linux
Comandos	Los comandos curl y kubectl deben estar disponibles. Para obtener mejores resultados, agregue estos comandos a PATH.
Conectividad	kubectl cli está configurado para comunicarse con el clúster K8s de destino y tener conectividad a Internet con su entorno de Data Infrastructure Insights . Si está detrás de un proxy durante la instalación, siga las instrucciones en el"Configuración del soporte de proxy" Sección de la instalación del Operador. Para obtener informes de datos y auditorías precisos, sincronice la hora en la máquina del Agente mediante el Protocolo de tiempo de red (NTP) o el Protocolo simple de tiempo de red (SNTP).
Otro	Si está ejecutando OpenShift 4.6 o superior, debe seguir las"Instrucciones de OpenShift" Además de garantizar que se cumplan estos requisitos previos.
Token de API	Si está volviendo a implementar el Operador (es decir, lo está actualizando o reemplazando), no es necesario crear un nuevo token de API; puede reutilizar el token anterior.

Cosas importantes a tener en cuenta antes de empezar

Si estás corriendo con unapoderado , tener unarepositorio personalizado , o están usandoOpenShift , lea atentamente las siguientes secciones.

Lea también sobrePermisos .

Configuración del soporte de proxy

Hay dos lugares donde puedes usar un proxy en tu inquilino para instalar el operador de monitoreo de Kubernetes de NetApp . Estos pueden ser los mismos sistemas proxy o sistemas separados:

- Se necesita un proxy durante la ejecución del fragmento de código de instalación (usando "curl") para conectar el sistema donde se ejecuta el fragmento a su entorno de Data Infrastructure Insights
- Proxy necesario para que el clúster de Kubernetes de destino se comunique con su entorno de Data Infrastructure Insights

Si usa un proxy para uno o ambos de estos, para instalar NetApp Kubernetes Operating Monitor primero debe asegurarse de que su proxy esté configurado para permitir una buena comunicación con su entorno de Data Infrastructure Insights . Por ejemplo, desde los servidores/máquinas virtuales desde los que desea instalar el Operador, debe poder acceder a Data Infrastructure Insights y poder descargar binarios desde Data Infrastructure Insights.

Para el proxy utilizado para instalar NetApp Kubernetes Operating Monitor, antes de instalar el operador, configure las variables de entorno *http_proxy/https_proxy*. Para algunos entornos de proxy, es posible que también necesite configurar la variable de entorno *no_proxy*.

Para configurar las variables, realice los siguientes pasos en su sistema **antes** de instalar el operador de monitoreo de Kubernetes de NetApp :

1. Establezca las variables de entorno *https_proxy* y/o *http_proxy* para el usuario actual:
   a. Si el proxy que se está configurando no tiene autenticación (nombre de usuario/contraseña), ejecute el siguiente comando:

```
 export https_proxy=<proxy_server>:<proxy_port>
.. Si el proxy que se está configurando tiene autenticación (nombre
de usuario/contraseña), ejecute este comando:
```

```
export
http_proxy=<proxy_username>:<proxy_password>@<proxy_server>:<proxy_po
rt>
```

Para que el proxy utilizado para su clúster de Kubernetes se comunique con su entorno de Data Infrastructure Insights , instale NetApp Kubernetes Monitoring Operator después de leer todas estas instrucciones.

Configure la sección proxy de AgentConfiguration en operator-config.yaml antes de implementar el operador de monitoreo de Kubernetes de NetApp .

```
agent:
  ...
  proxy:
    server: <server for proxy>
    port: <port for proxy>
    username: <username for proxy>
    password: <password for proxy>

    # In the noproxy section, enter a comma-separated list of
    # IP addresses and/or resolvable hostnames that should bypass
    # the proxy
    noproxy: <comma separated list>

    isTelegrafProxyEnabled: true
    isFluentbitProxyEnabled: <true or false> # true if Events Log enabled
    isCollectorsProxyEnabled: <true or false> # true if Network
Performance and Map enabled
    isAuProxyEnabled: <true or false> # true if AU enabled
  ...
...
```

Uso de un repositorio Docker personalizado o privado

De forma predeterminada, el operador de monitoreo de Kubernetes de NetApp extraerá imágenes de contenedores del repositorio de Data Infrastructure Insights . Si tiene un clúster de Kubernetes utilizado como destino para la supervisión, y ese clúster está configurado para extraer únicamente imágenes de contenedores desde un repositorio de Docker personalizado o privado o un registro de contenedores, debe configurar el acceso a los contenedores que necesita el operador de supervisión de Kubernetes de NetApp .

Ejecute el “Fragmento de extracción de imagen” desde el mosaico de instalación del Operador de monitoreo de NetApp . Este comando iniciará sesión en el repositorio de Data Infrastructure Insights , extraerá todas las dependencias de imágenes para el operador y cerrará sesión en el repositorio de Data Infrastructure Insights . Cuando se le solicite, ingrese la contraseña temporal del repositorio proporcionada. Este comando descarga todas las imágenes utilizadas por el operador, incluidas las funciones opcionales. Vea a continuación para qué funciones se utilizan estas imágenes.

Funcionalidad del operador principal y monitoreo de Kubernetes

- Monitoreo de netapp
- proxy kube-rbac
- métricas de estado de kube
- telégrafo
- usuario root sin distribución

Registro de eventos

- bit fluido

- exportador de eventos de kubernetes

Rendimiento y mapa de la red

- observador de ci-net

Envíe la imagen de Docker del operador a su repositorio de Docker privado/local/empresarial de acuerdo con sus políticas corporativas. Asegúrese de que las etiquetas de imagen y las rutas de directorio de estas imágenes en su repositorio sean coherentes con las del repositorio de Data Infrastructure Insights .

Edite la implementación del operador de monitoreo en operator-deployment.yaml y modifique todas las referencias de imágenes para usar su repositorio privado de Docker.

```
image: <docker repo of the enterprise/corp docker repo>/kube-rbac-
proxy:<kube-rbac-proxy version>
image: <docker repo of the enterprise/corp docker repo>/netapp-
monitoring:<version>
```

Edite AgentConfiguration en operator-config.yaml para reflejar la nueva ubicación del repositorio de Docker. Cree un nuevo imagePullSecret para su repositorio privado. Para obtener más detalles, consulte *https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/*

```
agent:
  ...
  # An optional docker registry where you want docker images to be pulled
from as compared to CI's docker registry
  # Please see documentation for
xref:{relative_path}task_config_telegraf_agent_k8s.html#using-a-custom-or-
private-docker-repository[using a custom or private docker repository].
  dockerRepo: your.docker.repo/long/path/to/test
  # Optional: A docker image pull secret that maybe needed for your
private docker registry
  dockerImagePullSecret: docker-secret-name
```

Instrucciones de OpenShift

Si está ejecutando OpenShift 4.6 o una versión superior, debe editar AgentConfiguration en *operator-config.yaml* para habilitar la configuración *runPrivileged*:

```
# Set runPrivileged to true SELinux is enabled on your kubernetes nodes
runPrivileged: true
```

Openshift puede implementar un nivel adicional de seguridad que puede bloquear el acceso a algunos componentes de Kubernetes.

Permisos

Si el clúster que está supervisando contiene recursos personalizados que no tienen un ClusterRole que"agregados para ver" Será necesario otorgar manualmente al operador acceso a estos recursos para monitorearlos con registros de eventos.

1. Edite *operator-additional-permissions.yaml* antes de instalar, o después de instalar, edite el recurso *ClusterRole/<namespace>-additional-permissions*
2. Cree una nueva regla para los apiGroups y recursos deseados con los verbos ["get", "watch", "list"]. Consulte https://kubernetes.io/docs/reference/access-authn-authz/rbac/
3. Aplicar sus cambios al clúster

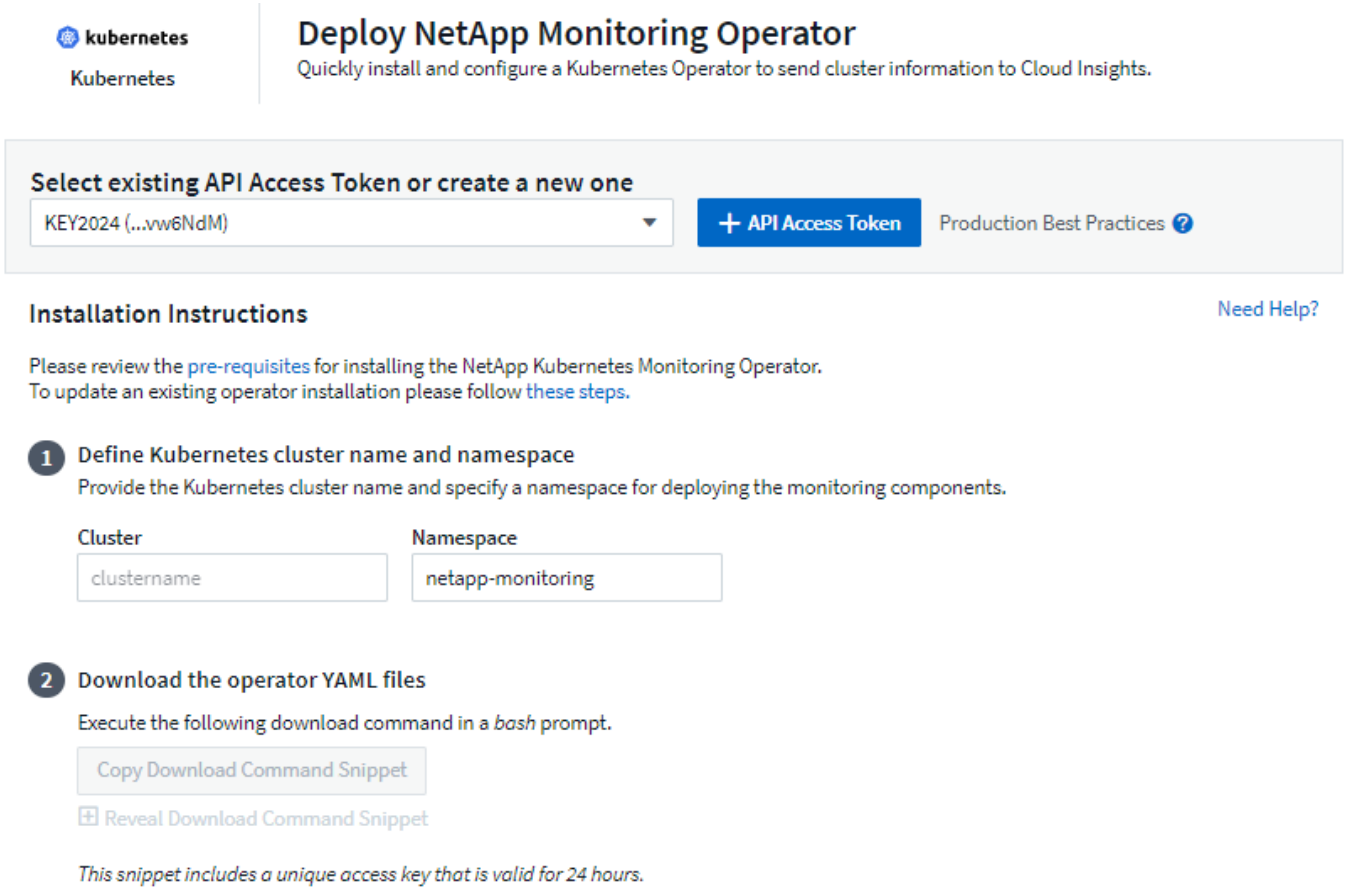
Instalación y configuración del operador de monitoreo de Kubernetes

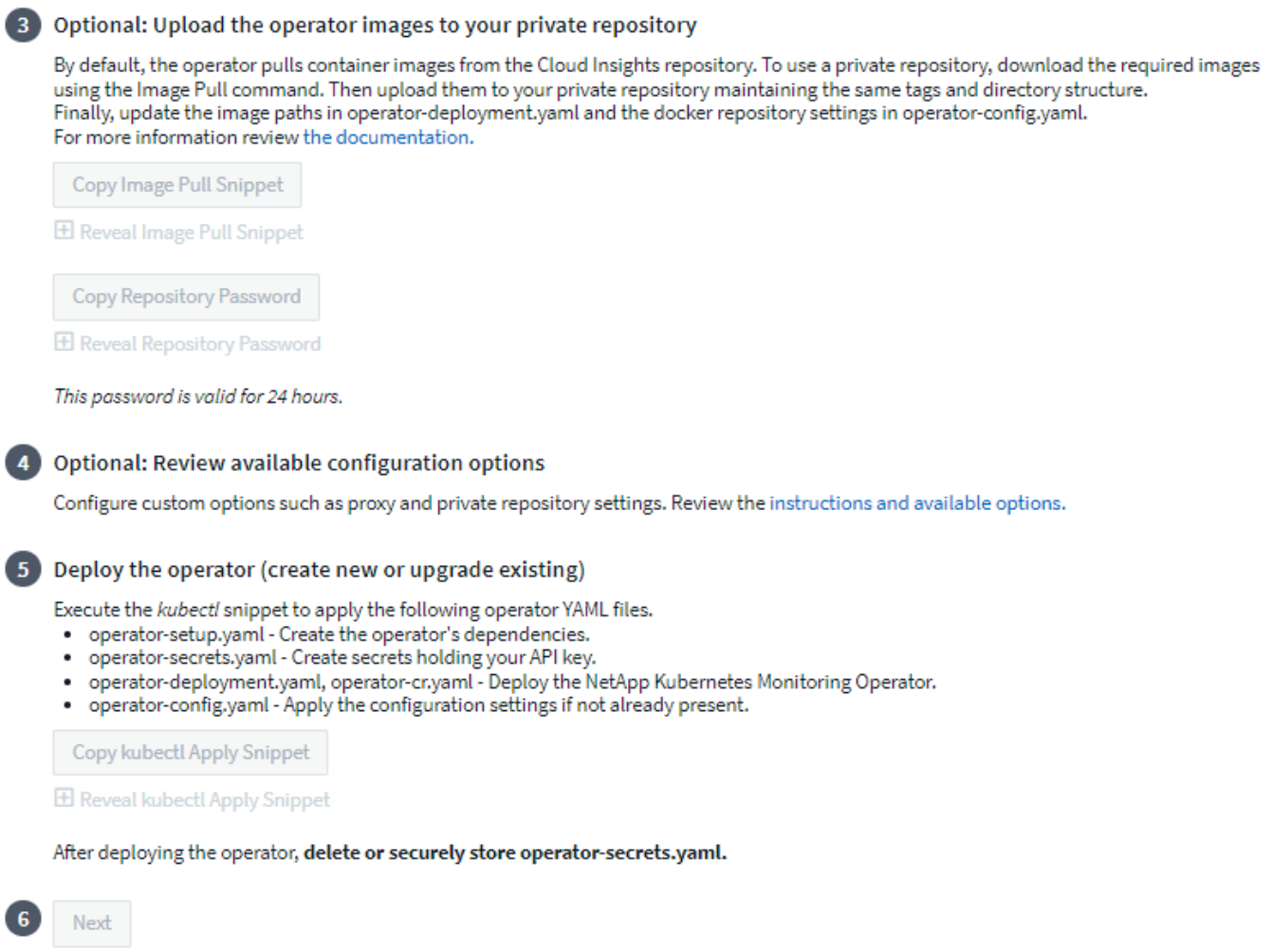
Data Infrastructure Insights ofrece el **Operador de monitoreo de Kubernetes** para la recopilación de Kubernetes. Vaya a **Kubernetes > Recopiladores > +Recopilador de Kubernetes** para implementar un nuevo operador.

Antes de instalar el operador de monitoreo de Kubernetes

Ver el"Prerrequisitos" documentación antes de instalar o actualizar el Operador de Monitoreo de Kubernetes.

Instalación del operador de monitorización de Kubernetes





Pasos para instalar el agente Kubernetes Monitoring Operator en Kubernetes:

1. Introduzca un nombre de clúster y un espacio de nombres únicos. Si tu estasactualización de un operador de Kubernetes anterior, use el mismo nombre de clúster y espacio de nombres.
2. Una vez ingresados estos datos, puedes copiar el fragmento del comando de descarga al portapapeles.
3. Pegue el fragmento en una ventana *bash* y ejecútelo. Se descargarán los archivos de instalación del operador. Tenga en cuenta que el fragmento tiene una clave única y es válido durante 24 horas.
4. Si tiene un repositorio personalizado o privado, copie el fragmento de Image Pull opcional, péguelo en un shell *bash* y ejecútelo. Una vez extraídas las imágenes, cópielas a su repositorio privado. Asegúrese de mantener las mismas etiquetas y estructura de carpetas. Actualice las rutas en *operator-deployment.yaml* así como la configuración del repositorio de Docker en *operator-config.yaml*.
5. Si lo desea, revise las opciones de configuración disponibles, como la configuración de proxy o repositorio privado. Puedes leer más sobre"opciones de configuración" .
6. Cuando esté listo, implemente el operador copiando el fragmento de kubectl Apply, descargándolo y ejecutándolo.
7. La instalación se realiza automáticamente. Cuando haya terminado, haga clic en el botón *Siguiente*.
8. Cuando se complete la instalación, haga clic en el botón *Siguiente*. Asegúrese de eliminar también o almacenar de forma segura el archivo *operator-secrets.yaml*.

