Kubernetes

Data Infrastructure Insights

NetApp
December 19, 2024

Tabla de contenidos

Kubernetes

Descripción general del clúster Kubernetes

El explorador de Kubernetes de información sobre infraestructuras de datos es una potente herramienta para mostrar el estado y el uso generales de tus clústeres de Kubernetes y te permite profundizar fácilmente en áreas de investigación.

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

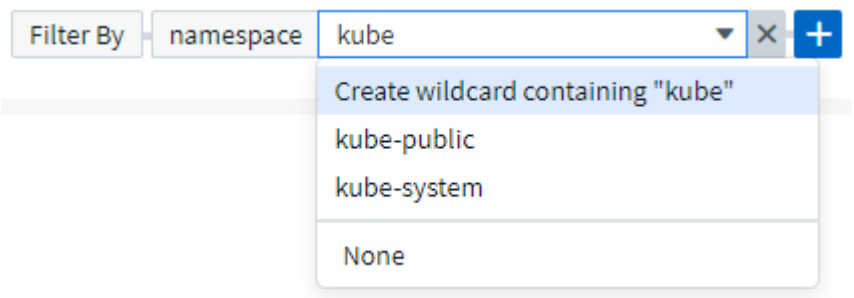
Lista Cluster

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

- Cluster **Nombre**. Al hacer clic en el nombre de un clúster se abrirá la **"página de detalles"** para ese clúster.
- **Porcentajes de saturación**. La saturación general es la más alta de saturación de CPU, memoria o almacenamiento.
- Número de **nodos** en el clúster. Al hacer clic en este número se abrirá la página de lista Node.
- Número de **pods** en el cluster. Al hacer clic en este número se abrirá la página de lista Pod.
- Número de **espacios de nombres** en el cluster. 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 abre la página de lista de cargas de trabajo.

Afinando el filtro

Cuando está filtrando, al comenzar a escribir, se le presenta la opción de crear un filtro * comodín* basado en el texto actual. Si selecciona esta opción, se devolverán todos los resultados que coincidan con la expresión comodín. También puede crear **expresiones** utilizando NOT O Y, o puede seleccionar la opción "Ninguno" para filtrar los valores nulos en el campo.



Los filtros basados en comodines o expresiones (p. ej., NOT, AND, "None", etc.) se muestran en azul oscuro en el campo de filtro. Los elementos seleccionados directamente de la lista se muestran en azul claro.

Los filtros de Kubernetes son contextuales, lo que significa, por ejemplo, que si se encuentra en una página específica del nodo, el filtro pod_name solo enumera los pods relacionados con ese nodo. Además, si aplica un filtro para un espacio de nombres específico, el filtro pod_name incluirá únicamente los pods de 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 valores numéricos, fechas o valores.

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

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

Componente	Requisito
La versión de Kubernetes	Kubernetes v1,20 y versiones posteriores.
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 funcionen con una arquitectura de Arm64. Supervisión de red: Debe ejecutar Linux kernel versión 4.18.0 o superior. No se admite el SO de fotones.
Etiquetas	Data Infrastructure Insights admite la supervisión de los nodos de Kubernetes que ejecutan Linux, especificando un selector de nodos de Kubernetes que busca las siguientes etiquetas de Kubernetes en estas plataformas: Kubernetes v1,20 y superiores: 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 la RUTA.

Componente	Requisito
Conectividad	la interfaz de línea de comandos de kubectl se configura 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 de "Configurar el soporte del proxy"la sección de instalación del operador. Para obtener informes precisos de auditoría y datos, sincronice la hora en el equipo del agente mediante Network Time Protocol (NTP) o Simple Network Time Protocol (SNTP).
Otros	Si está ejecutando OpenShift 4,6 o superior, debe seguir el "Instrucciones de OpenShift"y asegurarse de que se cumplen estos requisitos previos.
Token de API	Si va a volver a desplegar el Operador (es decir, lo está actualizando o reemplazando), no es necesario crear un nuevo token de API; puede volver a utilizar el token anterior.

Cosas importantes que debe tener en cuenta antes de comenzar

Si está ejecutando con un proxy, tiene un repositorio personalizado, o está utilizando OpenShift, lea detenidamente las siguientes secciones.

Lea también sobre Permisos.

Configurar el soporte del proxy

Hay dos lugares en los que puede usar un proxy en su inquilino para instalar el operador de monitoreo de Kubernetes de NetApp. Pueden ser los mismos sistemas proxy o independientes:

- Proxy necesario durante la ejecución del fragmento de código de instalación (mediante «curl») para conectar el sistema donde se ejecuta el fragmento a su entorno de Data Infrastructure Insights
- Proxy que necesita el clúster de Kubernetes de destino para comunicarse con su entorno de Data Infrastructure Insights

Si usa un proxy para una de estas o ambas, para instalar el monitor operativo de Kubernetes de NetApp, primero debe asegurarse de que su proxy esté configurado para permitir una buena comunicación con su entorno de información de infraestructura de datos. Por ejemplo, desde los servidores/VM desde los que desea instalar el Operador, debe poder acceder a Data Infrastructure Insights y poder descargar archivos binarios de Data Infrastructure Insights.

En el caso del proxy utilizado para instalar el monitor operativo de Kubernetes de NetApp, antes de instalar el operador, establezca las variables de entorno *http_proxy/https_proxy*. En algunos entornos proxy, también es posible que tenga que establecer la variable *no_proxy Environment*.

Para ajustar las variables, lleve a cabo los siguientes pasos en su sistema **antes de** instalar el operador de monitorización 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 estableciendo no tiene autenticación (nombre de usuario/contraseña), ejecute

el siguiente comando:

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

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

En el caso de que el proxy utilizado para su clúster de Kubernetes se comunique con su entorno de información sobre la infraestructura de datos, instale el operador de supervisión de Kubernetes de NetApp tras leer todas estas instrucciones.

Configure la sección proxy de AgentConfiguration en operator-config.yaml antes de implementar el operador de supervisión 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 de Docker personalizado o privado

De forma predeterminada, el operador de supervisión de Kubernetes de NetApp extraerá imágenes de contenedor del repositorio de información de infraestructura de datos. Si tiene un clúster de Kubernetes utilizado como destino para la supervisión, y ese clúster se configura para extraer solo imágenes de contenedor desde un repositorio de Docker privado o personalizado, debe configurar el acceso a los contenedores que necesita el operador de supervisión de Kubernetes de NetApp.

Ejecute «Image pull Snippet» desde el icono de instalación del operador de supervisión de NetApp. Este comando iniciará sesión en el repositorio de Data Infrastructure Insights, extraerá todas las dependencias de imágenes del operador y cerrará la sesión en el repositorio de Data Infrastructure Insights. Cuando se le solicite, introduzca la contraseña temporal del repositorio proporcionada. Este comando descarga todas las imágenes utilizadas por el operador, incluidas las funciones opcionales. Consulte a continuación las funciones para las que se utilizan estas imágenes.

Funcionalidad del operador principal y supervisión de Kubernetes

- supervisión de netapp
- proxy-rbac-kube
- métricas-estado-kube
- telegraf
- usuario raíz sin interrupciones

Registro de eventos

- bits fluidos
- exportador de eventos de kubernetes

Rendimiento de red y mapa

- ci-net-observador

Introduzca la imagen del operador docker en el repositorio de su proveedor de servicios de empresa/local/privado de acuerdo con las políticas de su empresa. Asegúrese de que las etiquetas de imagen y las rutas de directorio a estas imágenes del repositorio sean coherentes con las del repositorio de Data Infrastructure Insights.