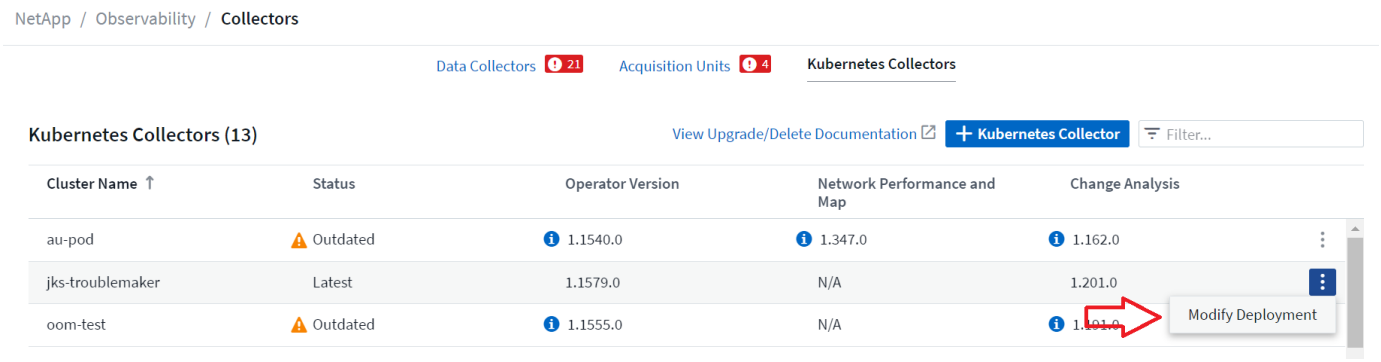
Si tiene un repositorio personalizado, lea sobreUsando un repositorio Docker personalizado/privado .

Componentes de monitorización de Kubernetes

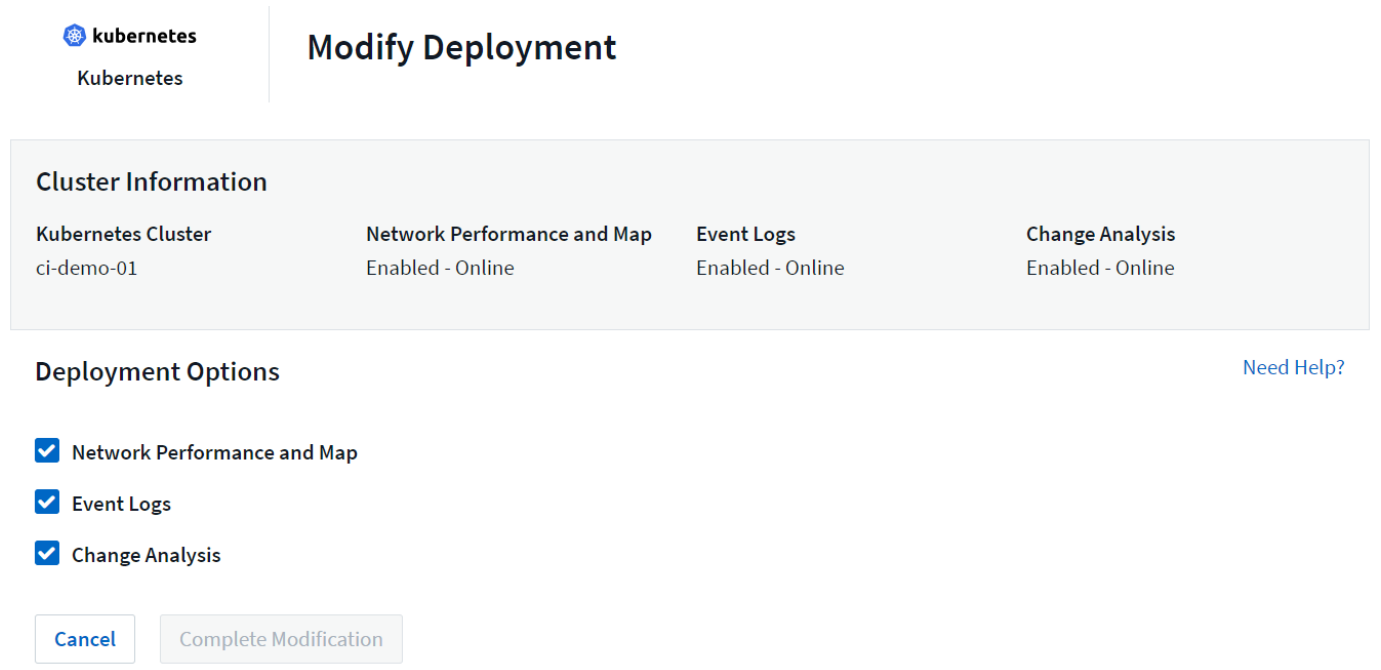
El monitoreo de Kubernetes de Data Infrastructure Insights consta de cuatro componentes de monitoreo:

- Métricas de clúster
- Rendimiento y mapa de la red (opcional)
- Registros de eventos (opcional)
- Análisis de cambios (opcional)

Los componentes opcionales anteriores están habilitados de forma predeterminada para cada recopilador de Kubernetes; si decide que no necesita un componente para un recopilador en particular, puede deshabilitarlo navegando a **Kubernetes > Recopiladores** y seleccionando *Modificar implementación* en el menú de "tres puntos" del recopilador a la derecha de la pantalla.



La pantalla muestra el estado actual de cada componente y le permite deshabilitar o habilitar componentes para ese recopilador según sea necesario.



Actualización al último operador de monitoreo de Kubernetes

Actualizaciones del pulsador DII

Puede actualizar el operador de monitoreo de Kubernetes a través de la página Recopiladores de Kubernetes de DII. Haga clic en el menú junto al clúster que desea actualizar y seleccione *Actualizar*. El operador verificará las firmas de la imagen, realizará una instantánea de su instalación actual y realizará la actualización. En unos minutos deberías ver el progreso del estado del operador desde Actualización en progreso hasta Último. Si encuentra un error, puede seleccionar el estado de Error para obtener más detalles y consultar la tabla de solución de problemas de actualizaciones con botón a continuación.

Actualizaciones con solo pulsar un botón con repositorios privados

Si su operador está configurado para utilizar un repositorio privado, asegúrese de que todas las imágenes necesarias para ejecutar el operador y sus firmas estén disponibles en su repositorio. Si encuentra un error durante el proceso de actualización por imágenes faltantes, simplemente agréguelas a su repositorio y vuelva a intentar la actualización. Para cargar las firmas de imágenes a su repositorio, utilice la herramienta de firma conjunta de la siguiente manera, asegurándose de cargar las firmas para todas las imágenes especificadas en 3 Opcional: Cargue las imágenes del operador a su repositorio privado > Fragmento de extracción de imagen

```
cosign copy example.com/src:v1 example.com/dest:v1
#Example
cosign copy <DII container registry>/netapp-monitoring:<image version>
<private repository>/netapp-monitoring:<image version>
```

Volver a una versión que se estaba ejecutando anteriormente

Si realizó la actualización mediante la función de actualización con solo presionar un botón y encuentra dificultades con la versión actual del operador dentro de los siete días posteriores a la actualización, puede volver a la versión que se estaba ejecutando anteriormente utilizando la instantánea creada durante el proceso de actualización. Haga clic en el menú junto al clúster que desea revertir y seleccione *Revertir*.

Actualizaciones manuales

Determine si existe una AgentConfiguration con el operador existente (si su espacio de nombres no es el predeterminado *netapp-monitoring*, sustitúyalo por el espacio de nombres apropiado):

```
 kubectl -n netapp-monitoring get agentconfiguration netapp-ci-monitoring-
configuration
Si existe una AgentConfiguration:
```

- InstalarEl último operador sobre el operador existente.
  - Asegúrese de que estáextrayendo las últimas imágenes de contenedores si está utilizando un repositorio personalizado.

Si la AgentConfiguration no existe:

- Tome nota del nombre de su clúster tal como lo reconoce Data Infrastructure Insights (si su espacio de nombres no es el predeterminado netapp-monitoring, sustitúyalo por el espacio de nombres apropiado):

```
 kubectl -n netapp-monitoring get agent -o
jsonpath='{.items[0].spec.cluster-name}'
* Cree una copia de seguridad del operador existente (si su espacio de
nombres no es el predeterminado netapp-monitoring, sustitúyalo por el
espacio de nombres apropiado):
```

```
 kubectl -n netapp-monitoring get agent -o yaml > agent_backup.yaml
* <<to-remove-the-kubernetes-monitoring-operator,Desinstalar>>el
operador existente.
* <<installing-the-kubernetes-monitoring-operator,Instalar>>El último
operador.
```

- Utilice el mismo nombre de clúster.
- Después de descargar los últimos archivos YAML de Operator, transfiera cualquier personalización que se encuentre en agent_backup.yaml al operator-config.yaml descargado antes de implementar.
- Asegúrese de que estáextrayendo las últimas imágenes de contenedores si está utilizando un repositorio personalizado.

Detener e iniciar el operador de monitoreo de Kubernetes

Para detener el operador de monitoreo de Kubernetes:

```
 kubectl -n netapp-monitoring scale deploy monitoring-operator
--replicas=0
Para iniciar el operador de monitoreo de Kubernetes:
```

```
kubectl -n netapp-monitoring scale deploy monitoring-operator --replicas=1
```

Desinstalación

Para eliminar el operador de monitoreo de Kubernetes

Tenga en cuenta que el espacio de nombres predeterminado para el operador de monitoreo de Kubernetes es "netapp-monitoring". Si ha configurado su propio espacio de nombres, sustitúyalo en estos y todos los comandos y archivos posteriores.

Las versiones más nuevas del operador de monitoreo se pueden desinstalar con los siguientes comandos:

```
kubectl -n <NAMESPACE> delete agent -l installed-by=nkmo-<NAMESPACE>
kubectl -n <NAMESPACE> delete
clusterrole,clusterrolebinding,crd,svc,deploy,role,rolebinding,secret,sa
-l installed-by=nkmo-<NAMESPACE>
```

Si el operador de monitoreo se implementó en su propio espacio de nombres dedicado, elimine el espacio de nombres:

```
 kubectl delete ns <NAMESPACE>
Nota: Si el primer comando devuelve “No se encontraron recursos”, utilice
las siguientes instrucciones para desinstalar versiones anteriores del
operador de monitoreo.
```

Ejecute cada uno de los siguientes comandos en orden. Dependiendo de su instalación actual, algunos de estos comandos pueden devolver mensajes de "objeto no encontrado". Estos mensajes pueden ignorarse sin problemas.

```
kubectl -n <NAMESPACE> delete agent agent-monitoring-netapp
kubectl delete crd agents.monitoring.netapp.com
kubectl -n <NAMESPACE> delete role agent-leader-election-role
kubectl delete clusterrole agent-manager-role agent-proxy-role agent-
metrics-reader <NAMESPACE>-agent-manager-role <NAMESPACE>-agent-proxy-role
<NAMESPACE>-cluster-role-privileged
kubectl delete clusterrolebinding agent-manager-rolebinding agent-proxy-
rolebinding agent-cluster-admin-rolebinding <NAMESPACE>-agent-manager-
rolebinding <NAMESPACE>-agent-proxy-rolebinding <NAMESPACE>-cluster-role-
binding-privileged
kubectl delete <NAMESPACE>-psp-nkmo
kubectl delete ns <NAMESPACE>
```

Si se creó previamente una restricción de contexto de seguridad:

```
kubectl delete scc telegraf-hostaccess
```

Acerca de Kube-state-metrics

El operador de monitoreo de Kubernetes de NetApp instala sus propias métricas de estado de kube para evitar conflictos con otras instancias.

Para obtener información sobre Kube-State-Metrics, consulte"esta página" .

Configuración/personalización del operador

Estas secciones contienen información sobre cómo personalizar la configuración de su operador, trabajar con proxy, utilizar un repositorio Docker personalizado o privado o trabajar con OpenShift.

Opciones de configuración

Las configuraciones más comúnmente modificadas se pueden configurar en el recurso personalizado *AgentConfiguration*. Puede editar este recurso antes de implementar el operador editando el archivo *operator-config.yaml*. Este archivo incluye ejemplos de configuraciones comentadas. Ver la lista de"configuraciones disponibles" para la versión más reciente del operador.

También puede editar este recurso después de que se haya implementado el operador utilizando el siguiente comando:

```
 kubectl -n netapp-monitoring edit AgentConfiguration
Para determinar si su versión implementada del operador admite
AgentConfiguration, ejecute el siguiente comando:
```

```
 kubectl get crd agentconfigurations.monitoring.netapp.com
Si ve un mensaje de “Error del servidor (No encontrado)”, su operador debe
actualizarse antes de poder usar AgentConfiguration.
```

Configuración del soporte de proxy

Hay dos lugares donde puedes usar un proxy en tu inquilino para instalar el Operador de Monitoreo de Kubernetes. Estos pueden ser los mismos sistemas proxy o sistemas separados:

- Se necesita un proxy durante la ejecución del fragmento de código de instalación (usando "curl") para conectar el sistema donde se ejecuta el fragmento a su entorno de Data Infrastructure Insights
- Proxy necesario para que el clúster de Kubernetes de destino se comunique con su entorno de Data Infrastructure Insights

Si usa un proxy para uno o ambos de estos, para instalar Kubernetes Operating Monitor primero debe asegurarse de que su proxy esté configurado para permitir una buena comunicación con su entorno de Data Infrastructure Insights . Si tiene un proxy y puede acceder a Data Infrastructure Insights desde el servidor/VM desde el que desea instalar el Operador, entonces es probable que su proxy esté configurado correctamente.

Para el proxy utilizado para instalar Kubernetes Operating Monitor, antes de instalar el Operador, configure las variables de entorno *http_proxy/https_proxy*. Para algunos entornos de proxy, es posible que también necesite configurar la variable de entorno *no_proxy*.

Para configurar las variables, realice los siguientes pasos en su sistema **antes** de instalar el operador de monitoreo de Kubernetes:

1. Establezca las variables de entorno *https_proxy* y/o *http_proxy* para el usuario actual:
    a. Si el proxy que se está configurando no tiene autenticación (nombre de usuario/contraseña), ejecute el siguiente comando:

```
 export https_proxy=<proxy_server>:<proxy_port>
.. Si el proxy que se está configurando tiene autenticación (nombre
de usuario/contraseña), ejecute este comando:
```

```
export
http_proxy=<proxy_username>:<proxy_password>@<proxy_server>:<proxy_po
rt>
```

Para que el proxy utilizado para su clúster de Kubernetes se comunique con su entorno de Data Infrastructure Insights , instale el Operador de monitoreo de Kubernetes después de leer todas estas instrucciones.

Configure la sección proxy de AgentConfiguration en operator-config.yaml antes de implementar el operador de monitoreo de Kubernetes.

```
agent:
  ...
  proxy:
    server: <server for proxy>
    port: <port for proxy>
    username: <username for proxy>
    password: <password for proxy>

    # In the noproxy section, enter a comma-separated list of
    # IP addresses and/or resolvable hostnames that should bypass
    # the proxy
    noproxy: <comma separated list>

    isTelegrafProxyEnabled: true
    isFluentbitProxyEnabled: <true or false> # true if Events Log enabled
    isCollectorsProxyEnabled: <true or false> # true if Network
Performance and Map enabled
    isAuProxyEnabled: <true or false> # true if AU enabled
  ...
...
```

Uso de un repositorio Docker personalizado o privado

De forma predeterminada, el operador de monitoreo de Kubernetes extraerá imágenes de contenedores del repositorio de Data Infrastructure Insights . Si tiene un clúster de Kubernetes utilizado como destino para la supervisión, y ese clúster está configurado para extraer únicamente imágenes de contenedores desde un repositorio de Docker personalizado o privado o un registro de contenedores, debe configurar el acceso a los contenedores que necesita el operador de supervisión de Kubernetes.

Ejecute el “Fragmento de extracción de imagen” desde el mosaico de instalación del Operador de monitoreo de NetApp . Este comando iniciará sesión en el repositorio de Data Infrastructure Insights , extraerá todas las

dependencias de imágenes para el operador y cerrará sesión en el repositorio de Data Infrastructure Insights . Cuando se le solicite, ingrese la contraseña temporal del repositorio proporcionada. Este comando descarga todas las imágenes utilizadas por el operador, incluidas las funciones opcionales. Vea a continuación para qué funciones se utilizan estas imágenes.

Funcionalidad del operador principal y monitoreo de Kubernetes

- Monitoreo de netapp
- proxy ci-kube-rbac
- ci-ksm
- ci-telegraf
- usuario root sin distribución

Registro de eventos

- ci-fluent-bit
- exportador de eventos de ci-kubernetes

Rendimiento y mapa de la red

- observador de ci-net

Envíe la imagen de Docker del operador a su repositorio de Docker privado/local/empresarial de acuerdo con sus políticas corporativas. Asegúrese de que las etiquetas de imagen y las rutas de directorio de estas imágenes en su repositorio sean coherentes con las del repositorio de Data Infrastructure Insights .

Edite la implementación del operador de monitoreo en operator-deployment.yaml y modifique todas las referencias de imágenes para usar su repositorio privado de Docker.

```
image: <docker repo of the enterprise/corp docker repo>/ci-kube-rbac-
proxy:<ci-kube-rbac-proxy version>
image: <docker repo of the enterprise/corp docker repo>/netapp-
monitoring:<version>
```

Edite AgentConfiguration en operator-config.yaml para reflejar la nueva ubicación del repositorio de Docker. Cree un nuevo imagePullSecret para su repositorio privado. Para obtener más detalles, consulte *https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/*

```
agent:
  ...
  # An optional docker registry where you want docker images to be pulled
from as compared to CI's docker registry
  # Please see documentation link here:
xref:{relative_path}task_config_telegraf_agent_k8s.html#using-a-custom-or-
private-docker-repository
  dockerRepo: your.docker.repo/long/path/to/test
  # Optional: A docker image pull secret that maybe needed for your
private docker registry
  dockerImagePullSecret: docker-secret-name
```

Instrucciones de OpenShift

Si está ejecutando OpenShift 4.6 o una versión superior, debe editar AgentConfiguration en *operator-config.yaml* para habilitar la configuración *runPrivileged*:

```
# Set runPrivileged to true SELinux is enabled on your kubernetes nodes
runPrivileged: true
```

Openshift puede implementar un nivel adicional de seguridad que puede bloquear el acceso a algunos componentes de Kubernetes.