Edite el despliegue de operador de supervisión en operator-deployment.yaml y modifique todas las referencias de imagen para utilizar su repositorio Docker privado.

```
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 de repositorio de Docker. Cree una nueva imagePullSecret para su repositorio privado, para 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 se ejecuta en OpenShift 4,6 o superior, debe editar la configuración de AgentConfiguration en *operator-config.yaml* para activar la configuración *runPrivileged*:

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

OpenShift puede implementar un nivel de seguridad añadido que puede bloquear el acceso a algunos componentes de Kubernetes.

Permisos

Si el cluster que está supervisando contiene recursos personalizados que no tienen un ClusterRole, lo que "agregados para ver", deberá otorgar manualmente al operador acceso a estos recursos para supervisarlos 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. Aplique los cambios al clúster

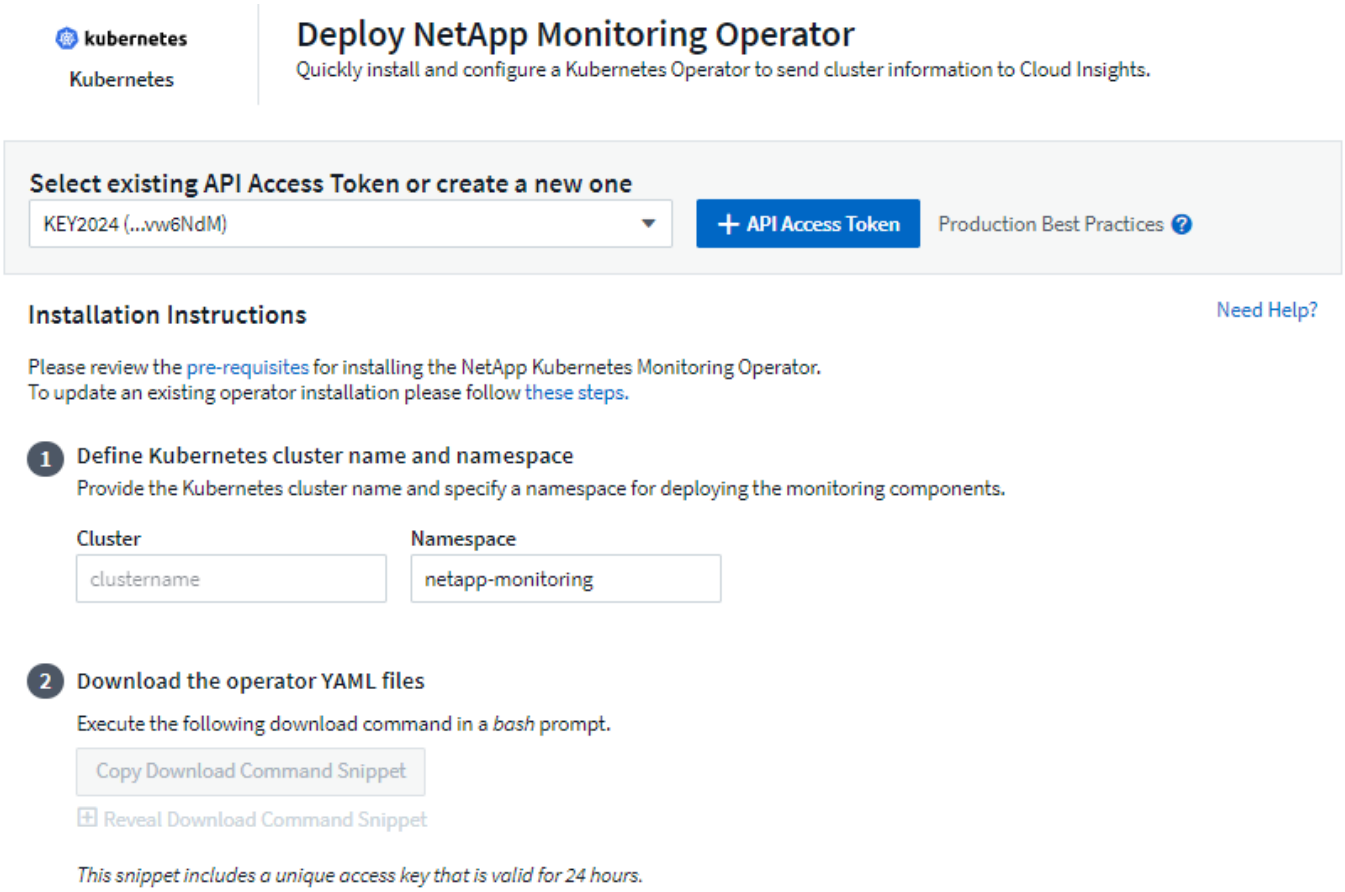
Instalación y configuración del operador de supervisión de Kubernetes

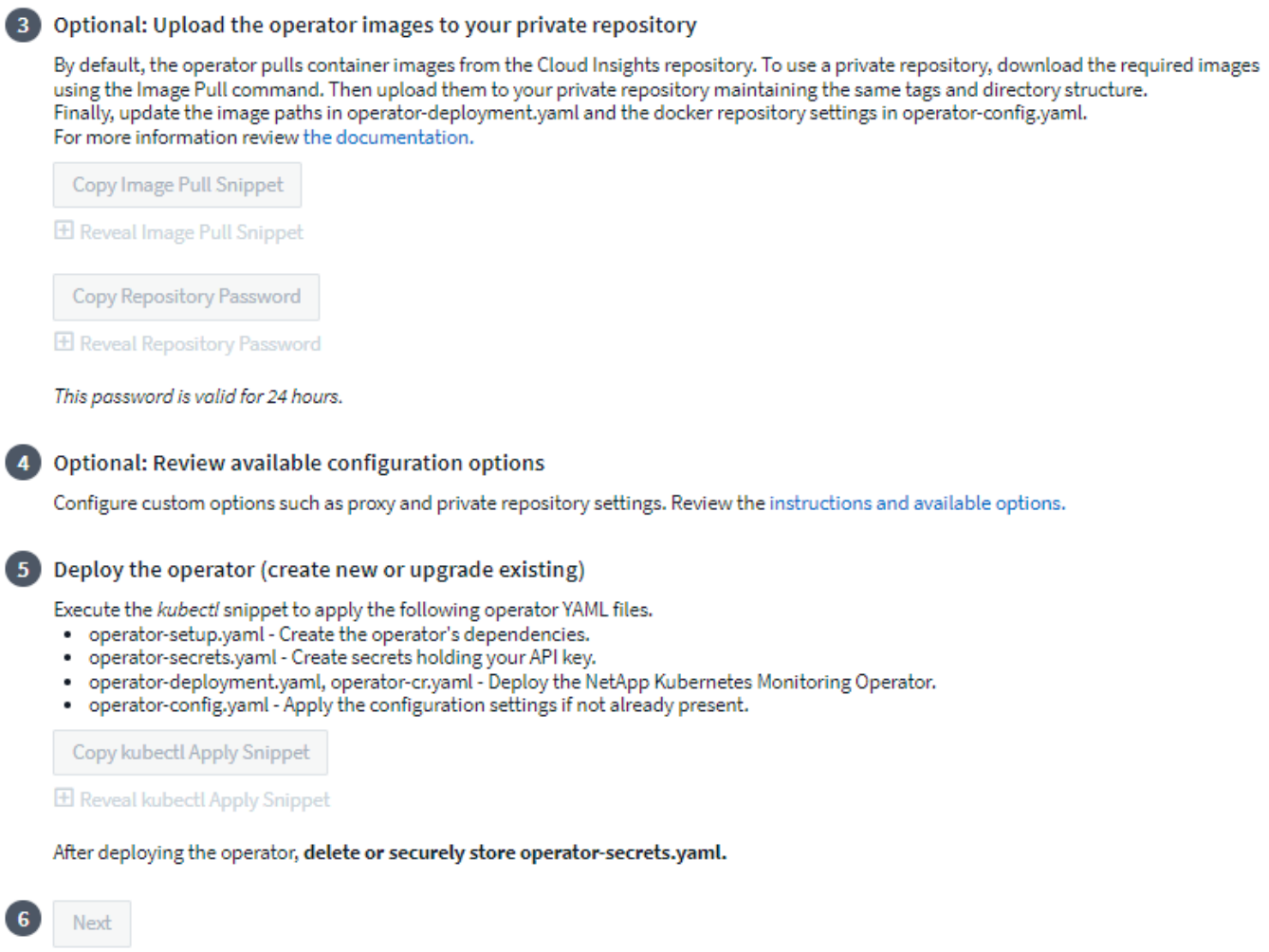
Data Infrastructure Insights ofrece el **Operador de Monitoreo de Kubernetes** para la colección de Kubernetes. Vaya a **Kubernetes > Colectores > +Kubernetes Collector** para implementar un nuevo operador.

Antes de instalar el operador de supervisión de Kubernetes

Consulte "Requisitos previos"la documentación antes de instalar o actualizar el operador de supervisión de Kubernetes.

Instalación del operador de supervisión de Kubernetes





Pasos para instalar el agente del operador de supervisión de Kubernetes en Kubernetes:

1. Introduzca un nombre de clúster y un espacio de nombres únicos. Si es actualizando de un operador de Kubernetes anterior, utilice el mismo nombre de clúster y espacio de nombres.
2. Una vez introducidos, puede copiar el fragmento del comando de descarga en el 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 opcional Image pull, péguelo en un shell *bash* y ejecútelo. Una vez extraídas las imágenes, cópielas en tu repositorio privado. Asegúrese de mantener las mismas etiquetas y la misma estructura de carpetas. Actualice las rutas de acceso 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 repositorio privado o proxy. Puedes leer más sobre "opciones de configuración".
6. Cuando esté listo, despliegue el Operador copiando el fragmento de aplicación kubectl, descargándolo y ejecutándolo.
7. La instalación se realiza automáticamente. Cuando haya terminado, haga clic en el botón *Next*.
8. Una vez finalizada la instalación, haga clic en el botón *Next*. Asegúrese también de eliminar o almacenar de forma segura el archivo *operator-secrets.yaml*.

Si está utilizando un proxy, lea acerca de configurando proxy.

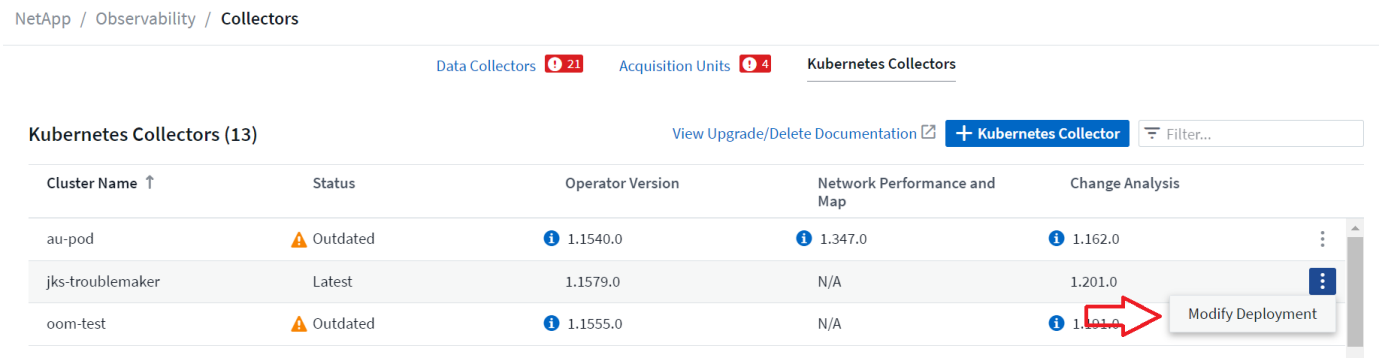
Si tiene un repositorio personalizado, lea acerca de utilizando un repositorio de docker personalizado/privado.

Componentes de supervisión de Kubernetes

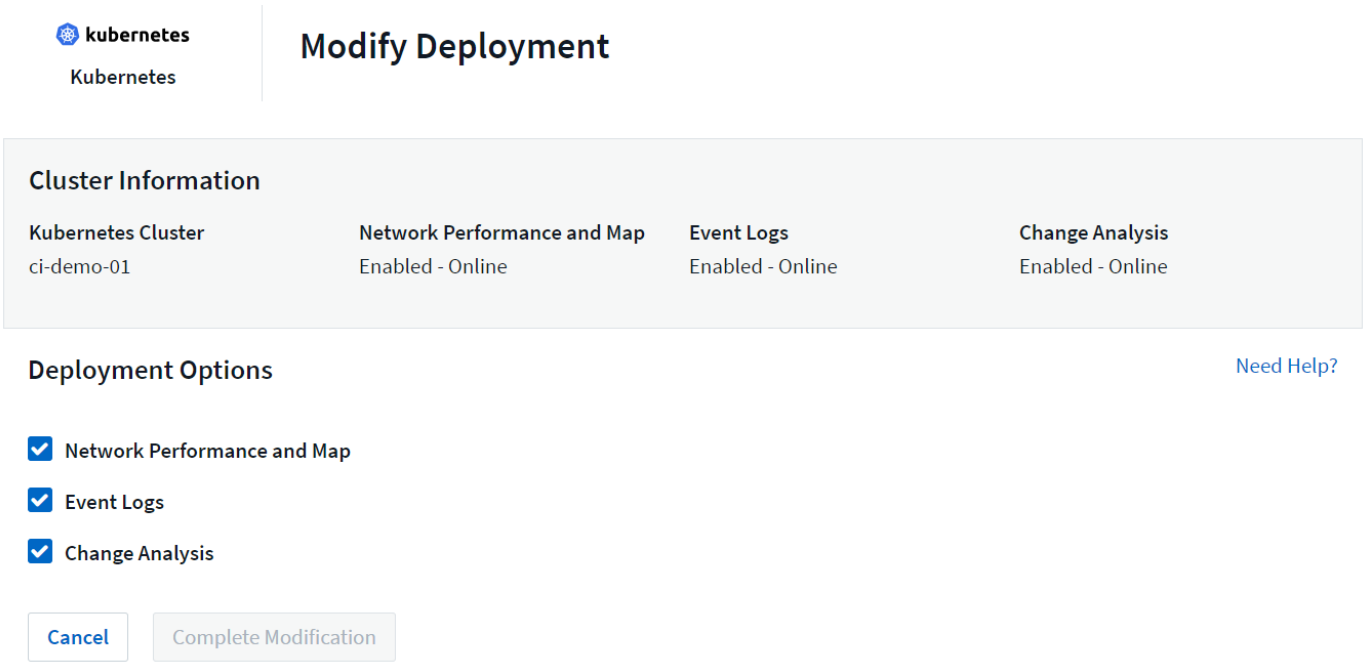
Información de la infraestructura de datos La supervisión de Kubernetes se compone de cuatro componentes de supervisión:

- Métricas de cluster
- Rendimiento de red y mapa (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 > Colectores** y seleccionando *Modify Deployment* 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 desactivar o activar componentes para ese recopilador según sea necesario.



Actualiza al operador de supervisión de Kubernetes más reciente

Determine si existe una AgentConfiguration con el operador existente (si el espacio de nombres no es el valor predeterminado *netapp-monitoring*, sustituya el espacio de nombres adecuado):

```
 kubectl -n netapp-monitoring get agentconfiguration netapp-monitoring-
configuration
Si existe una configuración de agente:
```

- Instale El último operador sobre el operador existente.
  - Asegúrese de que está extracción de las imágenes de contenedor más recientesutilizando un repositorio personalizado.

Si la configuración de agente no existe:

- Anote el nombre de su clúster como reconocido por Data Infrastructure Insights (si su espacio de nombres no es la supervisión NetApp predeterminada, sustituya el espacio de nombres adecuado):

```
 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 control de netapp predeterminado, sustituya el espacio
de nombres adecuado):
```

```
 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,Instale>> El último
operador.
```

  - Utilice el mismo nombre de clúster.
  - Después de descargar los últimos archivos YAML del operador, transfiera cualquier personalización encontrada en agent_backup.yaml al operator-config.yaml descargado antes de implementar.
  - Asegúrese de que está extracción de las imágenes de contenedor más recientesutilizando un repositorio personalizado.

Detener e iniciar el operador de supervisión de Kubernetes

Para detener el operador de supervisión de Kubernetes:

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

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

Desinstalando

Para eliminar el operador de supervisión de Kubernetes

Tenga en cuenta que el espacio de nombres predeterminado para el operador de supervisión de Kubernetes es la «supervisión de netapp». Si ha definido su propio espacio de nombres, sustituya este espacio de nombres en estos y todos los comandos y archivos subsiguientes.

Las versiones más recientes del operador de supervisión 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 supervisión se ha desplegado en su propio espacio de nombres dedicado, suprima el espacio de nombres:

```
 kubectl delete ns <NAMESPACE>
Si el primer comando devuelve "no se han encontrado recursos", utilice las
siguientes instrucciones para desinstalar versiones anteriores del
operador de supervisión.
```

Ejecute cada uno de los comandos siguientes en orden. Dependiendo de su instalación actual, algunos de estos comandos pueden devolver mensajes de 'no se ha encontrado el objeto'. Estos mensajes pueden ignorarse con seguridad.

```
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 ha creado previamente una restricción de contexto de seguridad:

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

Acerca de las métricas de estado de Kube

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

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

Configuración/Personalización del Operador

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

Opciones de configuración

La configuración más comúnmente modificada se puede configurar en el recurso personalizado *AgentConfiguration*. Puede editar este recurso antes de desplegar el operador editando el archivo *operator-config.yaml*. Este archivo incluye ejemplos de configuración comentados. Consulte la lista de "ajustes disponibles"para obtener la versión más reciente del operador.

También puede editar este recurso después de desplegar el operador mediante el siguiente comando:

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

```
 kubectl get crd agentconfigurations.monitoring.netapp.com
Si ve un mensaje “Error from server (NotFound)”, su operador debe
actualizarse antes de poder usar AgentConfiguration.
```

Configurar el soporte del proxy

Hay dos lugares en los que puede usar un proxy en su inquilino para instalar el operador de monitoreo de Kubernetes. Pueden ser los mismos sistemas proxy o independientes:

- Proxy necesario durante la ejecución del fragmento de código de instalación (mediante «curl») para conectar el sistema donde se ejecuta el fragmento a su entorno de Data Infrastructure Insights
- Proxy que necesita el clúster de Kubernetes de destino para comunicarse con su entorno de Data Infrastructure Insights

Si usas un proxy para una o ambas de ellas, para instalar el Monitor Operativo de Kubernetes, primero debes asegurarte de que tu proxy esté configurado para permitir una buena comunicación con tu entorno de Información de Infraestructura de Datos. Si tiene un proxy y puede acceder a Data Infrastructure Insights desde el servidor/VM desde el que desea instalar el Operador, es probable que su proxy esté configurado correctamente.

Para el proxy utilizado para instalar el monitor operativo de Kubernetes, antes de instalar el operador, defina las variables de entorno *http_proxy/https_proxy*. En algunos entornos proxy, también es posible que tenga que establecer la variable *no_proxy Environment*.

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 estableciendo no tiene autenticación (nombre de usuario/contraseña), ejecute el siguiente comando:

```
 export https_proxy=<proxy_server>:<proxy_port>
.. Si el proxy que se está estableciendo 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 Información de infraestructura de datos, instale el operador de supervisión de Kubernetes después de leer todas estas instrucciones.

Configure la sección proxy de AgentConfiguration en operator-config.yaml antes de implementar el operador de supervisión 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 de Docker personalizado o privado

De forma predeterminada, el operador de supervisión de Kubernetes extraerá imágenes de contenedor del repositorio de información de infraestructura de datos. Si tiene un clúster de Kubernetes utilizado como destino para la supervisión, y ese clúster está configurado para extraer solo imágenes de contenedor de un repositorio Docker privado o personalizado o un registro de contenedores, debe configurar el acceso a los contenedores que necesita el operador de supervisión de Kubernetes.

Ejecute «Image pull Snippet» desde el icono de instalación del operador de supervisión de NetApp. Este comando iniciará sesión en el repositorio de Data Infrastructure Insights, extraerá todas las dependencias de imágenes del operador y cerrará la sesión en el repositorio de Data Infrastructure Insights. Cuando se le solicite, introduzca la contraseña temporal del repositorio proporcionada. Este comando descarga todas las imágenes utilizadas por el operador, incluidas las funciones opcionales. Consulte a continuación las funciones para las que se utilizan estas imágenes.

Funcionalidad del operador principal y supervisión de Kubernetes

- supervisión de netapp
- ci-kube-rbac-proxy
- ci-ksm
- ci-telegraf
- usuario raíz sin interrupciones

Registro de eventos

- bits ci-fluido

- ci-kubernetes-event-exporter

Rendimiento de red y mapa

- ci-net-observador

Introduzca la imagen del operador docker en el repositorio de su proveedor de servicios de empresa/local/privado de acuerdo con las políticas de su empresa. Asegúrese de que las etiquetas de imagen y las rutas de directorio a estas imágenes del repositorio sean coherentes con las del repositorio de Data Infrastructure Insights.

Edite el despliegue de operador de supervisión en operator-deployment.yaml y modifique todas las referencias de imagen para utilizar su repositorio Docker privado.

```
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 de repositorio de Docker. Cree una nueva imagePullSecret para su repositorio privado, para 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 se ejecuta en OpenShift 4,6 o superior, debe editar la configuración de AgentConfiguration en *operator-config.yaml* para activar la configuración *runPrivileged*:

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

OpenShift puede implementar un nivel de seguridad añadido que puede bloquear el acceso a algunos componentes de Kubernetes.

Toleraciones y daños

Los *netapp-ci-telegraf-ds*, *netapp-ci-fluent-bit-ds* y *netapp-ci-net-observer-L4-ds* DaemonSets deben programar un pod en cada nodo del clúster para recopilar correctamente los datos en todos los nodos. El operador ha sido configurado para tolerar algunos **taints** bien conocidos. Si ha configurado algún daño personalizado en sus nodos, evitando así que los pods se ejecuten en cada nodo, puede crear una **tolerancia** para esos daños"En el campo *AgentConfiguration*". Si ha aplicado daños personalizados a todos los nodos del cluster, también debe agregar las toleraciones necesarias al despliegue del operador para permitir que el pod del operador se programe y ejecute.

Más información sobre Kubernetes "Tolerancias y taints".

Vuelva a la **"NetApp Kubernetes Monitoreo de la página de instalación del operador"**

Una nota sobre los secretos

Para eliminar el permiso del operador de supervisión de Kubernetes para ver los secretos en todo el clúster, elimine los siguientes recursos del archivo *operator-setup.yaml* antes de instalar:

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

Si se trata de una actualización, suprima también los recursos del 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 activado, modifique *AgentConfiguration* o *operator-config.yaml* para anular el comentario de la sección de gestión de cambios e incluya *kindsToIgnoreFromWatch: "secrets"* en la sección de gestión de cambios. Observe 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 imagen del operador de supervisión de Kubernetes

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

Kubernetes".

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

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

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

```
$ 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}]
```

Resolución de problemas

Algunas cosas que debe probar si encuentra problemas para configurar el operador de supervisión de Kubernetes:

Problema:	Pruebe lo siguiente:
No veo un hipervínculo/conexión entre mi volumen persistente Kubernetes y el dispositivo de almacenamiento back-end correspondiente. Mi volumen persistente de Kubernetes se configura usando el nombre de host del servidor de almacenamiento.	Siga los pasos para desinstalar el agente de Telegraf existente y, a continuación, vuelva a instalar el último agente de Telegraf. Debe utilizar Telegraf versión 2,0 o posterior, y Data Infrastructure Insights debe supervisar de forma activa su almacenamiento en clúster de Kubernetes.

Problema:	Pruebe lo siguiente:
Estoy viendo mensajes en los registros que se asemejan a lo siguiente: E0901 15 352:21 v1:39,962145 1 k8s reflector.go:43,168161 1] k8s.io/kube-state-metrics/internal/store/builder.go:352: Error al mostrar *v1.MutatingWebhookConfiguration: El servidor no pudo encontrar el recurso solicitado 178:k8s:178 reflector.go:E0901 15] 21.io/kube-state-leases/leases: No se pudo encontrar las métricas internas del servidor *log.log.lease_leases/server.log.log.log.log.leases	Estos mensajes pueden aparecer si ejecuta métricas de estado kube versión 2.0.0 o posteriores 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 deployment/kube-state-Metrics -o jsonpath='{..image}'* para evitar que estos mensajes ocurran, los usuarios pueden modificar su implementación de kube-state-Metrics para desactivar los siguientes arrendamientos: *Mulatingweblookingdeads puede usar específicamente las configuraciones de webs*. Recursos=certififeligingRequests,configmaps,cronjobs,demonsets,despliegues,Endpoints,horizontal,podautocalers,ingesses,trabajos,limitrangos, espacios de nombres,networkpolds,nodos,persistenteclaims,persistentvolumes,podritionmars,poss,poss,netmasposs,poss,poss,possitaposs,poss,poss,posavapposs,poss,poss,poss,poss,poss,netmasposs,poss,possitaposs,possita,poss,poss,poss,possitaposs,poss,poss,possita,poss,poss,poss,possitaposs,poss,possita,poss,poss,possita,poss,possita,poss,poss,possita,poss,poss,possita,possi validarconexiones web, volumeadjuntos"
Veo mensajes de error de Telegraf que se parecen a lo siguiente, pero Telegraf se inicia y ejecuta: Oct 11 14:23:41 ip-172-31-39-47 systemd[1]: Se ha iniciado el agente de servidor basado en plugin para informar las métricas en InfluxDB. Oct 11 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” Oct 2021 41Z:10 ip-23-31-39-47 telegmsraf[1827]: Time=”11 14-23:41=error abierto a nivel:172= Open /etc/telegraf/.cache/snowflake/ocsp_response_cache.json: No tal archivo o directorio\n” func=“gosnowflake.(*defaultLogger).Errorf file=“log.go:120” Oct 2021 41Z:10 ip-23-31-39-47 telegraf[1827]: 11 14-23:41-11T14:172! Arranque de Telegraf 1.19.3	Este es un problema conocido. Consulte "Este artículo de GitHub"si desea obtener más información. Mientras Telegraf esté activo y en funcionamiento, los usuarios pueden ignorar estos mensajes de error.
En Kubernetes, My Telegraf pod/s notifican el siguiente error: "Error al procesar mountstats info: Error al abrir el archivo mountstats: /Hostfs/proc/1/mountstats, error: Open /hostfs/proc/1/mountstats: Permission denegado"	Si SELinux está habilitado y se aplica, 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 de agentconfiguration y active la configuración runPrivileged. Para obtener más información, consulte la "Instrucciones de OpenShift".

Problema:	Pruebe lo siguiente:
En Kubernetes, mi pod Telegraf ReplicaSet informa del siguiente error: [inputs.prometheus] error en el plugin: No se pudo cargar keypair /etc/kubernetes/pki/etcd/Server.crt:/etc/kubernetes/pki /etcd/Server.key: Open /etc/kubernetes/pki/etcd/Server.crt: No existe ese archivo o directorio	El Pod Telegraf ReplicaSet está diseñado para ejecutarse en un nodo designado como maestro o etcd. Si el Pod ReplicaSet no se está ejecutando en uno de estos nodos, obtendrá estos errores. Compruebe si los nodos maestro/etcd tienen sugerencias. Si lo hacen, añada las toleraciones necesarias al Telegraf ReplicaSet, telegraf-rs. Por ejemplo, edite ReplicaSet… kubectl edite rs telegraf-rs… y añada las toleraciones adecuadas a la especificación. A continuación, reinicie el Pod ReplicaSet.
Tengo un entorno PSP/PSA. ¿Afecta esto a mi operador de supervisión?	Si su clúster de Kubernetes se ejecuta con la política de seguridad de Pod (PSP) o la admisión de seguridad de Pod (PSA), debe actualizar al último operador de supervisión de Kubernetes. Siga estos pasos para actualizar al Operador actual con soporte para PSP/PSA: 1. Desinstalar el operador de supervisión anterior: kubectl delete 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-cluster-rolebinding agent-2. Instale la última versión del operador de monitorización.
Me encontré con problemas tratando de implementar el Operador, y tengo PSP/PSA en uso.	1. Edite el agente con el siguiente comando: Kubectl -n <name-space> edit agent 2. Marque "Security-policy-enabled" como "false". Esto desactivará las políticas de seguridad de Pod y la admisión de seguridad de Pod y permitirá que el operador se despliegue. Confirme utilizando los siguientes comandos: Kubectl Get psp (debería mostrar la política de seguridad de Pod eliminada) knotbtl get all -n <namespace>
grep -i psp (debería mostrar que no se encuentra nada)	Se han visto errores "ImagePullBackoff"
Estos errores pueden verse si tiene un repositorio de Docker personalizado o privado y aún no ha configurado el operador de supervisión de Kubernetes para reconocerlo correctamente. Leer más acerca de la configuración para repositorio personalizado/privado.	Tengo un problema con la implementación de mi operador de supervisión y la documentación actual no me ayuda a resolverla.

Problema:	Pruebe lo siguiente:

Capture o anote el resultado de los siguientes comandos y póngase en contacto 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 (Workload Map) en el espacio de nombres del operador están en CrashLoopBackOff

Estos pods corresponden al recopilador de datos de asignación de cargas de trabajo para la observabilidad de red. Pruebe estos: • Compruebe 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»: «Failed in validation. Razón: La versión del kernel 3.10.0 es menor que la versión mínima del kernel de 4.18.0”, “tiempo”: “2022-11-09T08:23:08Z”} ---- • Los pods de Net-Observer requieren que la versión del kernel de Linux sea al menos 4.18.0. Compruebe la versión del núcleo con el comando “uname -r” y asegúrese de que son >= 4.18.0	Los pods se ejecutan en el espacio de nombres del operador (predeterminado: Supervisión de netapp), pero no se muestran datos en la interfaz de usuario para el mapa de cargas de trabajo o las métricas de Kubernetes en consultas
Compruebe la configuración de hora en los nodos del clúster K8S. Para obtener informes precisos de auditoría y datos, se recomienda encarecidamente sincronizar la hora en el equipo del agente mediante el Protocolo de hora de red (NTP) o el Protocolo de hora de red simple (SNTP).	Algunos de los pods del observador de red en el espacio de nombres del operador están en estado Pendiente
NET-observer es un DaemonSet y ejecuta un pod en cada nodo del cluster k8s. • Observe el pod que está en estado Pendiente y compruebe si está experimentando un problema de recursos para la CPU o la memoria. Asegúrese de que la memoria y la CPU requeridas estén disponibles en el nodo.	Estoy viendo lo siguiente en mis registros inmediatamente después de instalar el operador de supervisión de Kubernetes: [inputs.prometheus] Error en el plugin: Error al hacer la solicitud HTTP a http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: Get http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: Dial tcp: Lookup kube-state-metrics.<namespace>.svc.local: No hay tal host

Problema:	Pruebe lo siguiente:
Este mensaje normalmente solo aparece cuando se instala un nuevo operador y el pod *telegraf-rs* está activo antes de que el pod *ksm* esté activo. Estos mensajes deben detenerse una vez que todos los pods se estén ejecutando.	No veo que se esté recopilando ninguna métrica para los cronjobs de Kubernetes que existen en mi clúster.
Compruebe la versión de Kubernetes (es decir, `kubectl version`). Si es v1,20.x o inferior, esta es una limitación esperada. La versión de métricas de estado de kube implementada con el operador de supervisión de Kubernetes solo admite v1.cronjob. Con Kubernetes 1,20.x y más abajo, 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 de telegraf-ds ingresan CrashLoopBackOff y los registros de pod indican “su: Error de autenticación”.
Edite la sección telegraf en *AgentConfiguration* y establezca *dockerMetricCollectionEnabled* en false. Para obtener más información, consulte la sección del operador "opciones de configuración". … spec: … telegraf: … - Nombre: docker run-mode : - DaemonSet substitutions : - Clave: DOCKER_unix_SOCK_PLACEHOLDER valor: unix://run/docker.SOCK … …	Veo mensajes de error repetidos que se parecen a los siguientes en mis registros de Telegraf: E! [Agent] Error al escribir en outputs.http: Post «\https://<tenant_url>/rest/v1/lake/ingest/influxdb»: Fecha límite de contexto excedida (Cliente. Se ha excedido el tiempo de espera de cabeceras)
Edite la sección telegraf en *AgentConfiguration* y aumente *outputTimeout* a 10s. Para obtener más información, consulte la sección del operador "opciones de configuración".	Faltan datos *involved dobject* para algunos registros de eventos.
Asegúrese de haber seguido los pasos de la "Permisos" sección anterior.	¿Por qué veo que funcionan dos pods del operador de supervisión, uno llamado netapp-ci-monitoring-operator-<pod> y otro llamado monitoring-operator-<pod>?
A partir del 12 de octubre de 2023, Data Infrastructure Insights ha refactorizado el operador para prestar un mejor servicio a nuestros usuarios; para que esos cambios se adopten por completo, debe retire el operador antiguo y instale la nueva.	Los eventos de My kubernetes dejaron de generar informes inesperadamente para la información de Data Infrastructure.
Recupere el nombre del pod de evento-exportador: `` `kubectl -n netapp-monitoring get pods ``	grep event-exporter

Problema:	Pruebe lo siguiente:
awk '{print $1}'	sed 's/event-exporter./event-exporter/'` Debe ser «exportador-de-centro-eventos-netapp» o «exportador-de-eventos». A continuación, edite el agente de supervisión `kubectl -n netapp-monitoring edit agent` y defina el valor para LOG_FILE para reflejar el nombre de pod de evento-exportador adecuado encontrado en el paso anterior. Más concretamente, EL ARCHIVO_REGISTRO 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, uno también puede desinstalary vuelva a instalar el agente.
Estoy viendo que los pods han sido puestos en marcha por el operador de supervisión de Kubernetes se han bloqueado debido a la falta de recursos.	Consulte el operador de supervisión de Kubernetes "opciones de configuración"para aumentar los límites de CPU y/o memoria según sea necesario.
La falta de una imagen o una configuración no válida provocó que los pods de métricas de estado de netapp-ci-kube no se iniciaran o estuvieran listos. Ahora, StatefulSet se bloquea y los cambios de configuración no se aplican a los pods de métricas de estado-ci-kube.	El StatefulSet está en un "roto" estado. Después de resolver cualquier problema de configuración, renueve los pods de métricas de estado-ci-kube-state.
Los pods de métricas de estado-ci-kube-state no se pueden iniciar tras ejecutar una actualización del operador de Kubernetes y lanzar ErrImagePull (no lograr extraer la imagen).	Intente restablecer los pods manualmente.
Los mensajes de «Event descarded as as as older then maxEventAgeSeconds» se observan para mi clúster de Kubernetes en Log Analysis.	Modifique el Operador *agentconfiguration* y aumente el *event-exporter-maxEventAgeSeconds* (es decir, a 60s), *event-exporter-kubeQPS* (es decir, a 100) y *event-exporter-kubeBurst* (es decir, a 500). Para obtener más información sobre estas opciones de configuración, consulte la "opciones de configuración" página.