Tolerancias y Manchas

Los DaemonSets *netapp-ci-telegraf-ds*, *netapp-ci-fluent-bit-ds* y *netapp-ci-net-observer-l4-ds* deben programar un pod en cada nodo de su clúster para poder recopilar datos correctamente en todos los nodos. El operador ha sido configurado para tolerar algunas **manchas** bien conocidas. Si ha configurado alguna tolerancia personalizada en sus nodos, lo que impide que los pods se ejecuten en todos los nodos, puede crear una **tolerancia** para esas tolerancias."en la *Configuración del agente*" . Si ha aplicado tolerancias personalizadas a todos los nodos de su clúster, también debe agregar las tolerancias necesarias a la implementación del operador para permitir que el pod del operador se programe y ejecute.

Más información sobre Kubernetes"Manchas y tolerancias" .

Regresar a la"Página de instalación del operador de monitoreo de Kubernetes de NetApp"

Una nota sobre secretos

Para eliminar el permiso para que el operador de monitoreo de Kubernetes vea secretos en todo el clúster, elimine los siguientes recursos del archivo *operator-setup.yaml* antes de la instalación:

```
ClusterRole/netapp-ci<namespace>-agent-secret
ClusterRoleBinding/netapp-ci<namespace>-agent-secret
```

Si se trata de una actualización, elimine también los recursos de su clúster:

```
 kubectl delete ClusterRole/netapp-ci-<namespace>-agent-secret-clusterrole
 kubectl delete ClusterRoleBinding/netapp-ci-<namespace>-agent-secret-
clusterrolebinding
```

Si el análisis de cambios está habilitado, modifique *AgentConfiguration* o *operator-config.yaml* para descomentar la sección de administración de cambios e incluir *kindsToIgnoreFromWatch: '"secrets"'* en la sección de administración de cambios. Tenga en cuenta la presencia y posición de comillas simples y dobles en esta línea.

```
change-management:
  ...
  # # A comma separated list of kinds to ignore from watching from the
default set of kinds watched by the collector
  # # Each kind will have to be prefixed by its apigroup
  # # Example: '"networking.k8s.io.networkpolicies,batch.jobs",
"authorization.k8s.io.subjectaccessreviews"'
  kindsToIgnoreFromWatch: '"secrets"'
  ...
```

Verificación de las firmas de imágenes del operador de monitoreo de Kubernetes

La imagen del operador y todas las imágenes relacionadas que implementa están firmadas por NetApp. Puede verificar manualmente las imágenes antes de la instalación utilizando la herramienta de firma conjunta o configurar un controlador de admisión de Kubernetes. Para más detalles, consulte la"Documentación de Kubernetes" .

La clave pública utilizada para verificar las firmas de imágenes está disponible en el mosaico de instalación del Operador de Monitoreo en *Opcional: Cargue las imágenes del operador a su repositorio privado > Clave pública de firma de imagen*

Para verificar manualmente una firma de imagen, realice los siguientes pasos:

1. Copiar y ejecutar el fragmento de extracción de imagen
2. Copie e ingrese la contraseña del repositorio cuando se le solicite
3. Almacenar la clave pública de la firma de la imagen (dii-image-signing.pub en el ejemplo)
4. Verificar las imágenes usando cosign. Consulte el siguiente ejemplo de uso de cosignatarios

```
$ cosign verify --key dii-image-signing.pub --insecure-ignore-sct
--insecure-ignore-tlog <repository>/<image>:<tag>
Verification for <repository>/<image>:<tag> --
The following checks were performed on each of these signatures:
  - The cosign claims were validated
  - The signatures were verified against the specified public key
[{"critical":{"identity":{"docker-
reference":"<repository>/<image>"},"image":{"docker-manifest-
digest":"sha256:<hash>"},"type":"cosign container image
signature"},"optional":null}]
```

Solución de problemas

Algunas cosas que puedes probar si tienes problemas al configurar el operador de monitoreo de Kubernetes:

Problema:	Prueba esto:
No veo un hipervínculo/conexión entre mi volumen persistente de Kubernetes y el dispositivo de almacenamiento de back-end correspondiente. Mi volumen persistente de Kubernetes está configurado utilizando el nombre de host del servidor de almacenamiento.	Siga los pasos para desinstalar el agente Telegraf existente y luego vuelva a instalar el agente Telegraf más reciente. Debe utilizar la versión 2.0 o posterior de Telegraf, y el almacenamiento de su clúster de Kubernetes debe estar monitoreado activamente por Data Infrastructure Insights.

Problema:	Prueba esto:
Veo mensajes en los registros similares a los siguientes: E0901 15:21:39.962145 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: No se pudo enumerar *v1.MutatingWebhookConfiguration: el servidor no pudo encontrar el recurso solicitado E0901 15:21:43.168161 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: No se pudo enumerar *v1.Lease: el servidor no pudo encontrar el recurso solicitado (get leases.coordination.k8s.io) etc.	Estos mensajes pueden aparecer si está ejecutando kube-state-metrics versión 2.0.0 o superior con versiones de Kubernetes inferiores a 1.20. Para obtener la versión de Kubernetes: *kubectl version* Para obtener la versión de kube-state-metrics: *kubectl get deploy/kube-state-metrics -o jsonpath='{..image}'* Para evitar que aparezcan estos mensajes, los usuarios pueden modificar su implementación de kube-state-metrics para deshabilitar las siguientes concesiones: *mutatingwebhookconfigurations validatingwebhookconfigurations volumeattachments resources* Más específicamente, pueden usar el siguiente argumento de CLI: resources=certificatesigningrequests,configmaps,cronjobs,daemonsets,ployments,endpoints,horizontalpodautoscalers,ingresses,jobs,limitranges, namespaces,networkpolicies,nodes,persistentvolumeclaims,persistentvolumes, poddisruptionbudgets,pods,replicasets,replicationcontrollers,resourcequotas, Secretos, servicios, conjuntos con estado, clases de almacenamiento. La lista de recursos predeterminada es: "solicitudes de firma de certificados, mapas de configuración, trabajos cron, conjuntos de daemon, implementaciones, puntos finales, escaladores automáticos de pods horizontales, ingresos, trabajos, concesiones, rangos de límites, configuraciones de webhook mutantes, espacios de nombres, políticas de red, nodos, reclamaciones de volumen persistente, volúmenes persistentes, presupuestos de interrupción de pods, pods, conjuntos de réplicas, controladores de replicación, cuotas de recursos, secretos, servicios, conjuntos con estado, clases de almacenamiento, configuraciones de webhook de validación, archivos adjuntos de volumen".

Problema:	Prueba esto:
Veo mensajes de error de Telegraf similares a los siguientes, pero Telegraf se inicia y se ejecuta: 11 de octubre 14:23:41 ip-172-31-39-47 systemd[1]: iniciado El agente de servidor controlado por complemento para informar métricas en InfluxDB. 11 oct 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="No se pudo crear el directorio de caché. /etc/telegraf/.cache/snowflake, err: mkdir /etc/telegraf/.ca che: permiso denegado. ignorado\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" 11 oct 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="error al abrir. Ignorado. abrir /etc/telegraf/.cache/snowflake/ocsp_response_cache.json: no existe el archivo o directorio\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" 11 oct 14:23:41 ip-172-31-39-47 telegraf[1827]: 2021-10-11T14:23:41Z ¡Yo! Iniciando Telegraf 1.19.3	Este es un problema conocido. Referirse a"Este artículo de GitHub" Para más detalles. Mientras Telegraf esté en funcionamiento, los usuarios pueden ignorar estos mensajes de error.
En Kubernetes, mis pods de Telegraf informan el siguiente error: "Error al procesar la información de mountstats: no se pudo abrir el archivo mountstats: /hostfs/proc/1/mountstats, error: abrir /hostfs/proc/1/mountstats: permiso denegado".	Si SELinux está habilitado y en ejecución, es probable que impida que los pods de Telegraf accedan al archivo /proc/1/mountstats en el nodo Kubernetes. Para superar esta restricción, edite la configuración del agente y habilite la configuración runPrivileged. Para obtener más detalles, consulte las instrucciones de OpenShift.
En Kubernetes, mi pod Telegraf ReplicaSet informa el siguiente error: [inputs.prometheus] Error en el complemento: no se pudo cargar el par de claves /etc/kubernetes/pki/etcd/server.crt:/etc/kubernetes/pki/etcd/server.key: abrir /etc/kubernetes/pki/etcd/server.crt: no existe el archivo o directorio	El pod Telegraf ReplicaSet está diseñado para ejecutarse en un nodo designado como maestro o para etcd. Si el pod ReplicaSet no se está ejecutando en uno de estos nodos, obtendrá estos errores. Verifique si sus nodos maestros/etcd tienen manchas. Si es así, agregue las tolerancias necesarias al Telegraf ReplicaSet, telegraf-rs. Por ejemplo, edite ReplicaSet… kubectl edit rs telegraf-rs …y agregue las tolerancias apropiadas a la especificación. Luego, reinicie el pod ReplicaSet.

Problema:	Prueba esto:
Tengo un entorno PSP/PSA. ¿Esto afecta a mi operador de monitoreo?	Si su clúster de Kubernetes se ejecuta con la Política de seguridad de pod (PSP) o la Admisión de seguridad de pod (PSA) implementadas, debe actualizar al Operador de monitoreo de Kubernetes más reciente. Siga estos pasos para actualizar al Operador actual con soporte para PSP/PSA: 1. Desinstalar el operador de monitorización anterior: kubectl delete agent agent-monitoring-netapp -n netapp-monitoring kubectl delete ns netapp-monitoring kubectl delete crd agents.monitoring.netapp.com kubectl delete clusterrole agent-manager-role agent-proxy-role agent-metrics-reader kubectl delete clusterrolebinding agent-manager-rolebinding agent-proxy-rolebinding agent-cluster-admin-rolebinding 2. Instalar la última versión del operador de monitoreo.
Tuve problemas al intentar implementar el Operador y tengo PSP/PSA en uso.	1. Edite el agente utilizando el siguiente comando: kubectl -n <name-space> edit agent 2. Marcar 'security-policy-enabled' como 'falso'. Esto deshabilitará las Políticas de seguridad de pod y la Admisión de seguridad de pod y permitirá que el Operador realice la implementación. Confirme usando los siguientes comandos: kubectl get psp (debería mostrar que se eliminó la política de seguridad del pod) kubectl get all -n <namespace>
grep -i psp (debería mostrar que no se encontró nada)	Errores "ImagePullBackoff" detectados
Estos errores pueden aparecer si tiene un repositorio Docker personalizado o privado y aún no ha configurado el operador de monitoreo de Kubernetes para reconocerlo correctamente. Leer más Acerca de la configuración para un repositorio personalizado/privado.	Tengo un problema con la implementación de mi operador de monitoreo y la documentación actual no me ayuda a resolverlo.

Capture o anote de otro modo el resultado de los siguientes comandos y comuníquese con el equipo de soporte técnico.
`kubectl -n netapp-monitoring get all`
`kubectl -n netapp-monitoring describe all`
`kubectl -n netapp-monitoring logs <monitoring-operator-pod> --all-containers=true`
`kubectl -n netapp-monitoring logs <telegraf-pod> --all-containers=true`
	Los pods de net-observer (Mapa de carga de trabajo) en el espacio de nombres del operador están en CrashLoopBackOff

Problema:	Prueba esto:
Estos pods corresponden al recopilador de datos del mapa de carga de trabajo para la observabilidad de la red. Pruebe lo siguiente: • Verifique los registros de uno de los pods para confirmar la versión mínima del kernel. Por ejemplo: ---- {"ci-tenant-id":"your-tenant-id","collector-cluster":"your-k8s-cluster-name","environment":"prod","level":"error","msg":"erro r en la validación. Motivo: la versión del kernel 3.10.0 es inferior a la versión mínima del kernel 4.18.0","time":"2022-11-09T08:23:08Z"} ---- • Los pods de Net-observer requieren que la versión del kernel de Linux sea al menos 4.18.0. Verifique la versión del kernel usando el comando “uname -r” y asegúrese de que sea >= 4.18.0	Los pods se ejecutan en el espacio de nombres del operador (predeterminado: netapp-monitoring), pero no se muestran datos en la interfaz de usuario para el mapa de carga de trabajo ni las métricas de Kubernetes en las consultas.
Verifique la configuración de la hora en los nodos del clúster K8S. Para obtener informes de datos y auditorías precisos, se recomienda encarecidamente sincronizar la hora en la máquina del Agente mediante el Protocolo de tiempo de red (NTP) o el Protocolo simple de tiempo de red (SNTP).	Algunos de los pods de net-observer en el espacio de nombres del operador están en estado pendiente
Net-observer es un DaemonSet y ejecuta un pod en cada nodo del clúster k8s. • Observe el pod que está en estado pendiente y verifique si está experimentando un problema de recursos de CPU o memoria. Asegúrese de que la memoria y la CPU necesarias estén disponibles en el nodo.	Veo lo siguiente en mis registros inmediatamente después de instalar el operador de monitoreo de Kubernetes: [inputs.prometheus] Error en el complemento: error al realizar la solicitud HTTP a http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: Obtener http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: marcar tcp: buscar kube-state-metrics.<namespace>.svc.cluster.local: no existe dicho host
Este mensaje normalmente solo se ve cuando se instala un nuevo operador y el pod *telegraf-rs* está activo antes que el pod *ksm*. Estos mensajes deberían detenerse una vez que todos los pods estén ejecutándose.	No veo ninguna métrica recopilada para los CronJobs de Kubernetes que existen en mi clúster.
Verifique su versión de Kubernetes (es decir, `kubectl version`). Si es v1.20.x o anterior, esta es una limitación esperada. La versión kube-state-metrics implementada con el operador de monitoreo de Kubernetes solo admite v1.CronJob. Con Kubernetes 1.20.x y anteriores, el recurso CronJob está en v1beta.CronJob. Como resultado, kube-state-metrics no puede encontrar el recurso CronJob.	Después de instalar el operador, los pods telegraf-ds ingresan a CrashLoopBackOff y los registros de los pods indican "su: Error de autenticación".

Problema:	Prueba esto:
Edite la sección telegraf en *AgentConfiguration* y establezca *dockerMetricCollectionEnabled* en falso. Para obtener más detalles, consulte el manual del operador."opciones de configuración" . … especificación: … telégrafo: … - nombre: docker modo de ejecución: - sustituciones de DaemonSet: - clave: DOCKER_UNIX_SOCK_PLACEHOLDER valor: unix:///run/docker.sock … …	Veo mensajes de error repetidos similares al siguiente en mis registros de Telegraf: E! [agente] Error al escribir en outputs.http: Publicación "https://<tenant_url>/rest/v1/lake/ingest/influxdb": se excedió el plazo de contexto (se excedió el tiempo de espera del cliente mientras se esperaban los encabezados)
Edite la sección telegraf en *AgentConfiguration* y aumente *outputTimeout* a 10 s. Para obtener más detalles, consulte el manual del operador."opciones de configuración" .	Me faltan datos de *involvedobject* para algunos registros de eventos.
Asegúrese de haber seguido los pasos de la"Permisos" Sección anterior.	¿Por qué veo dos pods de operador de monitoreo en ejecución, uno llamado netapp-ci-monitoring-operator-<pod> y el otro llamado monitoring-operator-<pod>?
A partir del 12 de octubre de 2023, Data Infrastructure Insights ha refactorizado el operador para brindar un mejor servicio a nuestros usuarios; para que esos cambios se adopten por completo, debeeliminar el antiguo operador yinstalar el nuevo .	Mis eventos de Kubernetes dejaron de informarse inesperadamente a Data Infrastructure Insights.
Recuperar el nombre del pod del exportador de eventos: `kubectl -n netapp-monitoring get pods`	grep event-exporter

Problema:	Prueba esto:
awk '{print $1}'	sed 's/event-exporter./event-exporter/'` Debe ser "netapp-ci-event-exporter" o "event-exporter". A continuación, edite el agente de monitorización. `kubectl -n netapp-monitoring edit agent` , y configure el valor de LOG_FILE para reflejar el nombre del pod del exportador de eventos apropiado que se encontró en el paso anterior. Más específicamente, LOG_FILE debe establecerse en "/var/log/containers/netapp-ci-event-exporter.log" o "/var/log/containers/event-exporter*.log". …. fluent-bit: … - name: event-exporter-ci substitutions: - key: LOG_FILE values: - /var/log/containers/netapp-ci-event-exporter*.log … …. Alternativamente, también se puededesinstalar yreinstalar el agente.
Veo que los pods implementados por el operador de monitoreo de Kubernetes fallan debido a recursos insuficientes.	Consulte el operador de monitoreo de Kubernetes"opciones de configuración" para aumentar los límites de CPU y/o memoria según sea necesario.
Una imagen faltante o una configuración no válida provocó que los pods netapp-ci-kube-state-metrics no pudieran iniciarse o no estuvieran listos. Ahora el StatefulSet está bloqueado y los cambios de configuración no se aplican a los pods netapp-ci-kube-state-metrics.	El StatefulSet está en un"roto" estado. Después de solucionar cualquier problema de configuración, rebote los pods netapp-ci-kube-state-metrics.
Los pods netapp-ci-kube-state-metrics no se inician después de ejecutar una actualización del operador de Kubernetes y arrojan ErrImagePull (error al extraer la imagen).	Intente restablecer los pods manualmente.
Se están observando mensajes del tipo "Evento descartado por ser más antiguo que maxEventAgeSeconds" para mi clúster de Kubernetes en Análisis de registros.	Modifique el operador *agentconfiguration* y aumente *event-exporter-maxEventAgeSeconds* (es decir, a 60 s), *event-exporter-kubeQPS* (es decir, a 100) y *event-exporter-kubeBurst* (es decir, a 500). Para obtener más detalles sobre estas opciones de configuración, consulte la"opciones de configuración" página.

Problema:	Prueba esto:
Telegraf advierte o se bloquea debido a que no hay suficiente memoria bloqueable.	Intente aumentar el límite de memoria bloqueable para Telegraf en el nodo/sistema operativo subyacente. Si aumentar el límite no es una opción, modifique la configuración del agente NKMO y establezca *unprotected* en *true*. Esto le indicará a Telegraf que no intente reservar páginas de memoria bloqueadas. Si bien esto puede representar un riesgo de seguridad, ya que los secretos descifrados pueden intercambiarse en el disco, permite la ejecución en entornos donde no es posible reservar memoria bloqueada. Para obtener más detalles sobre las opciones de configuración *unprotected*, consulte la"opciones de configuración" página.
Veo mensajes de advertencia de Telegraf similares al siguiente: *W! [inputs.diskio] No se puede obtener el nombre del disco para "vdc": error al leer /dev/vdc: no existe el archivo o directorio*	Para el operador de monitoreo de Kubernetes, estos mensajes de advertencia son benignos y se pueden ignorar de forma segura. Alternativamente, edite la sección telegraf en AgentConfiguration y establezca *runDsPrivileged* en verdadero. Para más detalles, consulte la"opciones de configuración del operador" .

Problema:	Prueba esto:

Mi pod de fluent-bit está fallando con los siguientes errores: [2024/10/16 14:16:23] [error] [/src/fluent-bit/plugins/in_tail/tail_fs_inotify.c:360 errno=24] Demasiados archivos abiertos [2024/10/16 14:16:23] [error] No se pudo inicializar la entrada tail.0 [2024/10/16 14:16:23] [error] [engine] Falló la inicialización de la entrada	Intente cambiar la configuración de *fsnotify* en su clúster:
	
	`sudo sysctl fs.inotify.max_user_instances (take note of setting)`
	
	`sudo sysctl fs.inotify.max_user_instances=<something larger than current setting>`
	
	`sudo sysctl fs.inotify.max_user_watches (take note of setting)`
	
	`sudo sysctl fs.inotify.max_user_watches=<something larger than current setting>`
	
	Reinicie Fluent-bit.
	
	Nota: para que estas configuraciones sean persistentes después de reiniciar el nodo, debe colocar las siguientes líneas en */etc/sysctl.conf*
	
	`fs.inotify.max_user_instances=<something larger than current setting>`
	`fs.inotify.max_user_watches=<something larger than current setting>`

Problema:	Prueba esto:
Los pods DS de Telegraf informan errores relacionados con el complemento de entrada de Kubernetes que no puede realizar solicitudes HTTP debido a la imposibilidad de validar el certificado TLS. Por ejemplo: E! [inputs.kubernetes] Error en el complemento: error al realizar la solicitud HTTP a"<a href="https://<kubelet_IP>:10250/stats/summary": " class="bare">https://<kubelet_IP>:10250/stats/summary": Conseguir"<a href="https://<kubelet_IP>:10250/stats/summary": " class="bare">https://<kubelet_IP>:10250/stats/summary": tls: no se pudo verificar el certificado: x509: no se puede validar el certificado para <kubelet_IP> porque no contiene ninguna SAN IP	Esto ocurrirá si el kubelet usa certificados autofirmados y/o el certificado especificado no incluye <kubelet_IP> en la lista *Nombre alternativo del sujeto* del certificado. Para solucionar esto, el usuario puede modificar el"configuración del agente" , y establezca *telegraf:insecureK8sSkipVerify* en *true*. Esto configurará el complemento de entrada de telegraf para omitir la verificación. Alternativamente, el usuario puede configurar el kubelet para"servidorTLSBootstrap" , lo que activará una solicitud de certificado desde la API 'certificates.k8s.io'.

Información adicional se puede encontrar en el"Soporte" página o en el"Matriz de soporte del recopilador de datos" .

Opciones de configuración del operador de monitoreo de Kubernetes

El "Operador de monitoreo de Kubernetes" Ofrece amplias opciones de personalización mediante el archivo AgentConfiguration. Puede configurar límites de recursos, intervalos de recopilación, configuración de proxy, tolerancias y ajustes específicos de cada componente para optimizar la monitorización de su entorno de Kubernetes. Utilice estas opciones para personalizar Telegraf, Kube-State-Metrics, la recopilación de registros, el mapeo de cargas de trabajo, la gestión de cambios y otros componentes de monitorización.

Archivo de configuración del agente de muestra

A continuación se muestra un archivo *AgentConfiguration* de muestra, con descripciones para cada opción.

```
apiVersion: monitoring.netapp.com/v1alpha1
kind: AgentConfiguration
metadata:
  name: netapp-ci-monitoring-configuration
  namespace: "netapp-monitoring"
  labels:
    installed-by: nkmo-netapp-monitoring

spec:
  ##
  ## One can modify the following settings to configure and customize the
operator.
```

```
  ## Optional settings are commented out with their default values for
reference.
  ## To update them, uncomment the line, change the value, and apply the
updated AgentConfiguration.
  ##
  agent:
    ##
    ## [REQUIRED FIELD]
    ## A uniquely identifiable user-friendly cluster name
    ## The cluster name must be unique across all clusters in your Data
Infrastructure Insights (DII) environment.
    ##
    clusterName: "my_cluster"

    ##
    ## Proxy settings
    ## If applicable, specify the proxy through which the operator should
communicate with DII.
    ## Refer to additional documentation here:
    ## https://docs.netapp.com/us-
en/cloudinsights/task_config_telegraf_agent_k8s.html#configuring-proxy-
support
    ##
    # proxy:
    #   server:
    #   port:
    #   noproxy:
    #   username:
    #   password:
    #   isTelegrafProxyEnabled:
    #   isFluentbitProxyEnabled:
    #   isCollectorsProxyEnabled:

    ##
    ## [REQUIRED FIELD]
    ## Repository from which the operator pulls the required images
    ## By default, the operator pulls from the DII repository. To use a
private repository, set this field to the
    ## applicable repository name. Refer to additional documentation here:
    ## https://docs.netapp.com/us-
en/cloudinsights/task_config_telegraf_agent_k8s.html#using-a-custom-or-
private-docker-repository
    ##
    dockerRepo: 'docker.c01.cloudinsights.netapp.com'
    ##
    ## [REQUIRED FIELD]
```

```
    ## Name of the imagePullSecret required for dockerRepo
    ## When using a private repository, set this field to the applicable
secret name.
    ##
    dockerImagePullSecret: 'netapp-ci-docker'

    ##
    ## Automatic expiring API key rotation settings
    ## Allow the operator to automatically rotate its expiring API key,
generating a new API key and
    ## using it to replace the expiring one. The expiring API key itself
must support auto rotation.
    ##
    # tokenRotationEnabled: 'true'
    ##
    ## Threshold (number of days before expiration) at which the operator
should trigger rotation.
    ## The threshold must be less than the total duration of the API key.
    ##
    # tokenRotationThresholdDays: '30'

  push-button-upgrades:
    ##
    ## Allow the operator to be upgraded using the Data Infrastructure
Insights (DII) UI
    ##
    # enabled: 'true'

    ##
    ## Frequency at which the operator polls and checks for upgrade
requests from DII
    ##
    # polltimeSeconds: '60'

    ##
    ## Allow operator upgrade to proceed even if new images are not
present
    ##
    # ignoreImageNotPresent: 'false'

    ##
    ## Allow operator upgrade to proceed even if image signature
verification fails
    ## Warning: Enabling this setting is dangerous!
    ##
    # ignoreImageSignatureFailure: 'false'
```

```
    ##
    ## Allow operator upgrade to proceed even if image signature
verification fails
    ## Warning: Enabling this setting is dangerous!
    ##
    # ignoreYAMLSignatureFailure: 'false'

    ##
    ## Use dockerImagePullSecret to access the image repository and verify
the existence of the new images
    ##
    # imageValidationUseSecret: 'true'

    ##
    ## Time allowed for the old operator pod to shutdown before reporting
an upgrade failure to DII
    ##
    # upgradesShutdownTime: '240'

    ##
    ## Time allowed for the new operator pod to startup before reporting
an upgrade failure to DII
    ##
    # upgradesStartupTime: '600'

  telegraf:
    ##
    ## Frequency at which telegraf collects data
    ## The frequency should not exceed 60s.
    ##
    # collectionInterval: '60s'

    ##
    ## Maximum number of metrics per batch
    ## Telegraf sends metrics to outputs in batches. This controls the
size of those writes.
    ##
    # batchSize: '10000'

    ##
    ## Maximum number of unwritten metrics per output
    ## Telegraf caches metrics until they are successfully written by the
output. This controls how many metrics
    ## can be cached. Once the buffer is filled, the oldest metrics will
get dropped.
    ##
```

```
    # bufferLimit: '150000'

    ##
    ## Rounds collection interval to collectionInterval
    ## If collectionInterval is 60s, collection will occur on-the-minute
    ##
    # roundInterval: 'true'

    ##
    ## Jitter between plugins on collection
    ## Each input plugin sleeps a random amount of time within jitter
before collecting. This can be used to prevent
    ## multiple input plugins from querying the same resources at the same
time. The maximum collection interval would
    ## be collectionInterval + collectionJitter.
    ##
    # collectionJitter: '0s'

    ##
    ## Precision to which collected metrics are rounded
    ## When set to "0s", precision will be set by the units specified by
collectionInterval.
    ##
    # precision: '0s'

    ##
    ## Frequency at which telegraf flushes and writes data
    ## Frequency should not exceed collectionInterval.
    ##
    # flushInterval: '60s'

    ##
    ## Jitter between plugins on writes
    ## Each output plugin sleeps a random amount of time within jitter
before flushing. This can be used to prevent
    ## multiple output plugins from writing the same resources at the same
time, and causing large spikes. The maximum
    ## flush interval would be flushInterval + flushJitter.
    ##
    # flushJitter: '0s'

    ##
    ## Timeout for HTTP output plugins
    ## Time allowed for http output plugins to successfully writing before
failing.
    ##
```

```
    # outputTimeout: '5s'

    ##
    ## CPU/Mem limits and requests for netapp-ci-telegraf-ds DaemonSet
    ##
    # dsCpuLimit: '750m'
    # dsMemLimit: '800Mi'
    # dsCpuRequest: '100m'
    # dsMemRequest: '500Mi'

    ##
    ## CPU/Mem limits and requests for netapp-ci-telegraf-rs ReplicaSet
    ##
    # rsCpuLimit: '3'
    # rsMemLimit: '4Gi'
    # rsCpuRequest: '100m'
    # rsMemRequest: '500Mi'

    ##
    ## telegraf runs through the processor plugins a second time after the
aggregators plugins, by default. Use this
    ## option to skip the second run.
    ##
    # skipProcessorsAfterAggregators: 'false'

    ##
    ## Additional tolerations for netapp-ci-telegraf-ds DaemonSet and
netapp-ci-telegraf-rs ReplicaSet
    ## Inspect the netapp-ci-telegraf-rs ReplicaSet and netapp-ci-
telegraf-ds DaemonSet to view the default tolerations.
    ## If additional tolerations are needed, specify them here using the
following abbreviated single line format:
    ##
    ## Example: '{key: taint1, operator: Exists, effect: NoSchedule},{key:
taint2, operator: Exists, effect: NoExecute}'
    ##
    # dsTolerations: ''
    # rsTolerations: ''

    ##
    ## Additional node selector terms for netapp-ci-telegraf-rs ReplicaSet
    ## Inspect the netapp-ci-telegraf-rs ReplicaSet to view the default
node selectors terms. If additional node
    ## selector terms are needed, specify them here using the following
abbreviated single line format:
    ##
```

```
    ## Example: '{"key": "myLabel1","operator": "In","values":
["myVal1"]},{"key": "myLabel2","operator": "In","values": ["myVal2"]}'
    ##
    ## These additional node selector terms will be AND'd with the default
ones via matchExpressions.
    ##
    # rsNodeSelectorTerms: ''

    ##
    ## telegraf uses lockable memory to protect secrets in memory. If
telegraf issues warnings about insufficient
    ## lockable memory, try increasing the limit of lockable memory on the
applicable nodes. If increasing this limit
    ## is not an option for the given environment, set unprotected to true
so telegraf does not attempt to use
    ## lockable memory.
    ##
    # unprotected: 'false'

    ##
    ## Run the netapp-ci-telegraf-ds DaemonSet's telegraf-mountstats-
poller container in privileged mode
    ## The telegraf-mountstats-poller container needs read-only access to
system files such as those in /proc/ (i.e. to
    ## monitor NFS IO metrics, etc.). Some environments impose restricts
that prevent the container from reading these
    ## system files. Unless those restrictions are lifted, users may need
to run this container in privileged mode.
    ##
    # runPrivileged: 'false'

    ##
    ## Run the netapp-ci-telegraf-ds DaemonSet's telegraf container in
privileged mode
    ## The telegraf container needs read-only access to system files such
as those in /dev/ (i.e. for the telegraf
    ## diskio input plugin to retrieve disk metrics). Some environments
impose restricts that prevent the container from
    ## accessing these system files. Unless those restrictions are lifted,
users may need to run this container in
    ## privileged mode.
    ##
    # runDsPrivileged: 'false'

    ##
    ## Allow the netapp-ci-telegraf-ds DaemonSet's telegraf-ds, telegraf-
```

```
init, and telegraf-mountstats-poller containers
    ## to run with escalation privilege. This is needed to access/read
root-protected files (node UUID,
    ## /proc/1/mountstats, etc.). Allowing escalation privilege should
negate the need to run these containers in
    ## privileged mode.
    ##
    # allowDsPrivilegeEscalation: 'true'

    ##
    ## Allow the netapp-ci-telegraf-rs DaemonSet's telegraf-rs and
telegraf-rs-init containers
    ## to run with escalation privilege. This is needed to access/read
root-protected files (node UUID,
    ## etcd credentials when applicable, etc.). Allowing escalation
privilege should negate the need to run these
    ## containers in privileged mode.
    ##
    # allowRsPrivilegeEscalation: 'true'

    ##
    ## Enable collection of block IO metrics (kubernetes.pod_to_storage)
    ##
    # dsBlockIOEnabled: 'true'

    ##
    ## Enable collection of NFS IO metrics (kubernetes.pod_to_storage)
    ##
    # dsNfsIOEnabled: 'true'

    ##
    ## Enable collection of system-specific objects/metrics for managed
k8s clusters
    ## This consists of k8s objects within the kube-system and cattle-
system namespaces for managed k8s clusters
    ## (i.e. EKS, AKS, GKE, managed Rancher, etc.).
    ##
    # managedK8sSystemMetricCollectionEnabled: 'false'

    ##
    ## Enable collection of pod ephemeral storage metrics
(kubernetes.pod_volume)
    ##
    # podVolumeMetricCollectionEnabled: 'false'

    ##
```

```
    ## Declare Rancher cluster is managed
    ## Rancher can be deployed in managed or on-premise environments. The
operator contains logic to try to determine
    ## which type of environment Rancher is running in (i.e. to factor
into managedK8sSystemMetricCollectionEnabled).
    ## If the operator logic misidentifies whether Rancher is running in a
managed environment or not, use this option
    ## to declare Rancher is managed.
    ##
    # isManagedRancher: 'false'

    ##
    ## Locations for the etcd certificate and key files
    ## The operator looks at well-known locations for the etcd certificate
and key files. If this cannot find these
    ## files, the applicable telegraf input plugin will fail. Use this
option to specify the complete filepath to these
    ## files on the nodes.
    ## Note that the well-known locations for these files are typically
root-protected. This is one of the reasons why
    ## the netapp-ci-telegraf-rs ReplicaSet's telegraf-rs-init container
needs to run with escalation privileges.
    ##
    # rsHostEtcdCrt: ''
    # rsHostEtcdKey: ''

    ##
    ## Allow operator/telegraf communications with k8s without TLS
verification
    ## In some environments, TLS verification will not succeed (i.e.
certificates lack IP SANs). To skip the
    ## verification, use this option.
    ##
    # insecureK8sSkipVerify: 'false'

  kube-state-metrics:
    ##
    ## CPU/Mem limits and requests for netapp-ci-kube-state-metrics
StatefulSet
    ##
    # cpuLimit: '500m'
    # memLimit: '1Gi'
    # cpuRequest: '100m'
    # memRequest: '500Mi'

    ##
```

```
    ## Comma-separated list of k8s resources for which to collect metrics
    ## Refer to the kube-state-metrics --resources CLI option
    ##
    # resources:
'cronjobs,daemonsets,deployments,horizontalpodautoscalers,ingresses,jobs,n
amespaces,nodes,persistentvolumeclaims,persistentvolumes,pods,replicasets,
resourcequotas,services,statefulsets'

    ##
    ## Comma-separated list of k8s metrics to collect
    ## Refer to the kube-state-metrics --metric-allowlist CLI option
    ##
    # metrics:
'kube_cronjob_created,kube_cronjob_status_active,kube_cronjob_labels,kube_
daemonset_created,kube_daemonset_status_current_number_scheduled,kube_daem
onset_status_desired_number_scheduled,kube_daemonset_status_number_availab
le,kube_daemonset_status_number_misscheduled,kube_daemonset_status_number_
ready,kube_daemonset_status_number_unavailable,kube_daemonset_status_obser
ved_generation,kube_daemonset_status_updated_number_scheduled,kube_daemons
et_metadata_generation,kube_daemonset_labels,kube_deployment_status_replic
as,kube_deployment_status_replicas_available,kube_deployment_status_replic
as_unavailable,kube_deployment_status_replicas_updated,kube_deployment_sta
tus_observed_generation,kube_deployment_spec_replicas,kube_deployment_spec
_paused,kube_deployment_spec_strategy_rollingupdate_max_unavailable,kube_d
eployment_spec_strategy_rollingupdate_max_surge,kube_deployment_metadata_g
eneration,kube_deployment_labels,kube_deployment_created,kube_job_created,
kube_job_owner,kube_job_status_active,kube_job_status_succeeded,kube_job_s
tatus_failed,kube_job_labels,kube_job_status_start_time,kube_job_status_co
mpletion_time,kube_namespace_created,kube_namespace_labels,kube_namespace_
status_phase,kube_node_info,kube_node_labels,kube_node_role,kube_node_spec
_unschedulable,kube_node_created,kube_persistentvolume_capacity_bytes,kube
_persistentvolume_status_phase,kube_persistentvolume_labels,kube_persisten
tvolume_info,kube_persistentvolume_claim_ref,kube_persistentvolumeclaim_ac
cess_mode,kube_persistentvolumeclaim_info,kube_persistentvolumeclaim_label
s,kube_persistentvolumeclaim_resource_requests_storage_bytes,kube_persiste
ntvolumeclaim_status_phase,kube_pod_info,kube_pod_start_time,kube_pod_comp
letion_time,kube_pod_owner,kube_pod_labels,kube_pod_status_phase,kube_pod_
status_ready,kube_pod_status_scheduled,kube_pod_container_info,kube_pod_co
ntainer_status_waiting,kube_pod_container_status_waiting_reason,kube_pod_c
ontainer_status_running,kube_pod_container_state_started,kube_pod_containe
r_status_terminated,kube_pod_container_status_terminated_reason,kube_pod_c
ontainer_status_last_terminated_reason,kube_pod_container_status_ready,kub
e_pod_container_status_restarts_total,kube_pod_overhead_cpu_cores,kube_pod
_overhead_memory_bytes,kube_pod_created,kube_pod_deletion_timestamp,kube_p
od_init_container_info,kube_pod_init_container_status_waiting,kube_pod_ini
t_container_status_waiting_reason,kube_pod_init_container_status_running,k
```

```
ube_pod_init_container_status_terminated,kube_pod_init_container_status_te
rminated_reason,kube_pod_init_container_status_last_terminated_reason,kube
_pod_init_container_status_ready,kube_pod_init_container_status_restarts_t
otal,kube_pod_status_scheduled_time,kube_pod_status_unschedulable,kube_pod
_spec_volumes_persistentvolumeclaims_readonly,kube_pod_container_resource_
requests_cpu_cores,kube_pod_container_resource_requests_memory_bytes,kube_
pod_container_resource_requests_storage_bytes,kube_pod_container_resource_
requests_ephemeral_storage_bytes,kube_pod_container_resource_limits_cpu_co
res,kube_pod_container_resource_limits_memory_bytes,kube_pod_container_res
ource_limits_storage_bytes,kube_pod_container_resource_limits_ephemeral_st
orage_bytes,kube_pod_init_container_resource_limits_cpu_cores,kube_pod_ini
t_container_resource_limits_memory_bytes,kube_pod_init_container_resource_
limits_storage_bytes,kube_pod_init_container_resource_limits_ephemeral_sto
rage_bytes,kube_pod_init_container_resource_requests_cpu_cores,kube_pod_in
it_container_resource_requests_memory_bytes,kube_pod_init_container_resour
ce_requests_storage_bytes,kube_pod_init_container_resource_requests_epheme
ral_storage_bytes,kube_replicaset_status_replicas,kube_replicaset_status_r
eady_replicas,kube_replicaset_status_observed_generation,kube_replicaset_s
pec_replicas,kube_replicaset_metadata_generation,kube_replicaset_labels,ku
be_replicaset_created,kube_replicaset_owner,kube_resourcequota,kube_resour
cequota_created,kube_service_info,kube_service_labels,kube_service_created
,kube_service_spec_type,kube_statefulset_status_replicas,kube_statefulset_
status_replicas_current,kube_statefulset_status_replicas_ready,kube_statef
ulset_status_replicas_updated,kube_statefulset_status_observed_generation,
kube_statefulset_replicas,kube_statefulset_metadata_generation,kube_statef
ulset_created,kube_statefulset_labels,kube_statefulset_status_current_revi
sion,kube_statefulset_status_update_revision,kube_node_status_capacity,kub
e_node_status_allocatable,kube_node_status_condition,kube_pod_container_re
source_requests,kube_pod_container_resource_limits,kube_pod_init_container
_resource_limits,kube_pod_init_container_resource_requests,kube_horizontal
podautoscaler_spec_max_replicas,kube_horizontalpodautoscaler_spec_min_repl
icas,kube_horizontalpodautoscaler_status_condition,kube_horizontalpodautos
caler_status_current_replicas,kube_horizontalpodautoscaler_status_desired_
replicas'

    ##
    ## Comma-separated list of k8s label keys that will be used to
determine which labels to export/collect
    ## Refer to the kube-state-metrics --metric-labels-allowlist CLI
option
    ##
    # labels:
'cronjobs=[*],daemonsets=[*],deployments=[*],horizontalpodautoscalers=[*],
ingresses=[*],jobs=[*],namespaces=[*],nodes=[*],persistentvolumeclaims=[*]
,persistentvolumes=[*],pods=[*],replicasets=[*],resourcequotas=[*],service
s=[*],statefulsets=[*]'
```

```
    ##
    ## Additional tolerations for netapp-ci-kube-state-metrics StatefulSet
    ## Inspect the netapp-ci-kube-state-metrics StatefulSet to view the
default tolerations. If additional
    ## tolerations are needed, specify them here using the following
abbreviated single line format:
    ##
    ## Example: '{key: taint1, operator: Exists, effect: NoSchedule},{key:
taint2, operator: Exists, effect: NoExecute}'
    ##
    # tolerations: ''

    ##
    ## Additional node selector terms for netapp-ci-kube-state-metrics
StatefulSet
    ## Inspect the kube-state-metrics StatefulSet to view the default node
selectors terms. If additional node selector
    ## terms are needed, specify them here using the following abbreviated
single line format:
    ##
    ## Example: '{"key": "myLabel1","operator": "In","values":
["myVal1"]},{"key": "myLabel2","operator": "In","values": ["myVal2"]}'
    ##
    ## These additional node selector terms will be AND'd with the default
ones via matchExpressions.
    ##
    # nodeSelectorTerms: ''

    ##
    ## Number of kube-state-metrics shards
    ## For large clusters, kube-state-metrics may be overwhelmed with
collecting and exporting the amount of metrics
    ## generated. This can lead to collection timeouts for the netapp-ci-
telegraf-rs pod. If this is observed, use this
    ## option to increase the number of kube-state-metrics shards to
redistribute the workload.
    ##
    # shards: '2'

  logs:
    ##
    ## Allow the netapp-ci-fluent-bit-ds DaemonSet's fluent-bit container
to run with escalation privilege.
    ## This is needed to access/read root-protected files (event-exporter
pod log, fluent-bit DB file, etc.).
```

```
    ##
    # fluent-bit-allowPrivilegeEscalation: 'true'

    ##
    ## Read content from the head of the file, not the tail
    ##
    # readFromHead: "true"

    ##
    ## Network protocol for DNS (i.e. UDP, TCP, etc.)
    ##
    # dnsMode: "UDP"

    ##
    ## DNS resolver (i.e. LEGACY, ASYNC, etc.)
    ##
    # fluentBitDNSResolver: "LEGACY"

    ##
    ## Additional tolerations for netapp-ci-fluent-bit-ds DaemonSet
    ## Inspect the netapp-ci-fluent-bit-ds DaemonSet to view the default
tolerations. If additional tolerations are
    ## needed, specify them here using the following abbreviated single
line format:
    ##
    ## Example: '{key: taint1, operator: Exists, effect: NoSchedule},{key:
taint2, operator: Exists, effect: NoExecute}'
    ##
    # fluent-bit-tolerations: ''

    ##
    ## CPU/Mem limits and requests for netapp-ci-fluent-bit-ds DaemonSet
    ##
    # fluent-bit-cpuLimit: '500m'
    # fluent-bit-memLimit: '1Gi'
    # fluent-bit-cpuRequest: '50m'
    # fluent-bit-memRequest: '100Mi'

    ##
    ## Top-level host path in which the kubernetes container logs reside,
including any symlinks from var/log/containers
    ## For example, if /var/log/containers/*.log is a symlink to
/kubernetes/log to
    ## /kubernetes/var/lib/docker/containers/*/*.log, fluent-bit-
containerLogPath should be set to '/kubernetes'.
    ##
```

```
    # fluent-bit-containerLogPath: '/var/lib/docker/containers'

    ## fluent-bit DB file path/location

    ##
    ## fluent-bit DB file path/location
    ## By default, fluent-bit is configured to use /var/log/netapp-
monitoring_flb_kube.db.  This path usually requires
    ## escalated privileges for read/write. Users who want to avoid
escalation privilege can use this option to specify
    ## a different DB file path/location. The custom path/location should
allow non-root users to read/write.
    ## Ideally, the path/location should be persistent.
    ##
    # fluent-bit-dbFile: '/var/log/netapp-monitoring_flb_kube.db'

    ##
    ## Additional tolerations for netapp-ci-event-exporter Deployment
    ## Inspect the netapp-ci-event-exporter Deployment to view the default
tolerations. If additional tolerations are
    ## needed, specify them here using the following abbreviated single
line format:
    ##
    ## Example: '{key: taint1, operator: Exists, effect: NoSchedule},{key:
taint2, operator: Exists, effect: NoExecute}'
    ##
    # event-exporter-tolerations: ''

    ##
    ## CPU/Mem limits and requests for netapp-ci-event-exporter Deployment
    ##
    # event-exporter-cpuLimit: '500m'
    # event-exporter-memLimit: '1Gi'
    # event-exporter-cpuRequest: '50m'
    # event-exporter-memRequest: '100Mi'

    ##
    ## Max age for events to be processed and exported; older events are
discarded
    ##
    # event-exporter-maxEventAgeSeconds: '10'

    ##
    ## Client-side throttling
    ## Set event-exporter-kubeBurst to roughly match event rate
    ## Set event-exporter-kubeQPS to approximately 1/5 of event-exporter-
```

```
kubeBurst
    ##
    # event-exporter-kubeQPS: 20
    # event-exporter-kubeBurst: 100

    ##
    ## Additional node selector terms for netapp-ci-event-exporter
Deployment
    ## Inspect the event-exporter Deployment to view the default node
selectors terms. If additional node selector terms
    ## are needed, specify them here using the following abbreviated
single line format:
    ##
    ## Example: '{"key": "myLabel1","operator": "In","values":
["myVal1"]},{"key": "myLabel2","operator": "In","values": ["myVal2"]}'
    ##
    ## These additional node selector terms will be AND'd with the default
ones via matchExpressions.
    ##
    # event-exporter-nodeSelectorTerms: ''

  workload-map:
    ## Run workload-map container with escalation privilege to coordinate
memlocks
    ##
    ## Allow the netapp-ci-net-observer-l4-ds DaemonSet's net-observer
container to run with escalation privilege.
    ## This is needed to coordinate memlocks.
    ##
    # allowPrivilegeEscalation: 'true'

    ##
    ## CPU/Mem limits and requests for netapp-ci-net-observer-l4-ds
DaemonSet
    ##
    # cpuLimit: '500m'
    # memLimit: '500Mi'
    # cpuRequest: '100m'
    # memRequest: '500Mi'

    ##
    ## Metric aggregation interval (in seconds)
    ## Set metricAggregationInterval between 30 and 120
    ##
    # metricAggregationInterval: '60'
```

```
    ##
    ## Interval for bpf polling
    ## Set bpfPollInterval between 3 and 15
    ##
    # bpfPollInterval: '8'

    ##
    ## Enable reverse DNS lookups on observed IPs
    ##
    # enableDNSLookup: 'true'

    ##
    ## Additional tolerations for netapp-ci-net-observer-l4-ds DaemonSet
    ## Inspect the netapp-ci-net-observer-l4-ds DaemonSet to view the
default tolerations. If additional tolerations
    ## are needed, specify them here using the following abbreviated
single line format:
    ##
    ## Example: '{key: taint1, operator: Exists, effect: NoSchedule},{key:
taint2, operator: Exists, effect: NoExecute}'
    ##
    # l4-tolerations: ''

    ##
    ## Run the netapp-ci-net-observer-l4-ds DaemonSet's net-observer
container in privileged mode
    ## Some environments impose restricts that prevent the net-observer
container from running.
    ## Unless those restrictions are lifted, users may need to run this
container in privileged mode.
    ##
    # runPrivileged: 'false'

  change-management:
    ##
    ## CPU/Mem limits and requests for netapp-ci-change-observer-watch-rs
ReplicaSet
    ##
    # cpuLimit: '1'
    # memLimit: '1Gi'
    # cpuRequest: '500m'
    # memRequest: '500Mi'

    ##
    ## Interval (in seconds) after which a non-successful deployment of a
workload will be marked as failed
```