Problema:	Pruebe lo siguiente:
Telegraf advierte de, o se bloquea debido a, memoria bloqueable insuficiente.	Intente aumentar el límite de memoria bloqueable para Telegraf en el sistema operativo/nodo subyacente. Si aumentar el límite no es una opción, modifique la configuración de agentconfiguration NKMO y establezca *UNPROTECTED* en *TRUE*. Esto indicará a Telegraf que no intente reservar páginas de memoria bloqueadas. Aunque esto puede suponer un riesgo para la seguridad, ya que los secretos descifrados se pueden intercambiar en el disco, permite su ejecución en entornos en los que no es posible reservar la memoria bloqueada. Para obtener más información sobre las opciones de configuración *UNPROTECTED*, consulte la "opciones de configuración" página.
Veo mensajes de advertencia de Telegraf parecidos a los siguientes: *W! [Inputs.diskio] No se puede recopilar el nombre del disco para "vdc": Error al leer /dev/vdc: No tal archivo o directorio*	Para el operador de supervisión de Kubernetes, estos mensajes de advertencia son benignos y se pueden ignorar con seguridad. Alternativamente, edite la sección telegraf en AgentConfiguration y establezca *runDsPrivileged* en true. Para obtener más información, consulte la "opciones de configuración del operador".

Problema:	Pruebe lo siguiente:

Mi pod de bits fluidos está fallando con los siguientes errores: [2024/10/16 14:16 23:23] [error] [/src/fluent-bit/plugins/in_tail/tail_fs_inotify.c:360 errno=24] Demasiados archivos abiertos [2024/16:2024:10/16 14] [error] fallo al inicializar la entrada tail,0 [16/23:10/16 14] [error] [error]	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 el bit fluido.
	Nota: Para que estos ajustes sean persistentes en todos los reinicios de nodos, debe poner 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>`

Puede encontrar información adicional en la "Soporte técnico" página o en el "Matriz de compatibilidad de recopilador de datos".

Opciones de configuración del operador de supervisión de Kubernetes

La "Operador de supervisión de Kubernetes" configuración se puede personalizar.

La siguiente tabla enumera las posibles opciones para el archivo *AgentConfiguration*:

Componente	Opción	Descripción
agente		Opciones de configuración comunes a todos los componentes que el operador puede instalar. Estas pueden considerarse como opciones “globales”.
	Repositorio de documentos	Una anulación de dockerRepo para extraer imágenes de repositorios de Docker privados de clientes en comparación con el repositorio de Docker de Data Infrastructure Insights. El valor predeterminado es el repositorio de datos de información sobre infraestructura de datos
	DockerImagePullSecret	Opcional: Un secreto para el repositorio privado de los clientes
	Nombre del clúster	Campo de texto libre que identifica de forma única un clúster en todos los clústeres de clientes. Debería ser único para un inquilino de Data Infrastructure Insights. El valor predeterminado es lo que introduce el cliente en la interfaz de usuario del campo «Cluster Name»
	Proxy Format: Proxy: Servidor: Puerto: Nombre de usuario: Contraseña: NoProxy: IsTelegrafProxyEnabled: IsAuProxyEnabled: IsFluentbitProxyEnabled: IsCollectorProxyEnabled:	Opcional para establecer el proxy. Este suele ser el proxy corporativo del cliente.
telegraf		Opciones de configuración que pueden personalizar la instalación de telegraf del Operador
	Intervalo de colección	Intervalo de recopilación de métricas, en segundos (Max=60s)
	DsCpuLimit	Límite de CPU para telegraf ds
	DsMemLimit	Límite de memoria para telegraf ds
	DsCpuRequest	Solicitud de CPU para telegraf ds
	DsMemRequest	Solicitud de memoria para telegraf ds
	RsCpuLimit	Límite de CPU para telegraf rs
	RsMemLimit	Límite de memoria para telegraf rs
	RsCpuRequest	Solicitud de CPU para telegraf rs
	RsMemRequest	Solicitud de memoria para telegraf rs
	Privilegios de ejecución	Ejecute el contenedor *telegraf-mountstats-poller* de telegraf DaemonSet en modo privilegiado. Establezca esta opción en true si SELinux está habilitado en sus nodos de Kubernetes.
	RunDsPrivileged	Establezca runDsPrivileged en true para ejecutar el contenedor telegraf DaemonSet en modo privilegiado.

Componente	Opción	Descripción
	Tamaño de lote	Consulte "Documentación de configuración de Telegraf"
	Buffer Limit	Consulte "Documentación de configuración de Telegraf"
	RoundInterval	Consulte "Documentación de configuración de Telegraf"
	Colección Jitter	Consulte "Documentación de configuración de Telegraf"
	precisión	Consulte "Documentación de configuración de Telegraf"
	FlushInterval	Consulte "Documentación de configuración de Telegraf"
	FlushJitter	Consulte "Documentación de configuración de Telegraf"
	Tiempo de espera de salida	Consulte "Documentación de configuración de Telegraf"
	DsToleraciones	telegraf-ds toleraciones adicionales.
	RsToleraciones	toleraciones adicionales de telegraf-rs.
	SkipProcessorsAfterAggregators	Consulte "Documentación de configuración de Telegraf"
	sin protección	Vea esto "Problema conocido de Telegraf". Si se establece *UNPROTECTED*, se indicará al operador de supervisión de Kubernetes que ejecute Telegraf con el `--unprotected` indicador.
métricas-estado-kube		Opciones de configuración que pueden personalizar la instalación de métricas de estado kube del Operador
	CpuLimit	Límite de CPU para la implementación de métricas de estado-kube
	MemLimit	Límite de MEM para el despliegue de métricas de estado-kube
	CpuRequest	Solicitud de CPU para el despliegue de métricas de estado de kube
	MemRequest	Solicitud de MEM para el despliegue de métricas de estado de kube
	recursos	lista de recursos separados por comas para capturar. ejemplo: cronjobs,daemonsets,deployments,ingresses,jobs,namespaces,nodes,persistentvolumeclaims, persistentvolumes,pods,replicasets,resourcequota,services,statefulsets
	toleraciones	kube-state-metrics toleraciones adicionales.

Componente	Opción	Descripción
	etiquetas	una lista separada por comas de los recursos que kube-state-metrics debe capturar + ejemplo: cronjobs=[**],daemonsets=[**],despliegues=[**],ingresses=[**],jobs=[**],namespaces=[**],nodes=[**], persistentvolumeclaims=[**],persistentvolumes=[**],replicads=[**
registros		Opciones de configuración que pueden personalizar la recopilación de registros y la instalación del operador
	ReadFromHead	verdadero/falso, debe leer con fluidez el log de la cabecera
	tiempo de espera	timeout, en segundos
	DnsMode	TCP/UDP, modo para DNS
	toleraciones de bits fluidas	toleraciones adicionales de fluent-bit-ds.
	toleraciones-exportador-de-eventos	toleraciones adicionales de evento-exportador.
	Event-exporter-maxEventAgeSeconds	antigüedad máxima de evento de exportador-evento. Consulte https://github.com/jkroepke/resmoio-kubernetes-event-exporter
asignación de carga de trabajo		Opciones de configuración que pueden personalizar la recopilación de mapas de carga de trabajo y la instalación del operador.
	CpuLimit	Límite de CPU para ds de observador neto
	MemLimit	límite de mem para ds de observador neto
	CpuRequest	Solicitud de CPU para ds de observador de red
	MemRequest	solicitud de mem para ds de observador neto
	MetricAggregationInterval	intervalo de agregación de métricas, en segundos
	BpfPollInterval	Intervalo de sondeo de BPF, en segundos
	Habilitar DNSLookup	True/false, active la búsqueda de DNS
	l4-toleraciones	net-observer-l4-ds toleraciones adicionales.
	Privilegios de ejecución	True/false - Establece runPrivileged en true si SELinux está habilitado en los nodos de Kubernetes.
gestión del cambio		Opciones de configuración para la administración y análisis de cambios de Kubernetes
	CpuLimit	Límite de CPU para change-observer-watch-rs
	MemLimit	Límite de MEM para change-observer-watch-rs
	CpuRequest	Solicitud de CPU para change-observer-watch-rs
	MemRequest	solicitud de mem para change-observer-watch-rs

Componente	Opción	Descripción
	FailureDeclarationIntervalMins	Intervalo en minutos tras el cual un despliegue incorrecto de una carga de trabajo se marcará como erróneo
	DeployAggrIntervalSeconds	Frecuencia a la que se envían los eventos de implementación de carga de trabajo en curso
	No WorkloadAggrIntervalSeconds	Frecuencia a la que se combinan y se envían las implementaciones sin cargas de trabajo
	TermsToRedact	Un conjunto de expresiones regulares utilizadas en los nombres env y los mapas de datos cuyo valor será redactado como ejemplo: “Pwd”, “password”, “token”, “apikey”, “api-key”, “jwt”
	KindsToWatch adicional	Una lista separada por comas de tipos adicionales para ver desde el conjunto predeterminado de tipos observados por el recopilador
	KindsToIgnoreFromWatch	Una lista separada por comas de tipos que ignorar de la observación del conjunto predeterminado de tipos observados por el recopilador
	LogRecordAggrIntervalSeconds	Frecuencia con la que los registros de registro se envían a CI desde el recopilador
	toleraciones de vigilancia	tolerancia adicional change-observer-watch-ds. Formato de línea única abreviado solamente. Ejemplo: '{key: taint1, operator: Exists, effect: NoSchedule},{key: taint2, operator: Exists, effect: Noexecute}'

Archivo de configuración de AgentConfiguration de ejemplo

A continuación se muestra un archivo *AgentConfiguration* de ejemplo.

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

spec:
  # # You can modify the following fields to configure the operator.
  # # Optional settings are commented out and include 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
clustername.
    # # clusterName must be unique across all clusters in your Data
Infrastructure Insights environment.
    clusterName: "my_cluster"

    # # Proxy settings. The proxy that the operator should use to send
metrics to Data Infrastructure Insights.
    # # Please see 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] By default, the operator uses the CI repository.
    # # To use a private repository, change this field to your repository
name.
    # # Please see 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] The name of the imagePullSecret for dockerRepo.
    # # If you are using a private repository, change this field from
'netapp-ci-docker' to the name of your secret.
    dockerImagePullSecret: 'netapp-ci-docker'

    # # Allow the operator to automatically rotate its ApiKey before
expiration.
    # tokenRotationEnabled: 'true'
    # # Number of days before expiration that the ApiKey should be
rotated. This must be less than the total ApiKey duration.
    # tokenRotationThresholdDays: '30'

  telegraf:
    # # Settings to fine-tune metrics data collection. Telegraf config
names are included in parenthesis.
    # # See
https://github.com/influxdata/telegraf/blob/master/docs/CONFIGURATION.md#a
```

```
gent

    # # The default time telegraf will wait between inputs for all plugins
(interval). Max=60
    # collectionInterval: '60s'
    # # Maximum number of records per output that telegraf will write in
one batch (metric_batch_size).
    # batchSize: '10000'
    # # Maximum number of records per output that telegraf will cache
pending a successful write (metric_buffer_limit).
    # bufferLimit: '150000'
    # # Collect metrics on multiples of interval (round_interval).
    # roundInterval: 'true'
    # # Each plugin waits a random amount of time between the scheduled
collection time and that time + collection_jitter before collecting inputs
(collection_jitter).
    # collectionJitter: '0s'
    # # Collected metrics are rounded to the precision specified. When set
to "0s" precision will be set by the units specified by interval
(precision).
    # precision: '0s'
    # # Time telegraf will wait between writing outputs (flush_interval).
Max=collectionInterval
    # flushInterval: '60s'
    # # Each output waits a random amount of time between the scheduled
write time and that time + flush_jitter before writing outputs
(flush_jitter).
    # flushJitter: '0s'
    # # Timeout for writing to outputs (timeout).
    # outputTimeout: '5s'

    # # telegraf-ds CPU/Mem limits and requests.
    # # See https://kubernetes.io/docs/concepts/configuration/manage-
resources-containers/
    # dsCpuLimit: '750m'
    # dsMemLimit: '800Mi'
    # dsCpuRequest: '100m'
    # dsMemRequest: '500Mi'

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

    # # Skip second run of processors after aggregators
```

```
    # skipProcessorsAfterAggregators: 'true'

    # # telegraf additional tolerations. Use the following abbreviated
single line format only.
    # # Inspect telegraf-rs/-ds to view tolerations which are always
present.
    # # Example: '{key: taint1, operator: Exists, effect:
NoSchedule},{key: taint2, operator: Exists, effect: NoExecute}'
    # dsTolerations: ''
    # rsTolerations: ''


    # If telegraf warns of insufficient lockable memory, try increasing
the limit of lockable memory for Telegraf in the underlying operating
system/node.  If increasing the limit is not an option, set this to true
to instruct Telegraf to not attempt to reserve locked memory pages.  While
this might pose a security risk as decrypted secrets might be swapped out
to disk, it allows for execution in environments where reserving locked
memory is not possible.
    # unprotected: 'false'

    # # Run the telegraf DaemonSet's telegraf-mountstats-poller container
in privileged mode.  Set runPrivileged to true if SELinux is enabled on
your Kubernetes nodes.
    # runPrivileged: '{{
.Values.telegraf_installer.kubernetes.privileged_mode }}'

    # # Set runDsPrivileged to true to run the telegraf DaemonSet's
telegraf container in privileged mode
    # runDsPrivileged: '{{
.Values.telegraf_installer.kubernetes.ds.privileged_mode }}'

    # # Collect container Block IO metrics.
    # dsBlockIOEnabled: 'true'

    # # Collect NFS IO metrics.
    # dsNfsIOEnabled: 'true'

    # # Collect kubernetes.system_container metrics and objects in the
kube-system|cattle-system namespaces for managed kubernetes clusters (EKS,
AKS, GKE, managed Rancher).  Set this to true if you want collect these
metrics.
    # managedK8sSystemMetricCollectionEnabled: 'false'

    # # Collect kubernetes.pod_volume (pod ephemeral storage) metrics.
Set this to true if you want to collect these metrics.
```

```
    # podVolumeMetricCollectionEnabled: 'false'

    # # Declare Rancher cluster as managed.  Set this to true if your
Rancher cluster is managed as opposed to on-premise.
    # isManagedRancher: 'false'

    # # If telegraf-rs fails to start due to being unable to find the etcd
crt and key, manually specify the appropriate path here.
    # rsHostEtcdCrt: ''
    # rsHostEtcdKey: ''

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

    # # Comma-separated list of resources to enable.
    # # See resources in https://github.com/kubernetes/kube-state-
metrics/blob/main/docs/cli-arguments.md
    # resources:
'cronjobs,daemonsets,deployments,ingresses,jobs,namespaces,nodes,persisten
tvolumeclaims,persistentvolumes,pods,replicasets,resourcequotas,services,s
tatefulsets'

    # # Comma-separated list of metrics to enable.
    # # See metric-allowlist in https://github.com/kubernetes/kube-state-
metrics/blob/main/docs/cli-arguments.md
    # 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'

    # # Comma-separated list of Kubernetes label keys that will be used in
the resources' labels metric.
    # # See metric-labels-allowlist in https://github.com/kubernetes/kube-
state-metrics/blob/main/docs/cli-arguments.md
    # labels:
'cronjobs=[*],daemonsets=[*],deployments=[*],ingresses=[*],jobs=[*],namesp
aces=[*],nodes=[*],persistentvolumeclaims=[*],persistentvolumes=[*],pods=[
*],replicasets=[*],resourcequotas=[*],services=[*],statefulsets=[*]'

    # # kube-state-metrics additional tolerations. Use the following
abbreviated single line format only.
    # # No tolerations are applied by default
    # # Example: '{key: taint1, operator: Exists, effect:
NoSchedule},{key: taint2, operator: Exists, effect: NoExecute}'
    # tolerations: ''

    # # kube-state-metrics shards.  Increase the number of shards for
larger clusters if telegraf RS pod(s) experience collection timeouts
    # shards: '2'

  # # Settings for the Events Log feature.
  # logs:
    # # Set runPrivileged to true if Fluent Bit fails to start, trying to
open/create its database.
    # runPrivileged: 'false'

    # # If Fluent Bit should read new files from the head, not tail.
    # # See Read_from_Head in
https://docs.fluentbit.io/manual/pipeline/inputs/tail
    # readFromHead: "true"

    # # Network protocol that Fluent Bit should use for DNS: "UDP" or
"TCP".
    # dnsMode: "UDP"

    # # DNS resolver that Fluent Bit should use: "LEGACY" or "ASYNC"
    # fluentBitDNSResolver: "LEGACY"

    # # Logs additional tolerations. Use the following abbreviated single
line format only.
    # # Inspect fluent-bit-ds to view tolerations which are always
present. No tolerations are applied by default for event-exporter.
    # # Example: '{key: taint1, operator: Exists, effect:
NoSchedule},{key: taint2, operator: Exists, effect: NoExecute}'
```

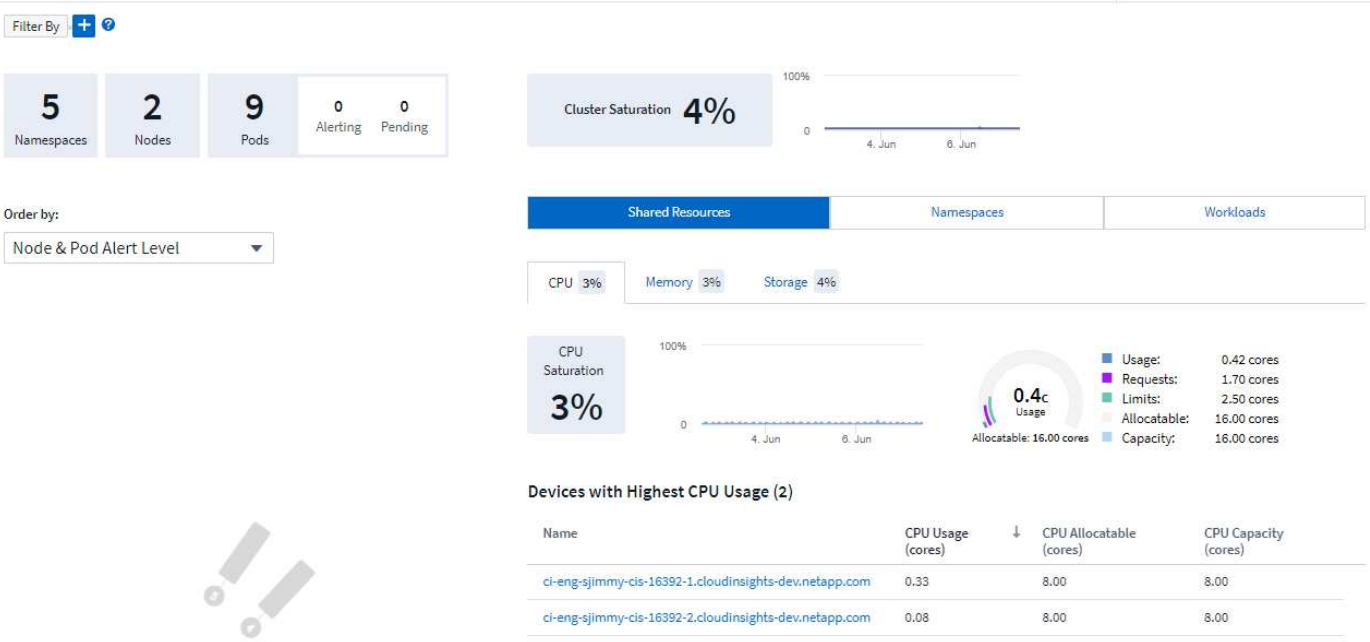
```
    # fluent-bit-tolerations: ''
    # event-exporter-tolerations: ''

    # # event-exporter CPU/Mem limits and requests.
    # # See https://kubernetes.io/docs/concepts/configuration/manage-
resources-containers/
    # event-exporter-cpuLimit: '500m'
    # event-exporter-memLimit: '1Gi'
    # event-exporter-cpuRequest: '50m'
    # event-exporter-memRequest: '100Mi'

    # # event-exporter max event age.
    # # See https://github.com/jkroepke/resmoio-kubernetes-event-exporter
    # event-exporter-maxEventAgeSeconds: '10'

    # # event-exporter client-side throttling
    # # Set kubeBurst to roughly match your events per minute and
kubeQPS=kubeBurst/5
    # # See https://github.com/resmoio/kubernetes-event-
exporter#troubleshoot-events-discarded-warning
    # event-exporter-kubeQPS: 20
    # event-exporter-kubeBurst: 100

    # # fluent-bit CPU/Mem limits and requests.
    # # See https://kubernetes.io/docs/concepts/configuration/manage-
resources-containers/
    # fluent-bit-cpuLimit: '500m'
    # fluent-bit-memLimit: '1Gi'
    # fluent-bit-cpuRequest: '50m'
    # fluent-bit-memRequest: '100Mi'

  # # Settings for the Network Performance and Map feature.
  # workload-map:
    # # netapp-ci-net-observer-l4-ds CPU/Mem limits and requests.
    # # See https://kubernetes.io/docs/concepts/configuration/manage-
resources-containers/
    # cpuLimit: '500m'
    # memLimit: '500Mi'
    # cpuRequest: '100m'
    # memRequest: '500Mi'

    # # Metric aggregation interval in seconds. Min=30, Max=120
    # metricAggregationInterval: '60'

    # # Interval for bpf polling. Min=3, Max=15
    # bpfPollInterval: '8'
```

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

    # # netapp-ci-net-observer-l4-ds additional tolerations. Use the
following abbreviated single line format only.
    # # Inspect netapp-ci-net-observer-l4-ds to view tolerations which are
always present.
    # # Example: '{key: taint1, operator: Exists, effect:
NoSchedule},{key: taint2, operator: Exists, effect: NoExecute}'
    # l4-tolerations: ''

    # # Set runPrivileged to true if SELinux is enabled on your Kubernetes
nodes.
    # # Note: In OpenShift environments, this is set to true
automatically.
    # runPrivileged: 'false'

  # change-management:
    # # change-observer-watch-rs CPU/Mem limits and requests.
    # # See https://kubernetes.io/docs/concepts/configuration/manage-
resources-containers/
    # cpuLimit: '1'
    # memLimit: '1Gi'
    # cpuRequest: '500m'
    # memRequest: '500Mi'

    # # Interval in minutes after which a non-successful deployment of a
workload will be marked as failed
    # failureDeclarationIntervalMins: '30'

    # # Frequency at which workload deployment in-progress events are sent
    # deployAggrIntervalSeconds: '300'

    # # Frequency at which non-workload deployments are combined and sent
    # nonWorkloadAggrIntervalSeconds: '15'

    # # A 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"'

    # # A comma separated list of additional kinds to watch from the
default set of kinds watched by the collector
    # # Each kind will have to be prefixed by its apigroup
```

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

    # # A 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: ''

    # # 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: ''

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

    # # change-observer-watch-ds additional tolerations. Use the following
abbreviated single line format only.
    # # Inspect change-observer-watch-ds to view tolerations which are
always present.
    # # 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 POD

Los números en la parte superior de la página muestran el número total de espacios de nombres, nodos y pods en el clúster, así como el número de pods que actualmente están alertando 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 y un gráfico que muestra la tendencia reciente a lo largo del tiempo. La saturación del clúster es la mayor saturación de CPU, memoria o almacenamiento en cada momento.

A continuación, la página muestra por defecto **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 adicionales de uso. Para el almacenamiento, el valor mostrado es el mayor de saturación de back-end y filesystem, que se calculan de forma independiente.

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

Espacios de nombres

En la pestaña Namespaces, se muestra una lista de todos los espacios de nombres en el entorno Kubernetes, donde se muestra el uso de la CPU y la memoria, así como el recuento de cargas de trabajo en cada espacio de nombres. Haga clic en los vínculos 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 igual modo, la pestaña cargas de trabajo muestra una lista de las cargas de trabajo de cada espacio de nombres, mostrando de nuevo el uso de la CPU y la memoria. Al hacer clic en el espacio de nombres, se vinculan los simulacros de cada uno.