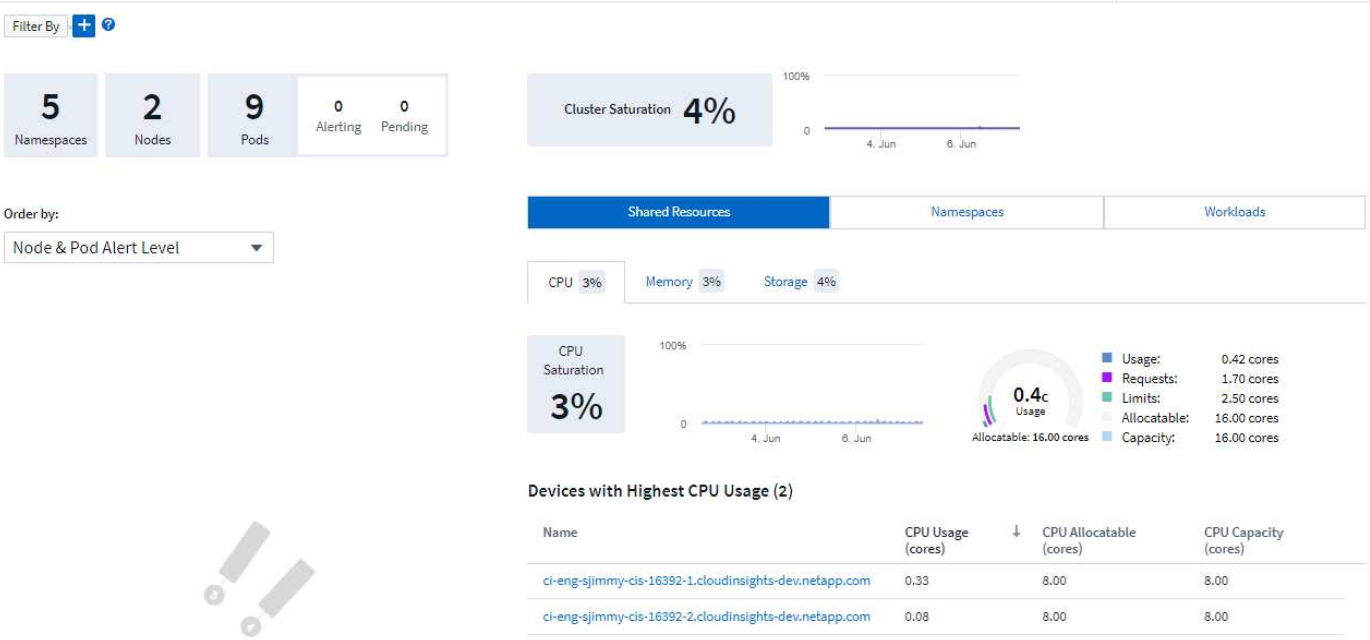
```
    ##
    # workloadFailureDeclarationIntervalSeconds: '30'

    ##
    ## Frequency (in seconds) at which workload deployments are combined
and sent
    ##
    # workloadDeployAggrIntervalSeconds: '300'

    ##
    ## Frequency (in seconds) at which non-workload deployments are
combined and sent
    ##
    # nonWorkloadDeployAggrIntervalSeconds: '15'

    ##
    ## Set of regular expressions used in env names and data maps whose
value will be redacted
    ##
    # termsToRedact: '"pwd", "password", "token", "apikey", "api-key",
"api_key", "jwt", "accesskey", "access_key", "access-key", "ca-file",
"key-file", "cert", "cafile", "keyfile", "tls", "crt", "salt",
".dockerconfigjson", "auth", "secret"'

    ##
    ## Additional node selector terms for netapp-ci-change-observer-watch-
rs ReplicaSet
    ## Inspect the netapp-ci-change-observer-watch-rs ReplicaSet to view
the default node selectors terms. If additional
    ## node selector terms are needed, specify them here using the
following abbreviated single line format:
    ##
    ## Example: '{"key": "myLabel1","operator": "In","values":
["myVal1"]},{"key": "myLabel2","operator": "In","values": ["myVal2"]}'
    ##
    ## These additional node selector terms will be AND'd with the default
ones via matchExpressions.
    ##
    # nodeSelectorTerms: ''

    ##
    ## Comma-separated list of additional kinds to watch
    ## Each kind should be prefixed by its API group. This list in
addition to the default set of kinds watched by the
    ## collector.
    ##
```

```
    ## Example: '"authorization.k8s.io.subjectaccessreviews"'
    ##
    # additionalKindsToWatch: ''

    ##
    ## Comma-separated list of additional field paths whose diff is
ignored as part of change analytics
    ## This list in addition to the default set of field paths ignored by
the collector.
    ##
    ## Example: '"metadata.specTime", "data.status"'
    ##
    # additionalFieldsDiffToIgnore: ''

    ##
    ## Comma-separated list of kinds to ignore from watching from the
default set of kinds watched by the collector
    ## Each kind should be prefixed by its API group.
    ##
    ## Example: '"networking.k8s.io.networkpolicies,batch.jobs",
"authorization.k8s.io.subjectaccessreviews"'
    ##
    # kindsToIgnoreFromWatch: ''

    ##
    ## Frequency with which log records are sent to DII from the collector
    ##
    # logRecordAggrIntervalSeconds: '20'

    ##
    ## Additional tolerations for netapp-ci-change-observer-watch-rs
ReplicaSet
    ## Inspect the netapp-ci-change-observer-watch-rs ReplicaSet to view
the default tolerations. If additional
    ## tolerations are needed, specify them here using the following
abbreviated single line format:
    ##
    ## Example: '{key: taint1, operator: Exists, effect: NoSchedule},{key:
taint2, operator: Exists, effect: NoExecute}'
    ##
    # watch-tolerations: ''
```

Página de detalles del clúster de Kubernetes

La página de detalles del clúster de Kubernetes muestra una descripción detallada de su

clúster de Kubernetes.