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

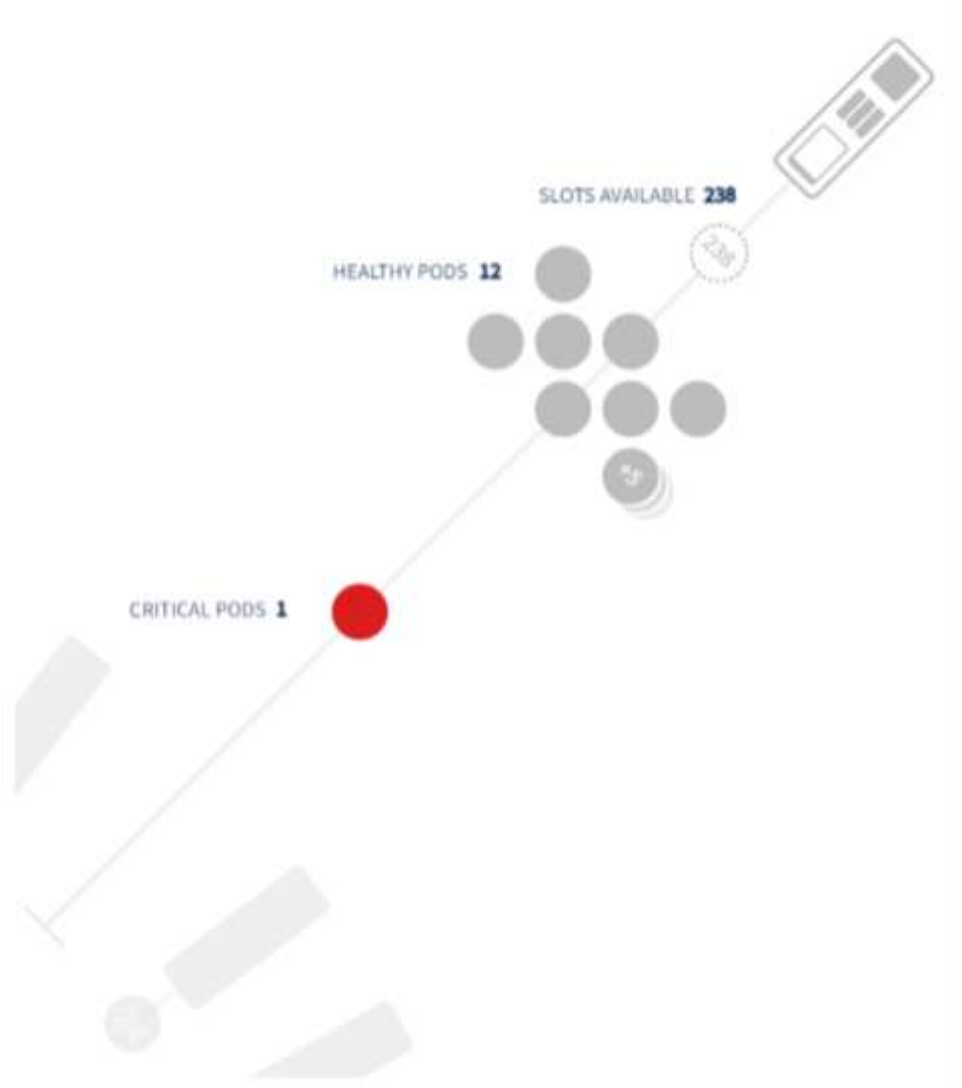
La "rueda" del clúster

La sección "rueda" de clúster proporciona el estado de nodo y cápsula de un vistazo, en el que puede 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 con los botones disponibles.

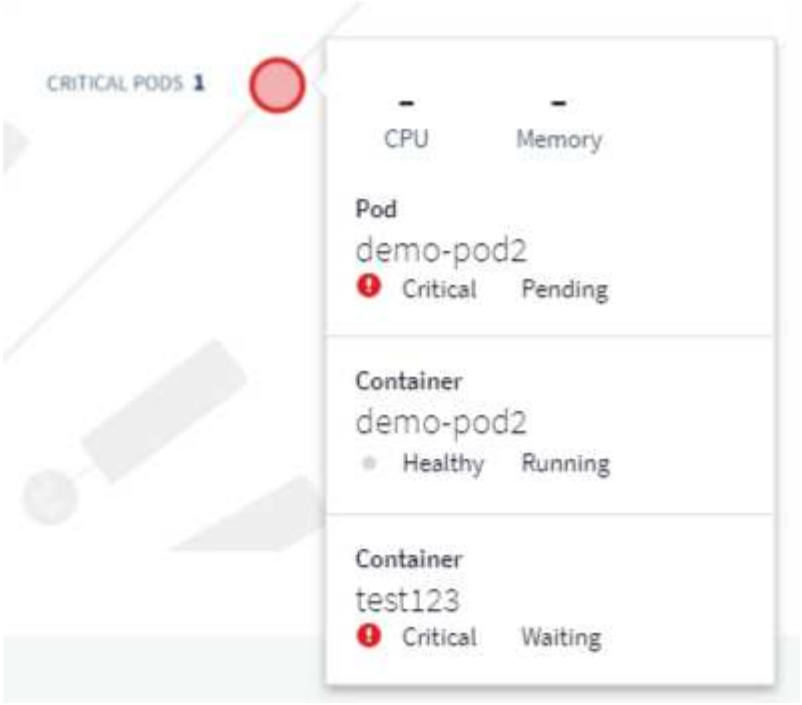
Los pods o nodos de alertas se muestran en rojo. Las áreas de "advertencia" se muestran en color naranja. Las vainas no programadas (es decir, sin conectar) se mostrarán en la esquina inferior de la "rueda" del grupo de instrumentos.

Si pasa el ratón por un pod (círculo) o Node (barra), se ampliará la vista del nodo.

Al hacer clic en el pod o en el nodo de esa vista se amplía a la vista nodo expandida.



Desde aquí, puede pasar el ratón sobre un elemento para ver detalles sobre ese elemento. Por ejemplo, al pasar por el pod crucial en este ejemplo, se muestran detalles sobre ese pod.



Puede ver la información del sistema de archivos, la memoria y la CPU pasando por encima de los elementos Node.

Una nota sobre los indicadores

Los indicadores de memoria y CPU muestran tres colores, ya que muestran *used* en relación con *Allocatable Capacity* y *total Capacity*.

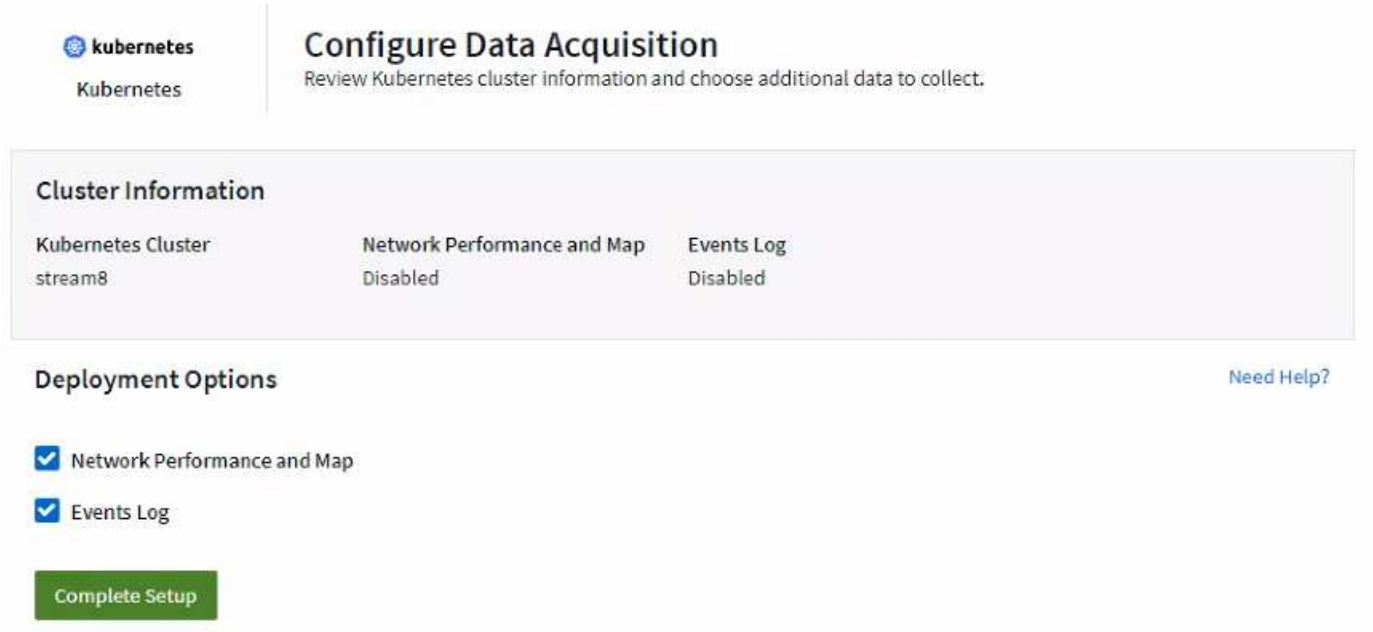
Supervisión y mapa del rendimiento de la red de Kubernetes

La función de supervisión y mapa del rendimiento de red de Kubernetes simplifica la solución de problemas mediante la asignación de dependencias entre servicios (también llamadas cargas de trabajo) y proporciona 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 funcionalidad ayuda a las organizaciones a reducir los costes generales mediante el análisis y la auditoría de los flujos de tráfico de Kubernetes.

Características principales: • El mapa de carga de trabajo presenta los flujos y dependencias de las cargas de trabajo de Kubernetes y destaca los problemas de red y de rendimiento. • Supervisar el tráfico de red entre los pods de Kubernetes, las cargas de trabajo y los nodos; identifica la fuente del tráfico y los problemas de latencia. • Reduzca los costes generales analizando el tráfico de red entre zonas, entre regiones y entre zonas.

Requisitos previos

Antes de poder usar el control y el mapa de rendimiento de red de Kubernetes, debe haber configurado el "Operador de supervisión Kubernetes de NetApp" para habilitar esta opción. Durante el despliegue del operador, seleccione la casilla de verificación Rendimiento de red y mapa para activarlo. También puedes habilitar esta opción navegando a una página de destino de Kubernetes y seleccionando «Modificar implementación».



Supervisiones

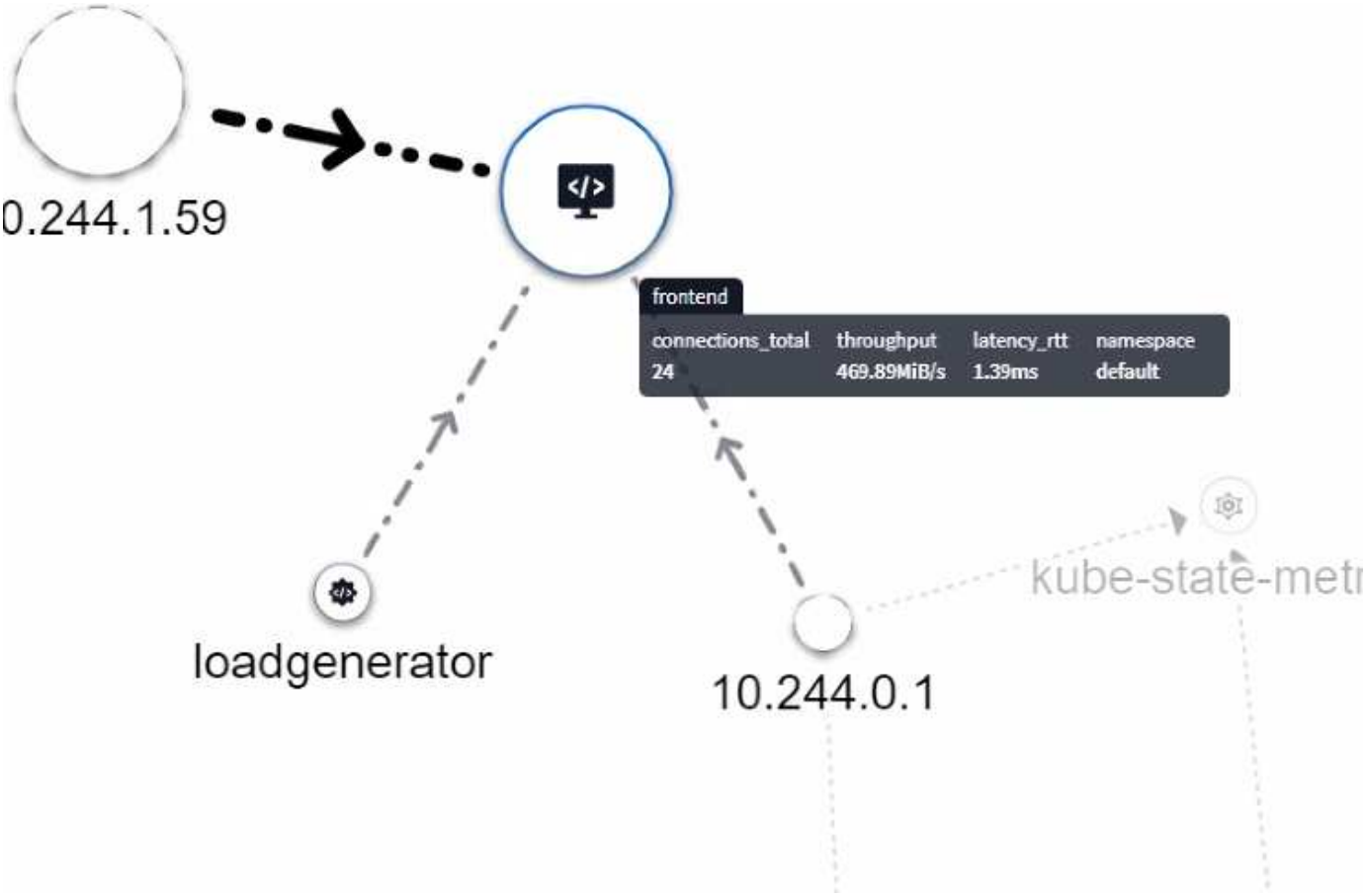
La asignación de carga de trabajo utiliza "supervisiones" para derivar información. Data Infrastructure Insights proporciona una serie de supervisores de Kubernetes predeterminados (tenga en cuenta que estos pueden estar *Paused* de forma predeterminada. Puede *Reanudar* (es decir, habilitar) los monitores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

Puedes crear alertas de métricas de Data Infrastructure Insights en cualquiera de los tipos de objeto siguientes. Asegúrese de que los datos están agrupados por el tipo de objeto predeterminado.