Número de espacios de nombres, nodos y pods

Los recuentos en la parte superior de la página muestran la cantidad total de espacios de nombres, nodos y pods en el clúster, así como la cantidad de popds que actualmente están en alerta y pendientes.

Recursos compartidos y saturación

En la parte superior derecha de la página de detalles se encuentra la saturación del clúster como porcentaje actual, así como un gráfico que muestra la tendencia reciente a lo largo del tiempo. La saturación del clúster es el nivel más alto de saturación de CPU, memoria o almacenamiento en cualquier momento.

Debajo de eso, la página muestra de forma predeterminada el uso de **Recursos compartidos**, con pestañas para CPU, Memoria y Almacenamiento. Cada pestaña muestra el porcentaje de saturación y la tendencia a lo largo del tiempo, con detalles de uso adicionales. Para el almacenamiento, el valor mostrado es el mayor entre la saturación del backend y del sistema de archivos, que se calculan de forma independiente.

Los dispositivos con mayor uso se muestran en una tabla en la parte inferior. Haga clic en cualquier enlace para explorar estos dispositivos.

Espacios de nombres

La pestaña Espacios de nombres muestra una lista de todos los espacios de nombres en su entorno de Kubernetes, mostrando el uso de CPU y memoria, así como un recuento de cargas de trabajo en cada espacio de nombres. Haga clic en los enlaces de Nombre para explorar cada espacio de nombres.

Shared Resources | Namespaces | Workloads

Namespaces (5)

Name ↓	CPU Usage (cores)	Memory Usage (GiB)	Workload Count
netapp-monitoring	0.25	0.38	4
kube-system	0.01	0.03	3
kube-public	0.00	0.00	0
kube-node-lease	0.00	0.00	0
default	0.00	<0.01	1

Cargas de trabajo

De manera similar, la pestaña Cargas de trabajo muestra una lista de las cargas de trabajo en cada espacio de nombres, mostrando nuevamente el uso de CPU y memoria. Al hacer clic en los vínculos del espacio de nombres se accede a cada uno de ellos.

Shared Resources | Namespaces | Workloads

Workloads (8)

Name ↓	CPU Usage (cores)	Memory Usage (GiB)	Namespace
telegraf-rs-lf9gg	0.24	0.24	netapp-monitoring
telegraf-ds-k957c	0.01	0.10	netapp-monitoring
nginx	0.00	<0.01	default
monitoring-operator-6fcf4755ff-p2cs6	<0.01	0.02	netapp-monitoring
metrics-server-7b4f8b595-f7j9f	<0.01	0.01	kube-system
local-path-provisioner-64d457c485-289gx	<0.01	0.01	kube-system
kube-state-metrics-7995866f8c-t8c49	<0.01	0.01	netapp-monitoring
coredns-5d69dc75db-nkw5p	<0.01	0.01	kube-system

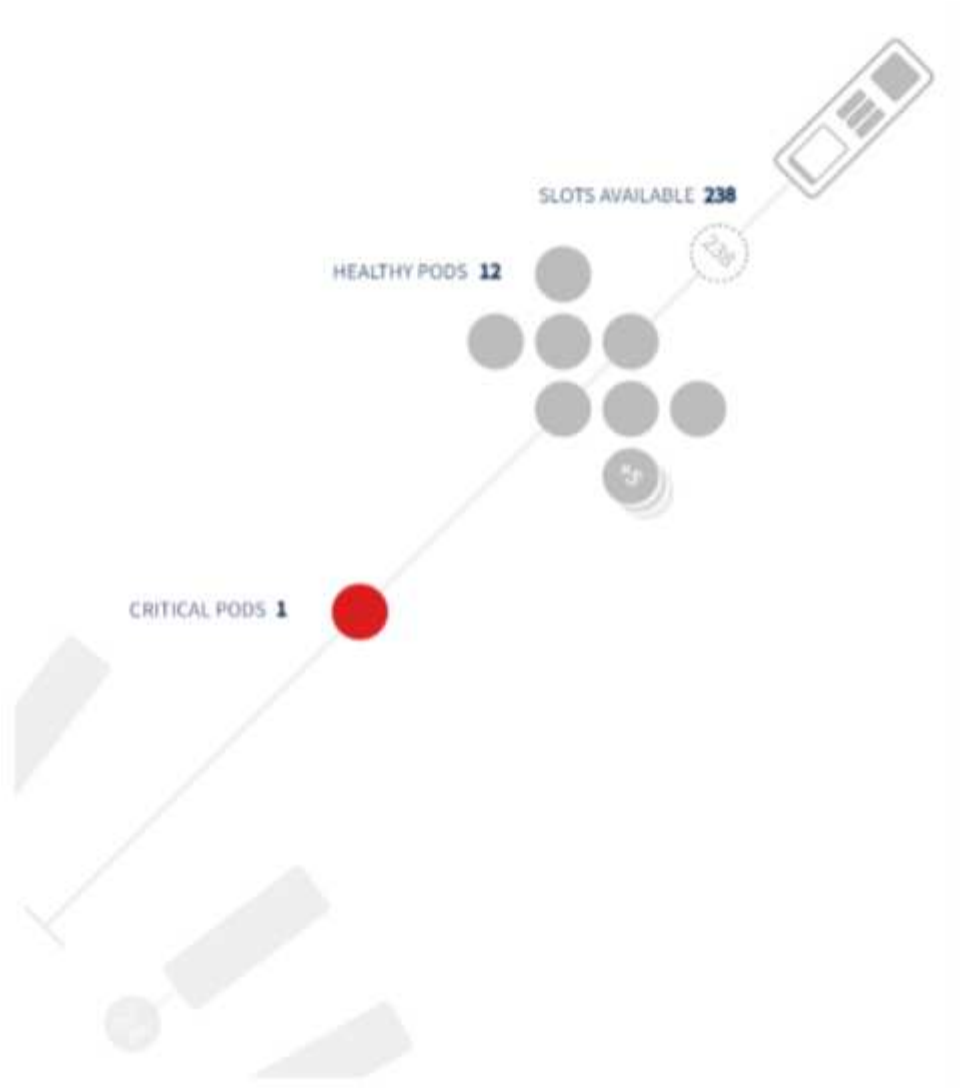
La "rueda" del clúster

La sección "Rueda" del clúster proporciona un vistazo del estado del nodo y del pod, que puede explorar en profundidad para obtener más información. Si su clúster contiene más nodos de los que se pueden mostrar en esta área de la página, podrá girar la rueda utilizando los botones disponibles.

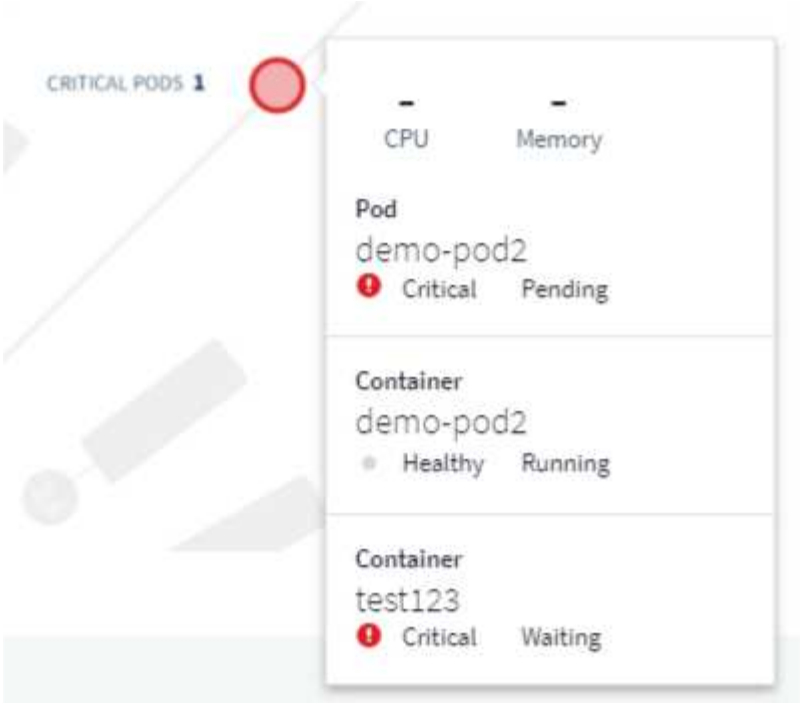
Los pods o nodos de alerta se muestran en rojo. Las áreas de "Advertencia" se muestran en naranja. Los pods que no estén programados (es decir, no estén adjuntos) se mostrarán en la esquina inferior de la "Rueda" del clúster.

Al pasar el cursor sobre un pod (círculo) o un nodo (barra), se ampliará la vista del nodo.

Al hacer clic en el pod o nodo en esa vista, se ampliará la vista del nodo expandida.



Desde aquí, puedes pasar el cursor sobre un elemento para mostrar detalles sobre ese elemento. Por ejemplo, al pasar el cursor sobre el pod crítico en este ejemplo, se muestran detalles sobre ese pod.



Puede ver información del sistema de archivos, la memoria y la CPU colocando el cursor sobre los elementos del nodo.

Una nota sobre los indicadores

Los indicadores de memoria y CPU muestran tres colores, ya que muestran lo *usado* en relación con la *capacidad asignable* y la *capacidad total*.

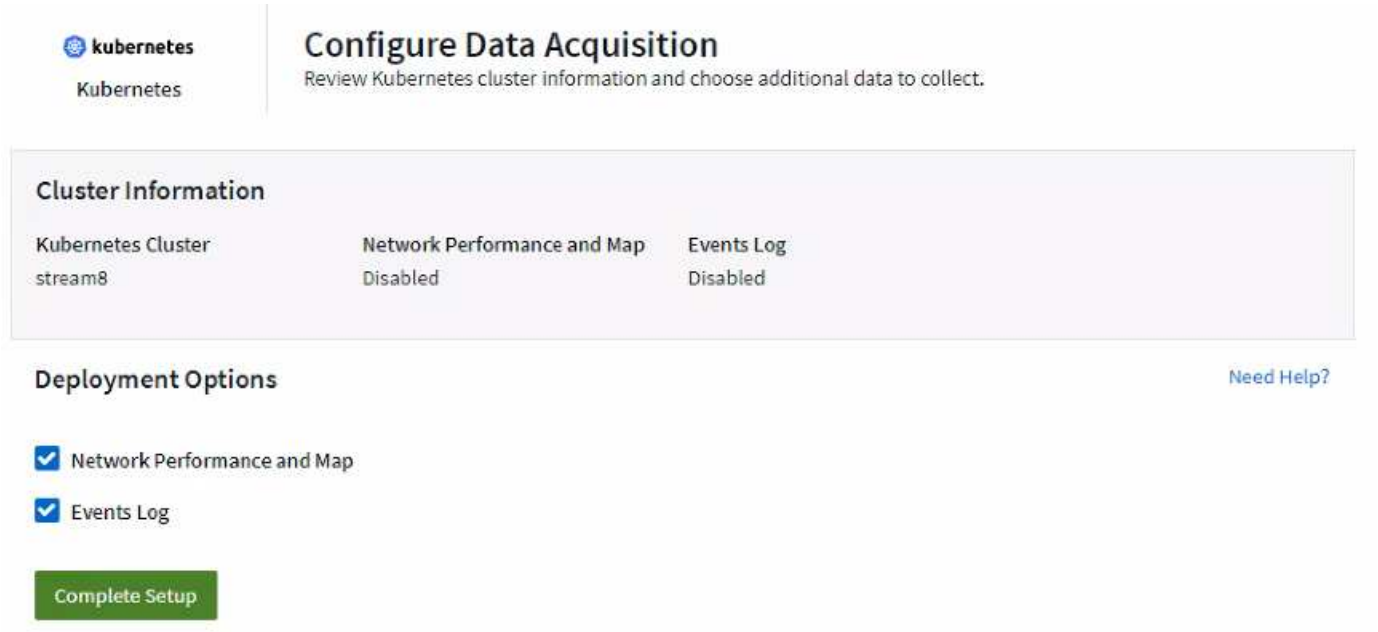
Monitoreo y mapa del rendimiento de la red de Kubernetes

La función de monitoreo y mapeo del rendimiento de la red de Kubernetes simplifica la resolución de problemas al mapear las dependencias entre servicios (también llamados cargas de trabajo) y brinda visibilidad en tiempo real de las latencias y anomalías del rendimiento de la red para identificar problemas de rendimiento antes de que afecten a los usuarios. Esta capacidad ayuda a las organizaciones a reducir los costos generales al analizar y auditar los flujos de tráfico de Kubernetes.

Características principales: • El mapa de carga de trabajo presenta las dependencias y los flujos de carga de trabajo de Kubernetes y resalta los problemas de red y rendimiento. • Supervisar el tráfico de red entre pods, cargas de trabajo y nodos de Kubernetes; identifica el origen del tráfico y los problemas de latencia. • Reducir los costos generales mediante el análisis del tráfico de red de ingreso, egreso, entre regiones y entre zonas.

Prerrequisitos

Antes de poder usar el Mapa y el Monitoreo del Rendimiento de la Red de Kubernetes, debe haber configurado el"Operador de monitorización de Kubernetes de NetApp" para habilitar esta opción. Durante la implementación del Operador, seleccione la casilla de verificación "Rendimiento de red y mapa" para habilitarla. También puede habilitar esta opción navegando a una página de destino de Kubernetes y seleccionando "Modificar implementación".



Monitores

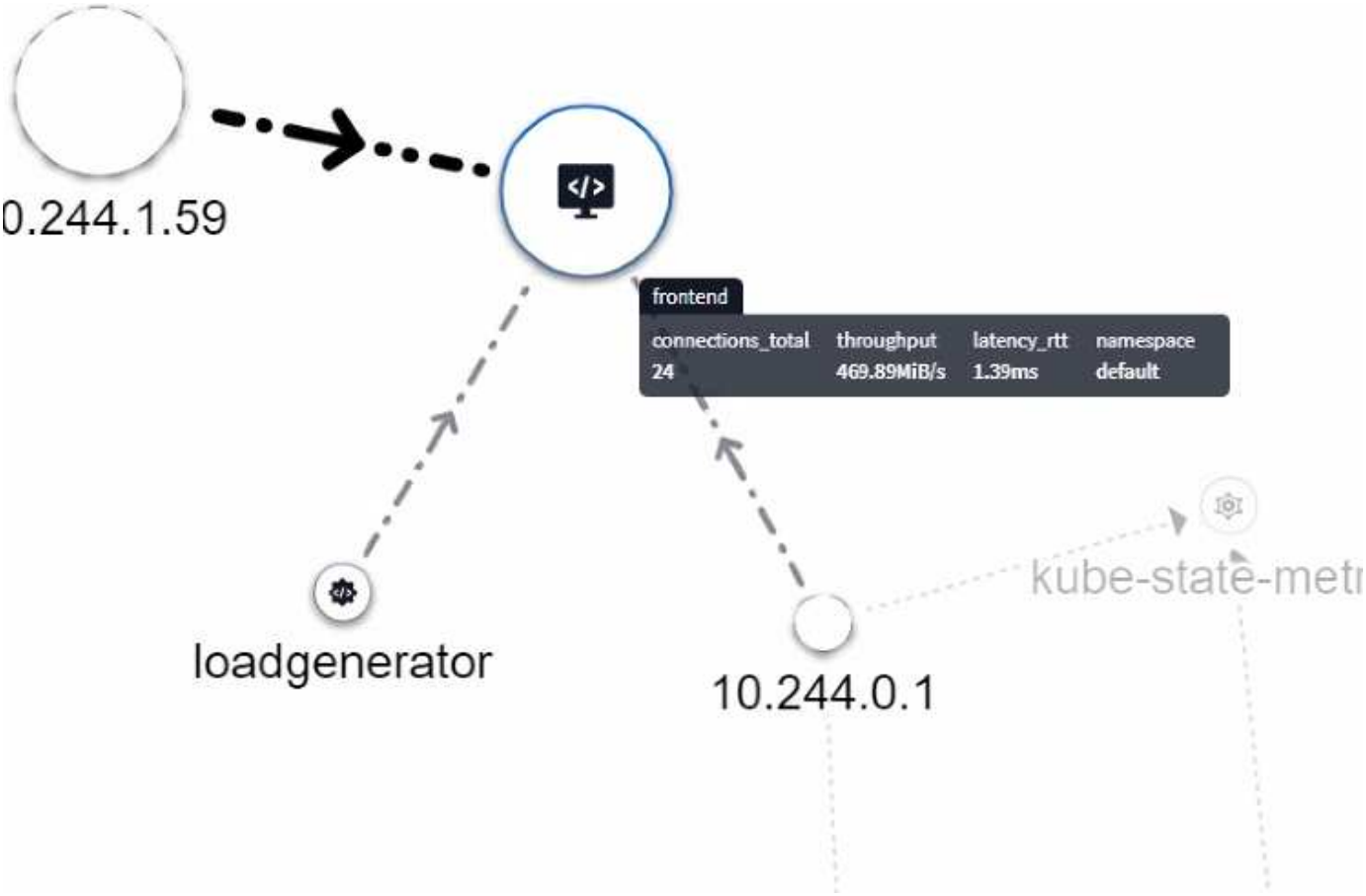
El mapa de carga de trabajo utiliza"monitores" para derivar información. Data Infrastructure Insights proporciona una serie de monitores Kubernetes predeterminados (tenga en cuenta que estos pueden estar *en pausa* de manera predeterminada). Puede *Reanudar* (es decir, habilitar) los monitores que desee) o puede crear monitores personalizados para objetos de Kubernetes, que el Mapa de carga de trabajo también utilizará.

Puede crear alertas de métricas de Data Infrastructure Insights en cualquiera de los tipos de objetos a continuación. Asegúrese de que los datos estén agrupados por el tipo de objeto predeterminado.

- kubernetes.workload
- kubernetes.daemonset
- implementación de kubernetes
- kubernetes.cronjob
- kubernetes.job
- conjunto de réplicas de kubernetes
- conjunto con estado de kubernetes
- kubernetes.pod
- tráfico de red kubernetes_l4

El mapa

El mapa muestra servicios/cargas de trabajo y sus relaciones entre sí. Las flechas muestran las direcciones del tráfico. Al pasar el cursor sobre una carga de trabajo, se muestra información resumida de esa carga de trabajo, como puede ver en este ejemplo:



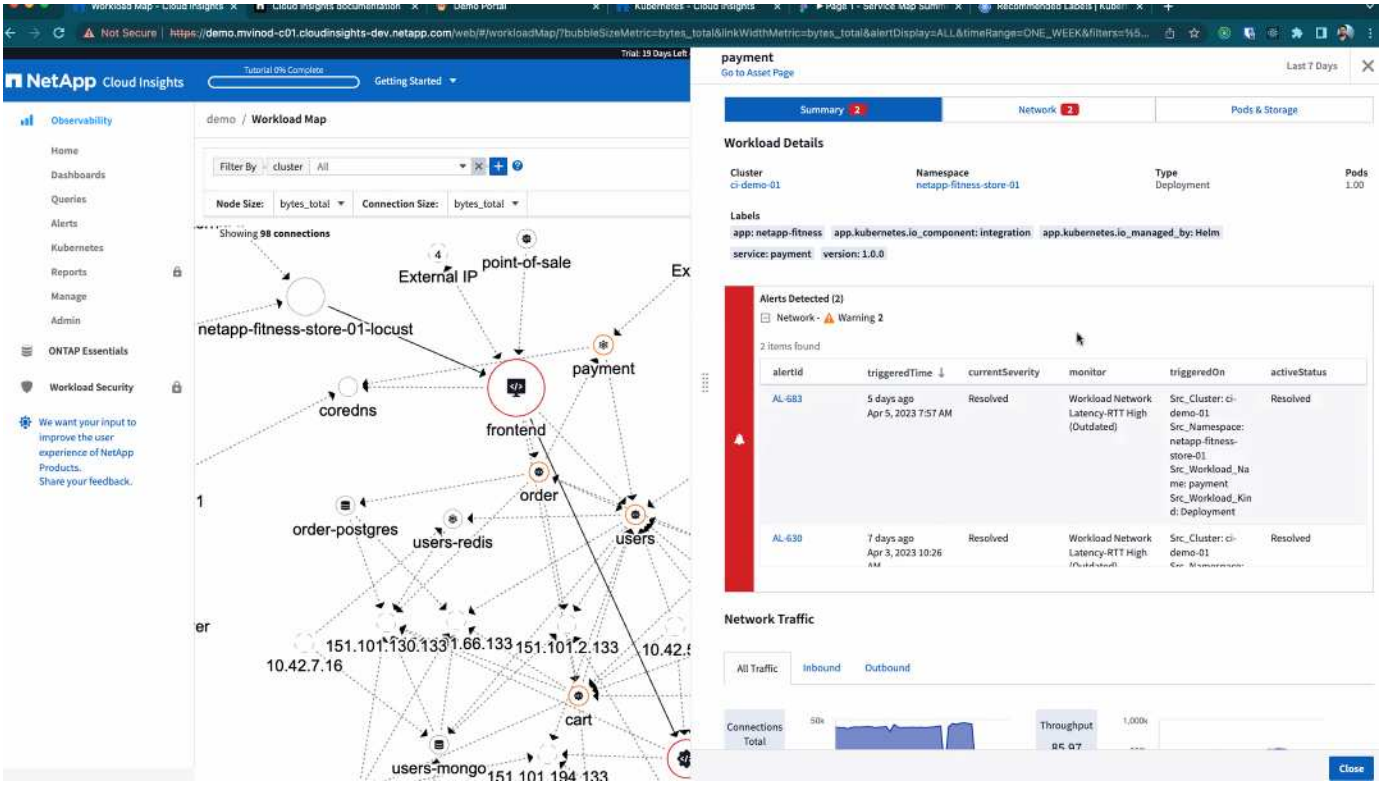
Los iconos dentro de los círculos representan diferentes tipos de servicios. Tenga en cuenta que los iconos solo son visibles si los objetos subyacentes tienenetiquetas .

El tamaño de cada círculo indica el tamaño del nodo. Tenga en cuenta que estos tamaños son relativos, el nivel de zoom de su navegador o el tamaño de la pantalla pueden afectar los tamaños reales de los círculos. De la misma manera, el estilo de línea de tráfico le brinda una vista rápida del tamaño de la conexión; las líneas sólidas en negrita representan mucho tráfico, mientras que las líneas punteadas claras representan poco tráfico.

Los números dentro de los círculos son la cantidad de conexiones externas que el servicio está procesando actualmente.

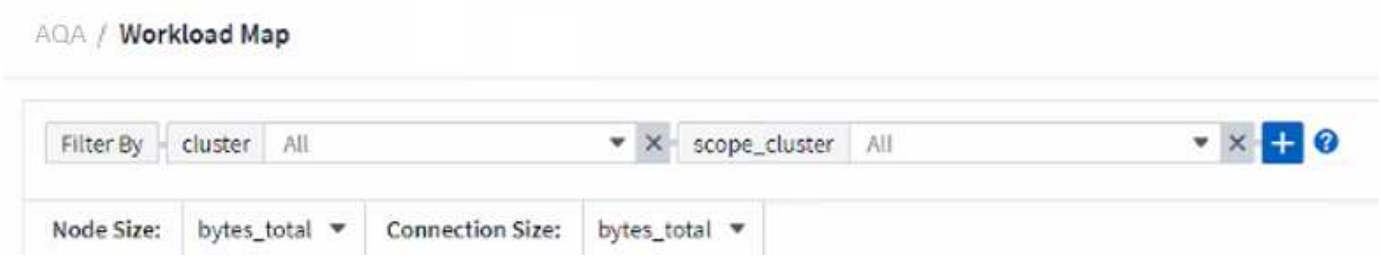
Detalles y alertas de la carga de trabajo

Los círculos que se muestran en color indican una alerta de nivel crítico o de advertencia para la carga de trabajo. Pase el cursor sobre el círculo para ver un resumen del problema o haga clic en el círculo para abrir un panel deslizable con más detalles.

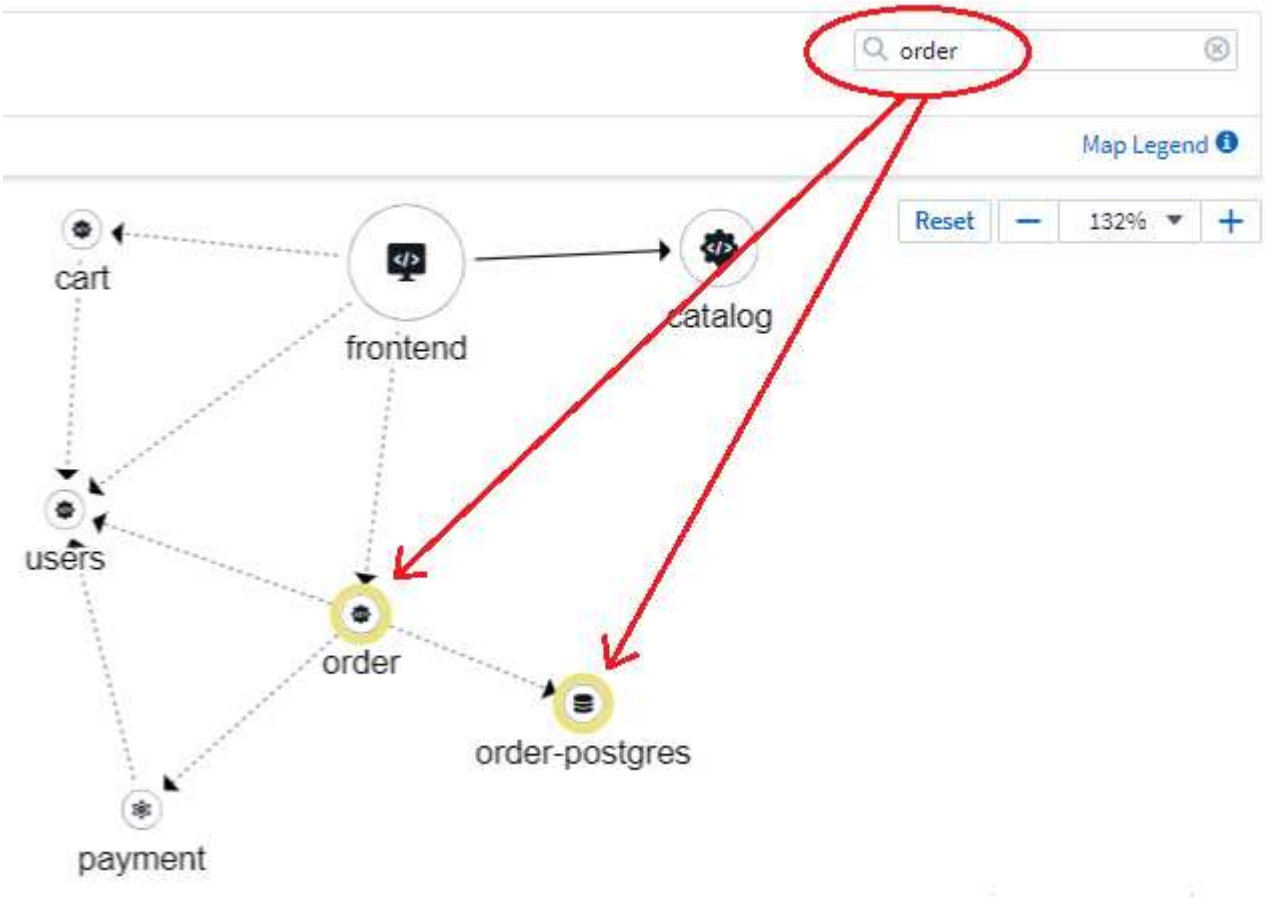


Búsqueda y filtrado

Al igual que con otras funciones de Data Infrastructure Insights , puede configurar fácilmente filtros para centrarse en los objetos específicos o los atributos de carga de trabajo que desee.



Del mismo modo, al escribir una cadena en el campo *Buscar* se resaltarán las cargas de trabajo coincidentes.



Etiquetas de carga de trabajo

Las etiquetas de carga de trabajo son necesarias si desea que el mapa identifique los tipos de cargas de trabajo que se muestran (es decir, los íconos circulares). Las etiquetas se derivan de la siguiente manera:

- Nombre del servicio/aplicación que se ejecuta en términos genéricos
- Si la fuente es un pod:
  - La etiqueta se deriva de la etiqueta de carga de trabajo del pod
  - Etiqueta esperada en la carga de trabajo: app.kubernetes.io/component
  - Referencia del nombre de la etiqueta: https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels/
  - Etiquetas recomendadas:
    - Interfaz

- backend
- base de datos
- cache
- cola
- Kafka

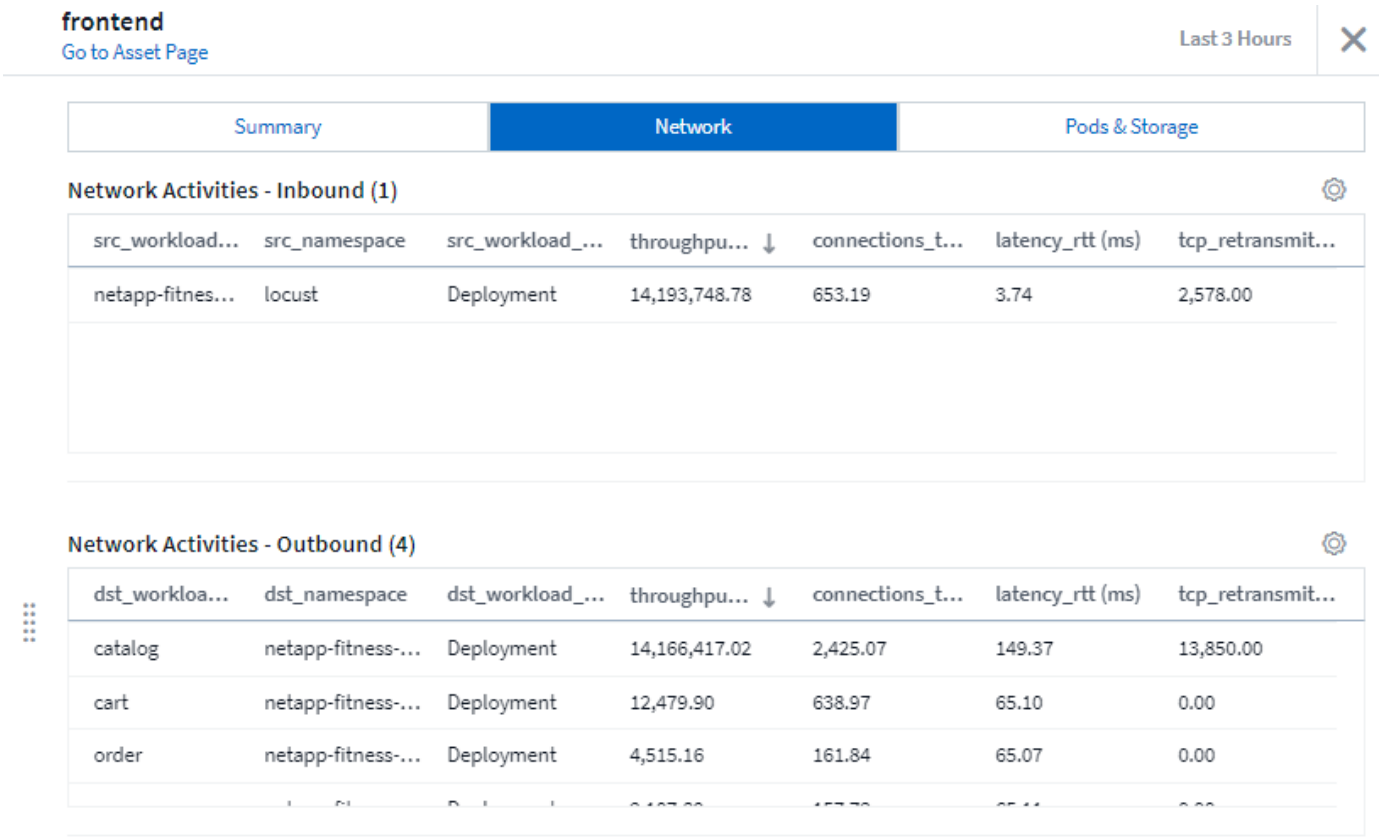
- Si la fuente es externa al clúster de Kubernetes:
  - Data Infrastructure Insights intentará analizar el nombre resuelto por DNS para extraer el tipo de servicio.

    Por ejemplo, con un nombre resuelto por DNS de *s3.eu-north-1.amazonaws.com*, el nombre resuelto se analiza para obtener *s3* como el tipo de servicio.

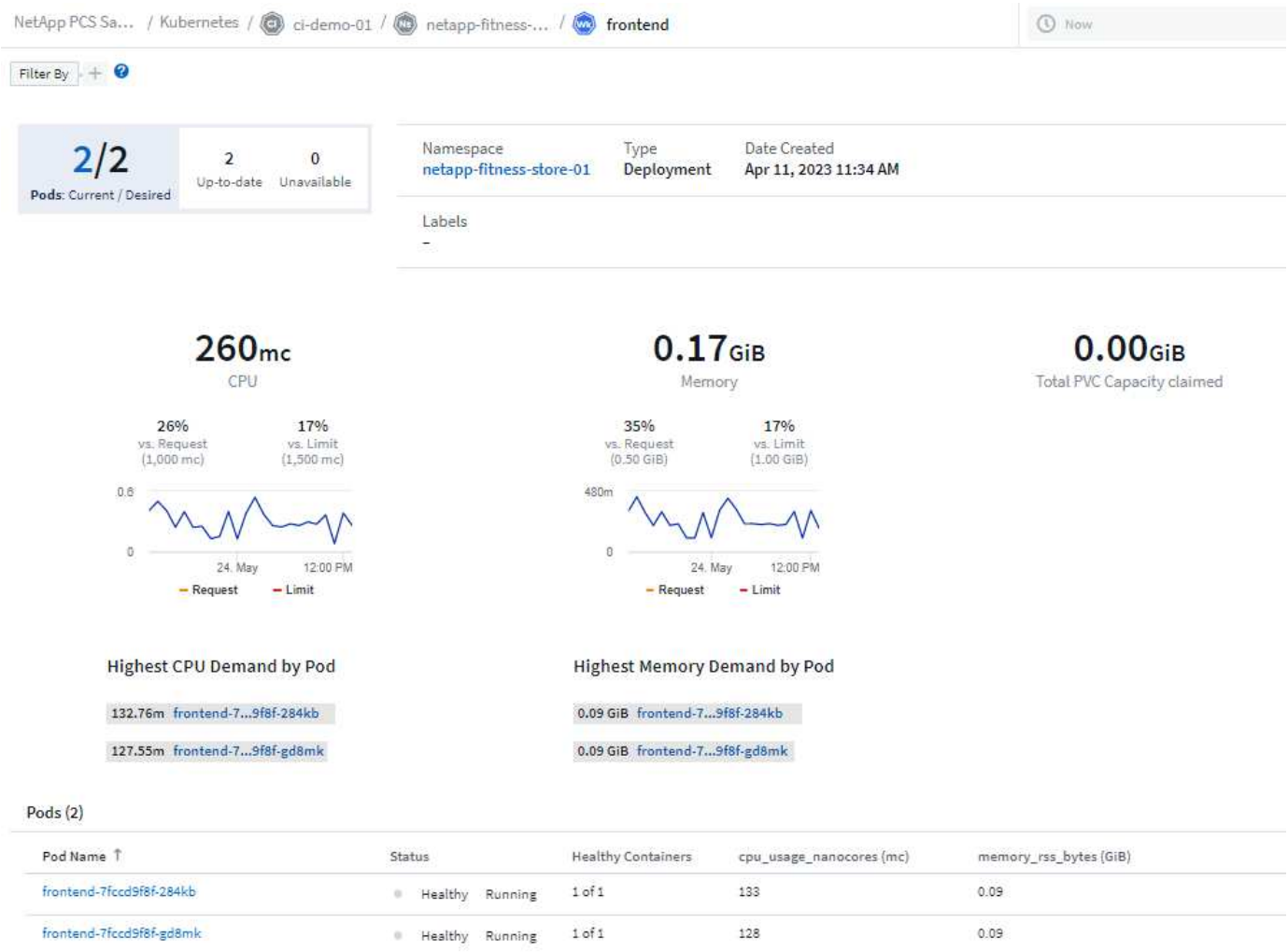
Sumérgete profundamente

Al hacer clic derecho en una carga de trabajo, se le presentan opciones adicionales para explorar más a fondo. Por ejemplo, desde aquí puedes hacer zoom para ver las conexiones para esa carga de trabajo.