- kubernetes
- inicio de kubernetes
- kubernetes.deployment
- kubernetes.cronjob
- kubernetes.job
- kubernetes.replicaset
- kubernetes.statefulset
- kubernetes.pod
- kubernetes.network_traffic_l4

El mapa

El mapa muestra los servicios/cargas de trabajo y sus relaciones entre sí. Las flechas muestran las direcciones del tráfico. El pasar por encima de una carga de trabajo muestra información resumida de esa carga de trabajo, como puede observarse en este ejemplo:



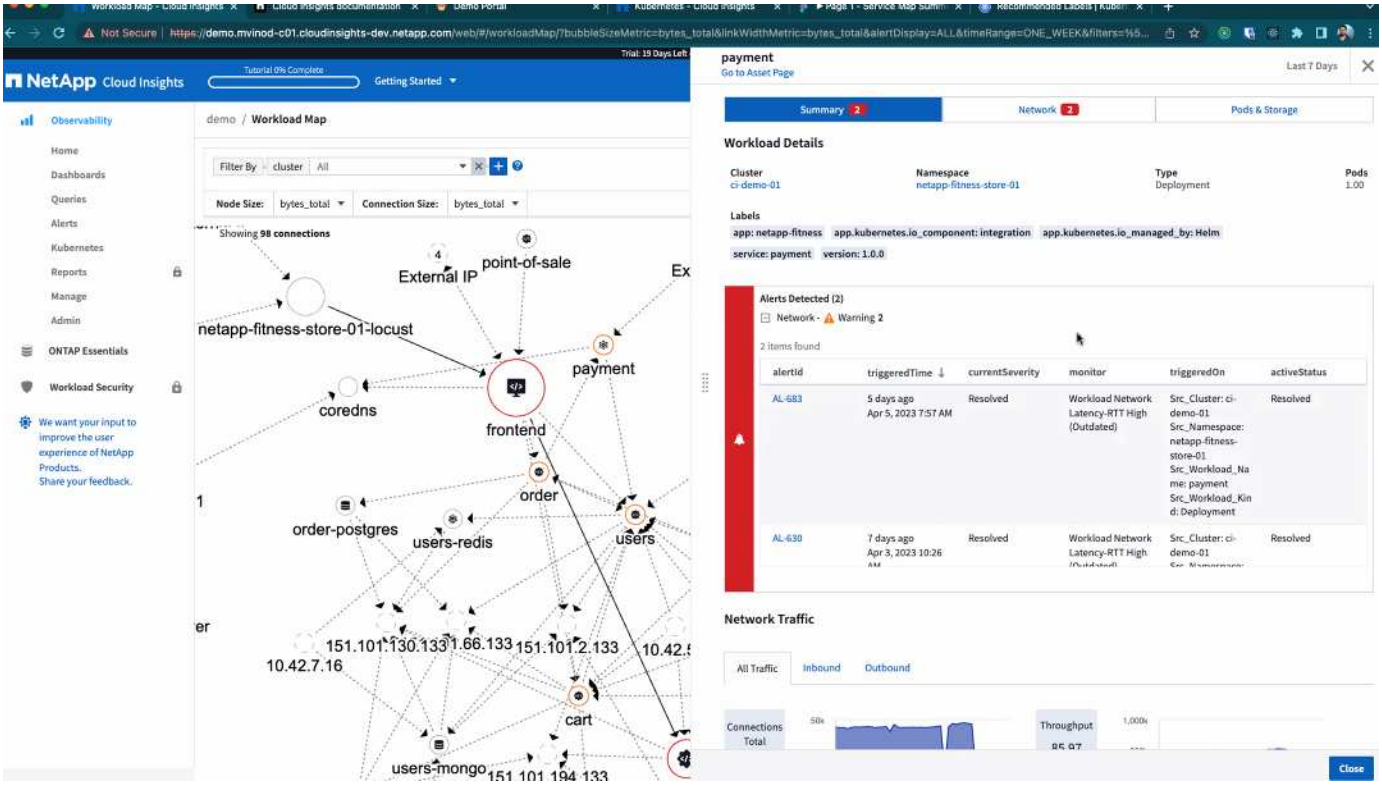
Los iconos dentro de los círculos representan diferentes tipos de servicio. Tenga en cuenta que los iconos sólo son visibles si los objetos subyacentes tienen etiquetas.

El tamaño de cada círculo indica el tamaño del nodo. Tenga en cuenta que estos tamaños son relativos, el nivel de zoom del navegador o el tamaño de la pantalla pueden afectar el tamaño real del círculo. De la misma manera, el estilo de la línea de tráfico le ofrece una vista rápida del tamaño de la conexión; las líneas sólidas en negrita son de alto tráfico, mientras que las líneas con puntos claros son de menor tráfico.

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

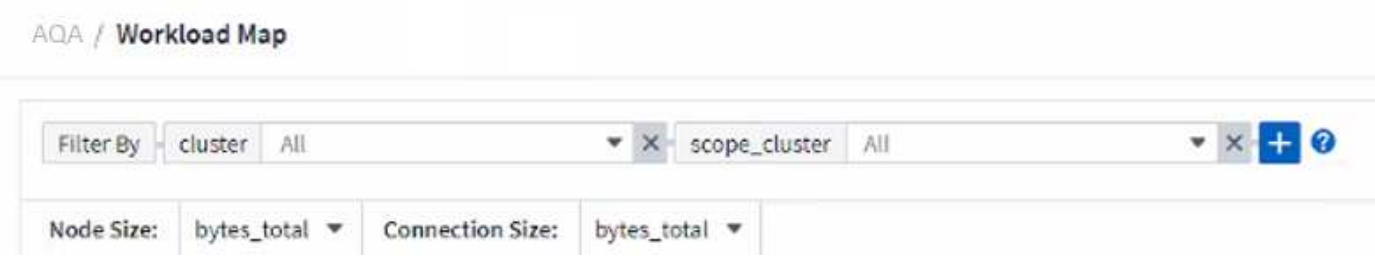
Detalles de carga de trabajo y alertas

Los círculos mostrados 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 deslizante con más detalles.

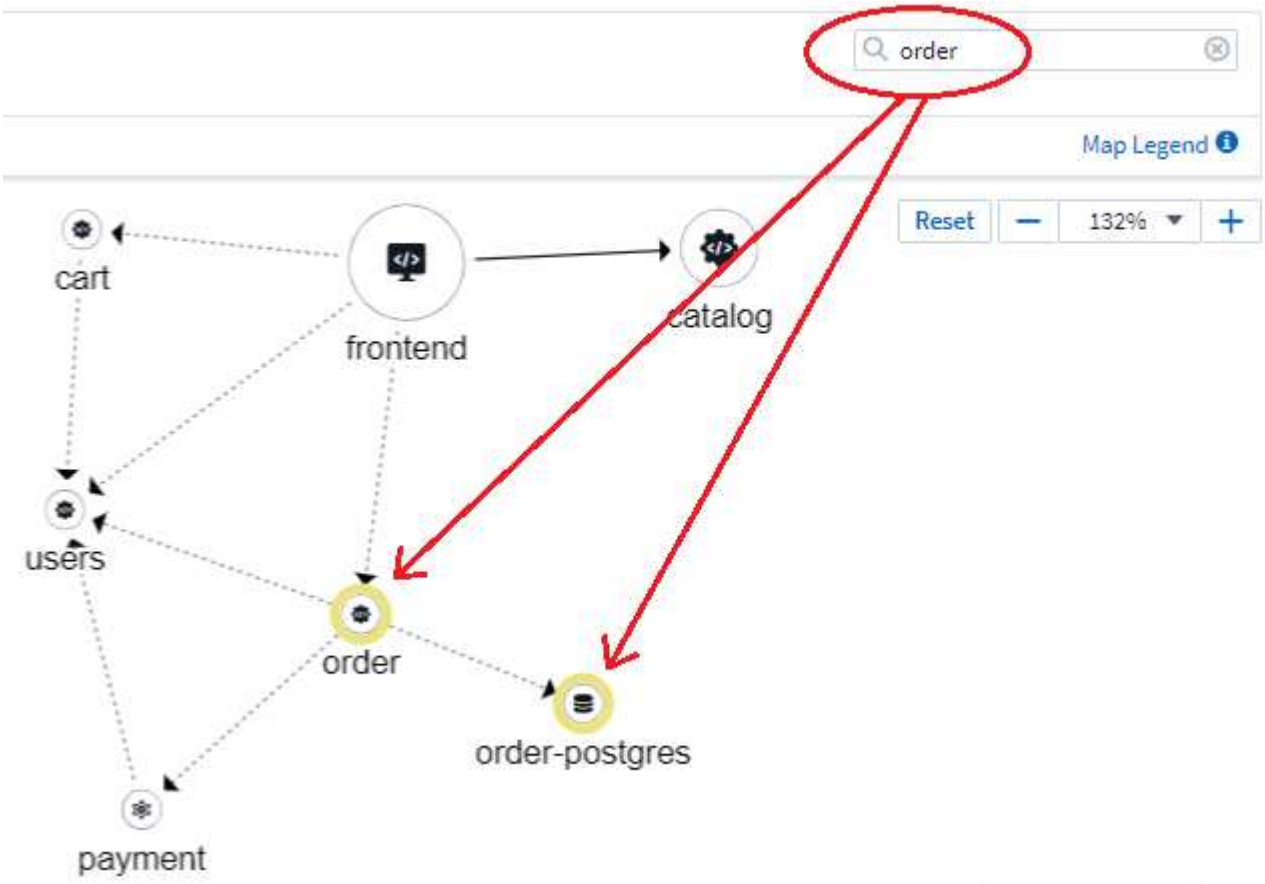


Búsqueda y filtrado

Como sucede 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 *Find* 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 iconos de círculo). 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 deriva de la etiqueta de la carga de trabajo del pod
  - Se esperaba una etiqueta en la carga de trabajo: App.kubernetes.io/component
  - Referencia de nombre de etiqueta: https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels/
  - Etiquetas recomendadas:
    - front-end

- backend
- base de datos
- caché
- cola
- kafka

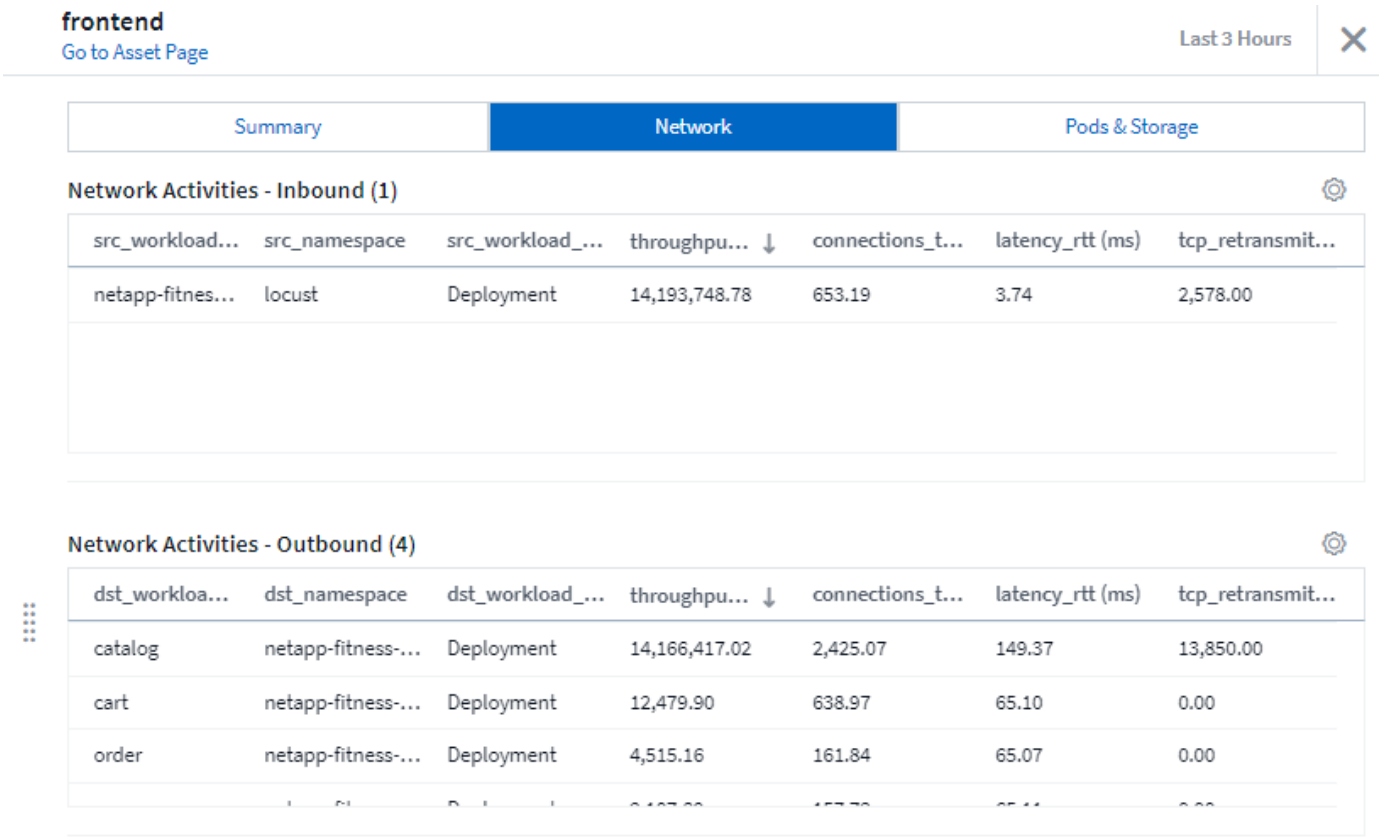
- Si el origen es externo al clúster de kubernetes:
  - Data Infrastructure Insights intentará analizar el nombre resuelto DNS para extraer el tipo de servicio.

    Por ejemplo, con un nombre DNS resuelto de *s3.eu-north-1.amazonaws.com*, el nombre resuelto se analiza para obtener *S3* como tipo de servicio.