O puede abrir el panel deslizable de detalles para ver directamente la pestaña *Resumen*, *Red* o *Pod y almacenamiento*.



Finalmente, al seleccionar *Ir a la página de activos* se abrirá la página de destino de activos detallada para la carga de trabajo.



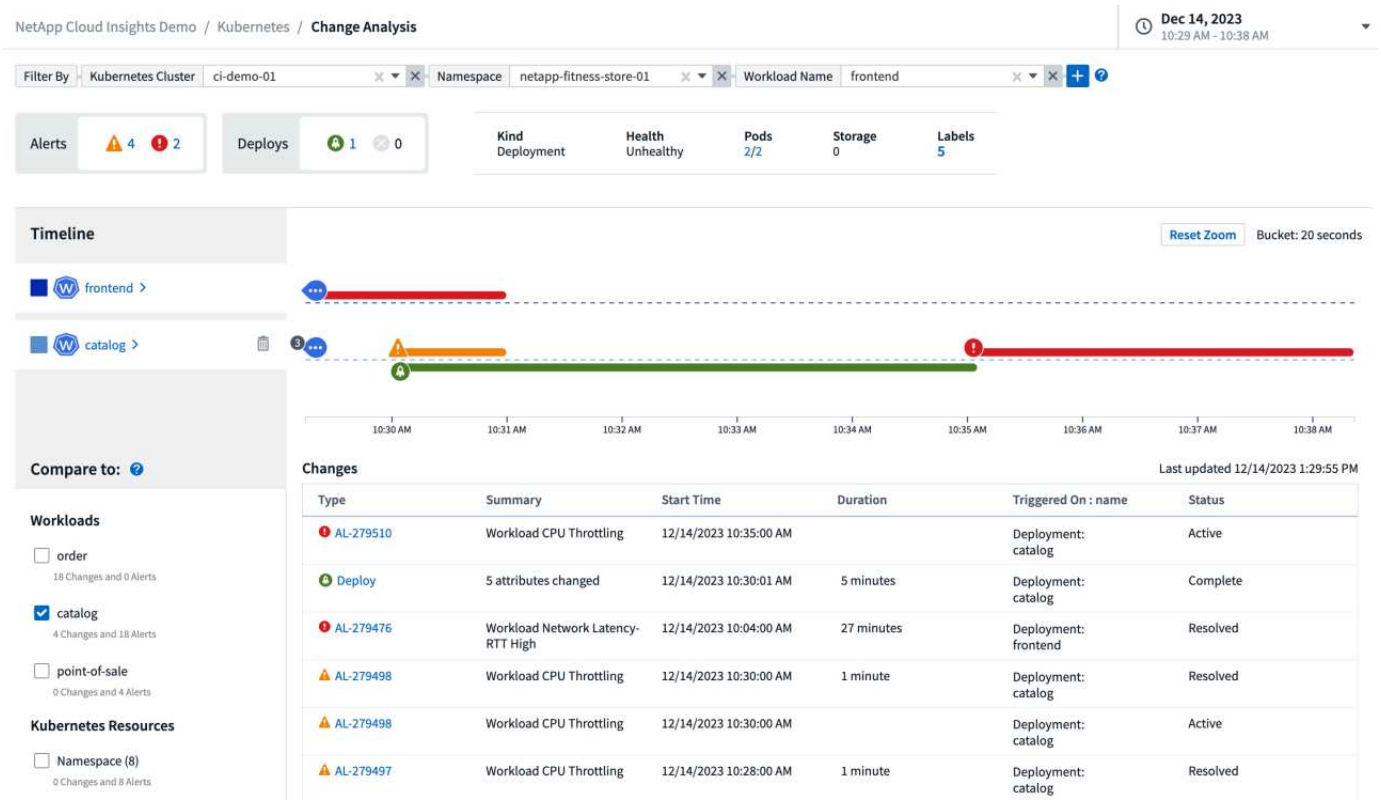
Análisis de cambios de Kubernetes

Kubernetes Change Analytics le proporciona una vista todo en uno de los cambios recientes en su entorno K8s. Las alertas y el estado de la implementación están a tu alcance. Con Change Analytics, puede realizar un seguimiento de cada cambio de implementación y configuración, y correlacionarlo con el estado y el rendimiento de los servicios, la infraestructura y los clústeres de K8.

¿Cómo ayuda el análisis de cambios?

- En entornos de Kubernetes de múltiples inquilinos, pueden producirse interrupciones debido a cambios mal configurados. Change Analytics ayuda con esto al proporcionar un panel único para ver y correlacionar el estado de las cargas de trabajo y los cambios de configuración. Esto puede ayudar a solucionar problemas en entornos dinámicos de Kubernetes.

Para ver Kubernetes Change Analytics, navegue a **Kubernetes > Análisis de cambios**.



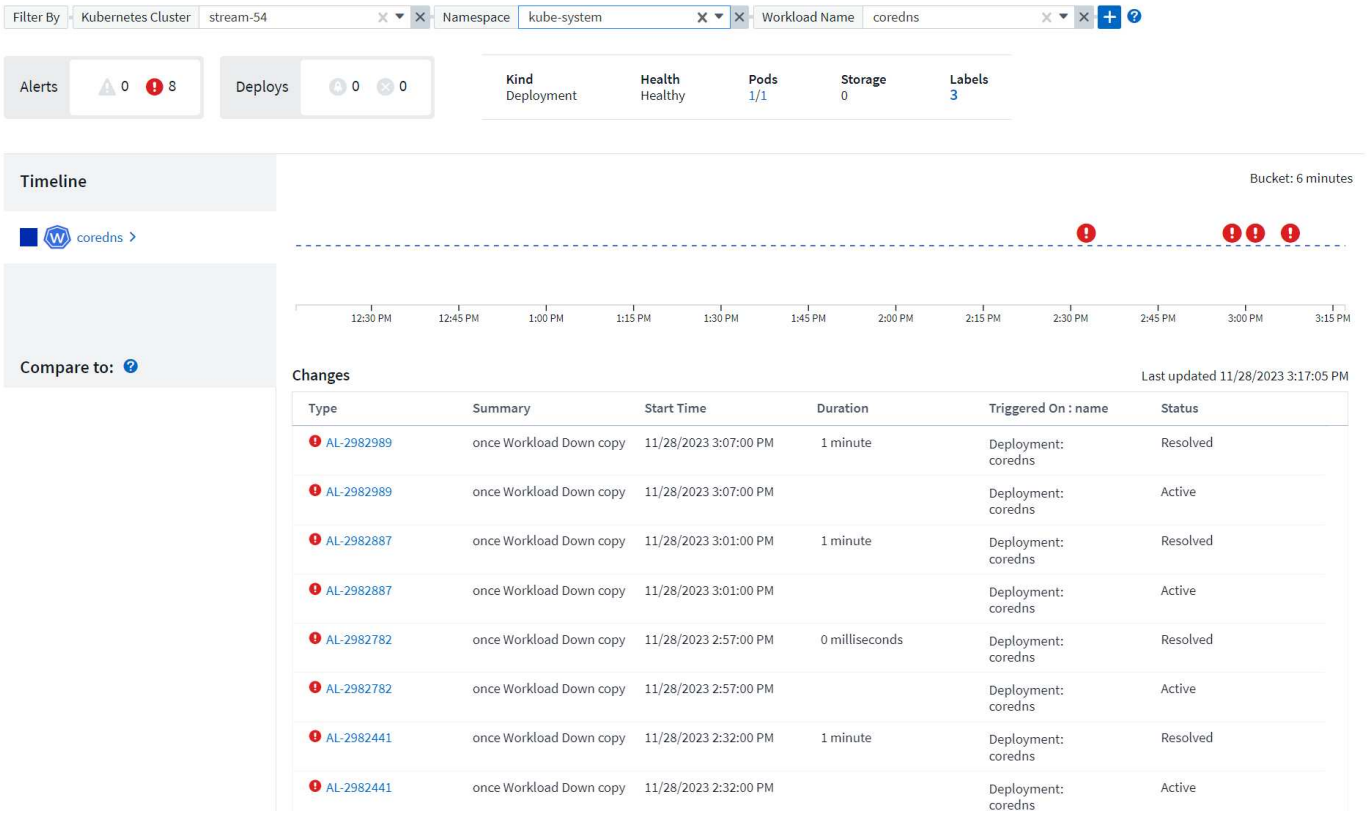
.

La página se actualiza automáticamente según el rango de tiempo de Data Infrastructure Insights seleccionado actualmente. Los rangos de tiempo más pequeños significan una actualización de pantalla más frecuente.

Filtración

Al igual que con todas las funciones de Data Infrastructure Insights, filtrar la lista de cambios es intuitivo: en la parte superior de la página, ingrese o seleccione valores para su clúster de Kubernetes, espacio de nombres o carga de trabajo, o agregue sus propios filtros seleccionando el botón {+].

Cuando filtra por un clúster, espacio de nombres y carga de trabajo específicos (junto con cualquier otro filtro que configure), se le muestra una línea de tiempo de implementaciones y alertas para esa carga de trabajo en ese espacio de nombres en ese clúster. Amplíe aún más el gráfico haciendo clic y arrastrando para centrarse en un rango de tiempo más específico.

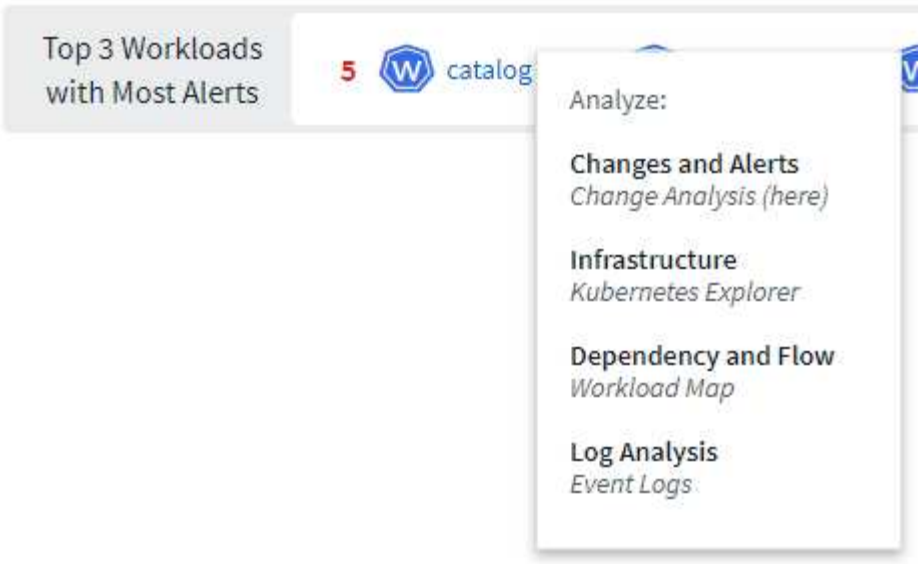


Estado rápido

Debajo del área de filtrado hay una serie de indicadores de alto nivel. A la izquierda se muestra el número de alertas (Advertencia y Crítica). Este número incluye alertas *Activas* así como *Resueltas*. Para ver solo alertas *Activas*, configure un filtro para "Estado" y elija "Activo".

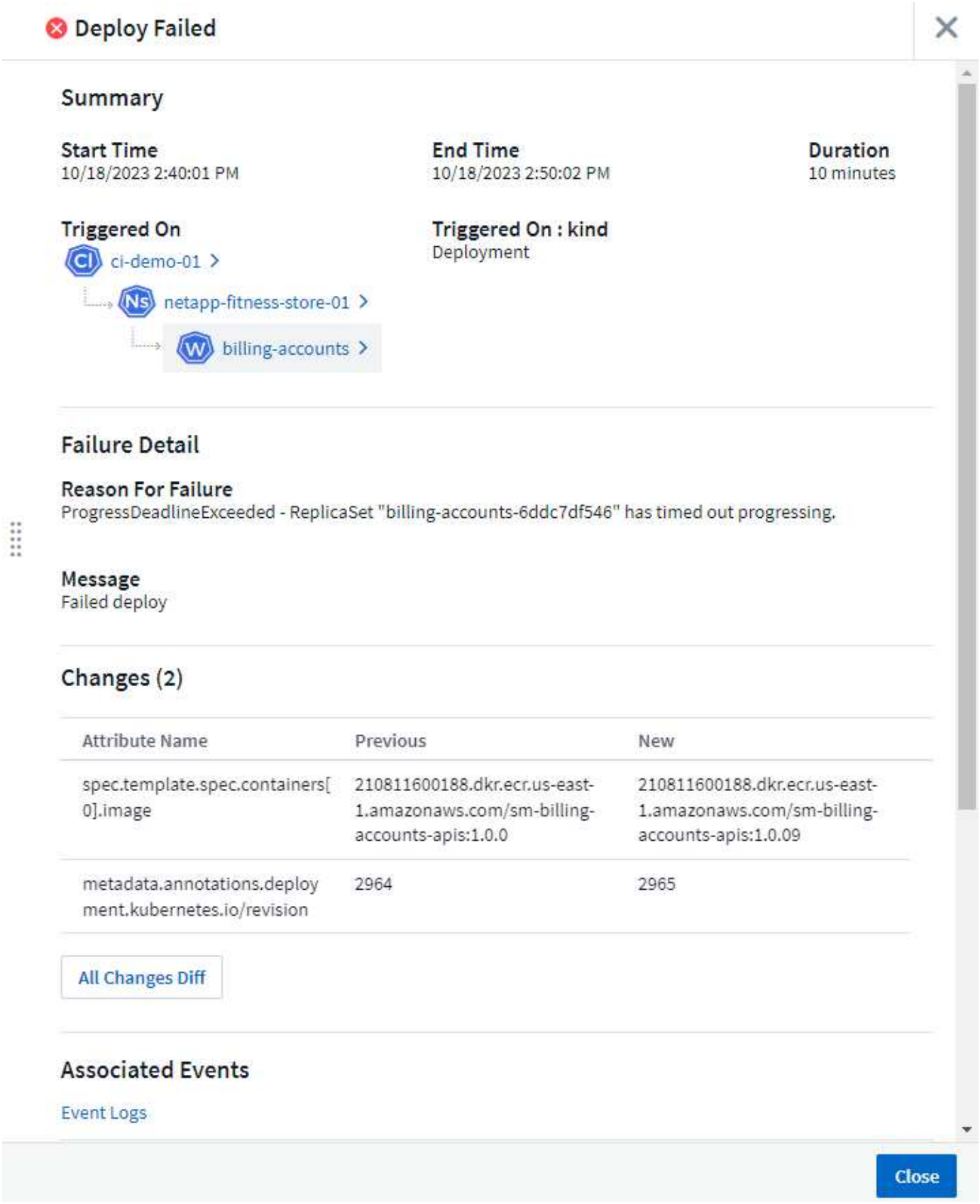
El estado de implementación también se muestra aquí. Nuevamente, el valor predeterminado es mostrar el recuento de implementaciones *Iniciadas*, *Completas* y *Fallidas*. Para ver solo las implementaciones *Fallidas*, configure un filtro para "Estado" y seleccione "Fallida".

A continuación se muestran las 3 cargas de trabajo principales con más alertas. El número en rojo junto a cada carga de trabajo indica la cantidad de alertas relacionadas con esa carga de trabajo. Haga clic en el enlace de carga de trabajo para explorar su infraestructura (Explorador de Kubernetes), dependencias (Mapa de carga de trabajo) o análisis de registros (Registros de eventos).



Panel de detalles

Al seleccionar un cambio en la lista, se abre un panel que describe el cambio con más detalle. Por ejemplo, al seleccionar una implementación fallida, se muestra un resumen de la implementación, con horas de inicio y finalización, duración y dónde se activó la implementación, con enlaces para explorar esos recursos. También muestra el motivo del fallo, cualquier cambio relacionado y cualquier evento asociado.



Al seleccionar una alerta también se proporcionan detalles sobre la alerta, incluido el monitor que la activó, así como un gráfico que muestra una línea de tiempo visual para la alerta.