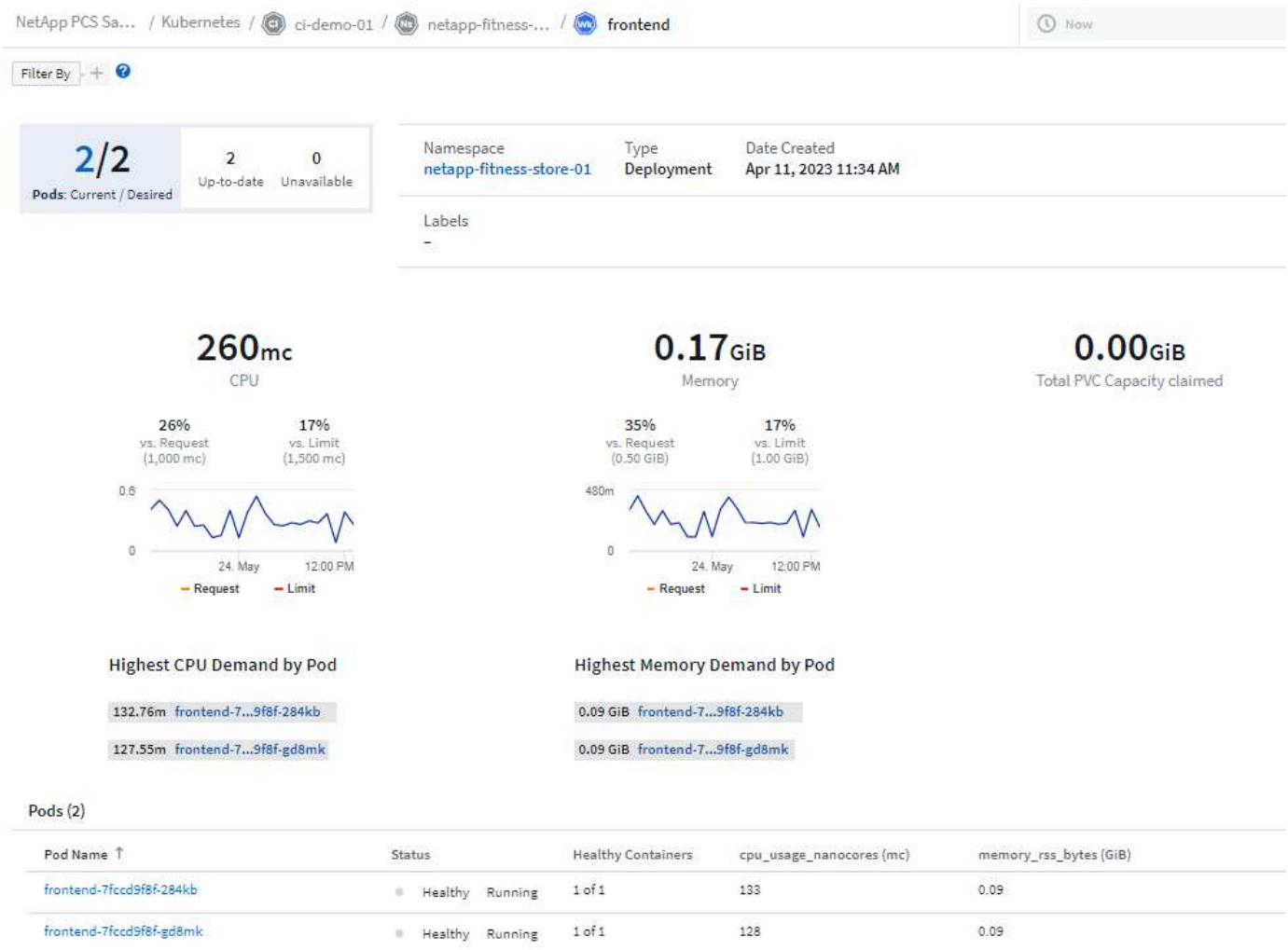
Vea lo profundo

Al hacer clic con el botón derecho en una carga de trabajo, encontrará opciones adicionales para explorar más a fondo. Por ejemplo, desde aquí puede aplicar el zoom para ver las conexiones de esa carga de trabajo.

O bien, puede abrir el panel desplegable de detalles para ver directamente las pestañas *Summary*, *Network* o *Pod & Storage*.



Por último, al seleccionar *Ir a la página de activos* se abrirá la página de destino detallada del activo para la carga de trabajo.



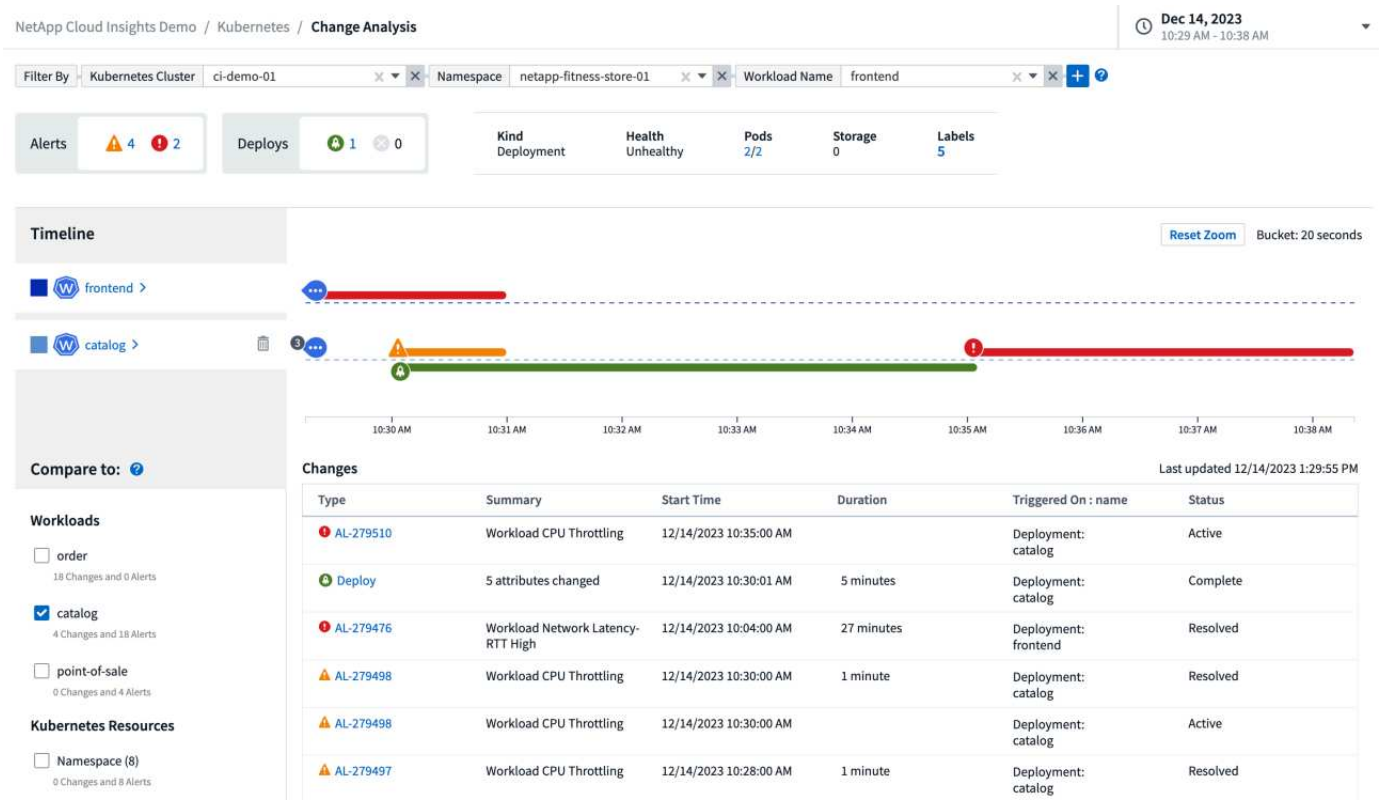
Análisis de cambios de Kubernetes

Kubernetes Change Analytics le proporciona una vista integral de los cambios recientes en su entorno K8s. Tiene a su alcance las alertas y el estado de la implementación. 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 K8s.

¿Cómo ayuda el análisis de cambios?

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

Para ver el análisis de cambios de Kubernetes, vaya a **Kubernetes > Análisis de cambios**.



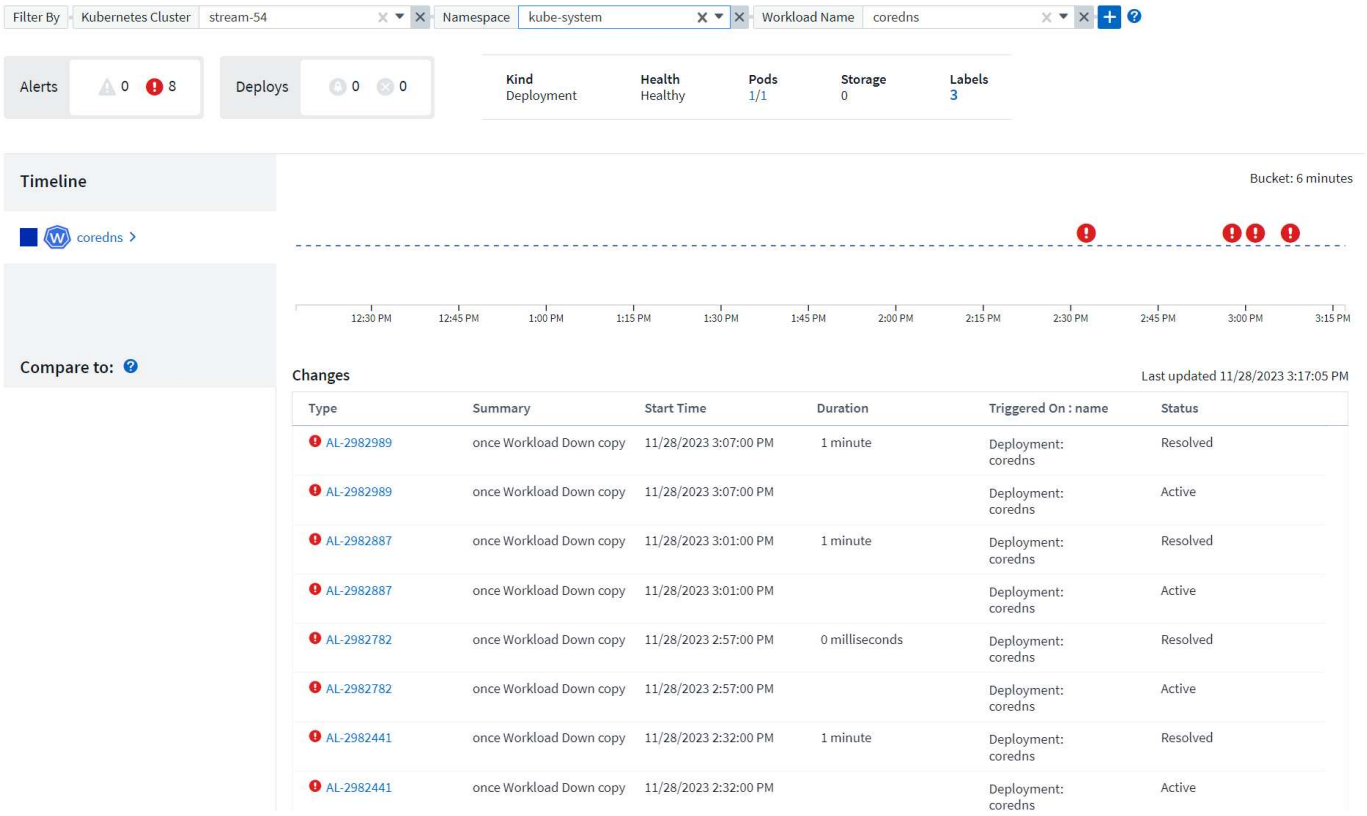
.

La página se actualiza automáticamente en función del intervalo de tiempo seleccionado en ese momento. Los intervalos de tiempo más pequeños significan que la pantalla se refresca con más frecuencia.

Filtrado

Como sucede con todas las funciones de Data Infrastructure Insights, filtrar la lista de cambios es intuitivo: En la parte superior de la página, introduzca o seleccione valores para su clúster de Kubernetes, espacio de nombres o carga de trabajo, o añada sus propios filtros seleccionando el botón {+].

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

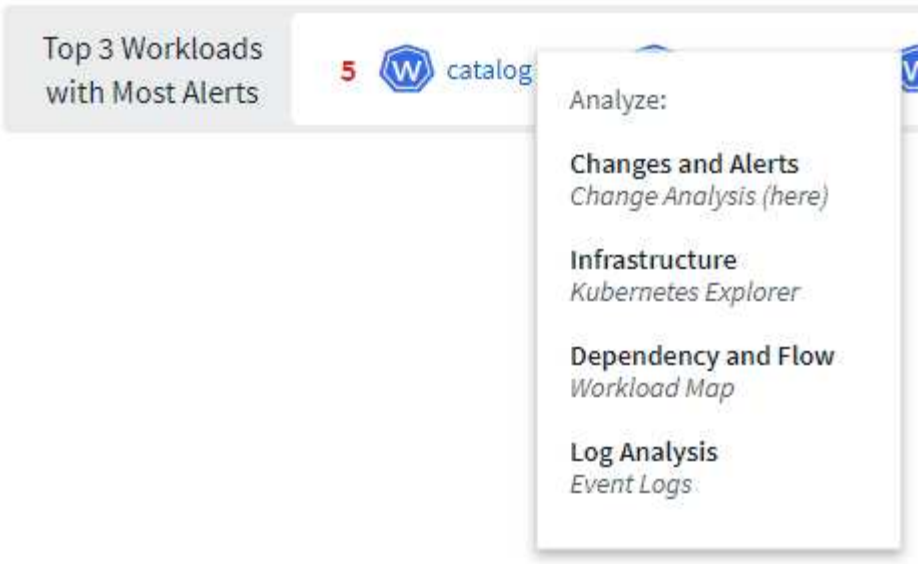


Estado rápido

Debajo del área de filtrado hay una serie de indicadores de alto nivel. A la izquierda aparece el número de alertas (Advertencia y Críticas). Este número incluye *Active* así como *Resolved* alertas. Para ver solo las alertas *Active*, establezca un filtro para “Estado” y seleccione “Activo”.

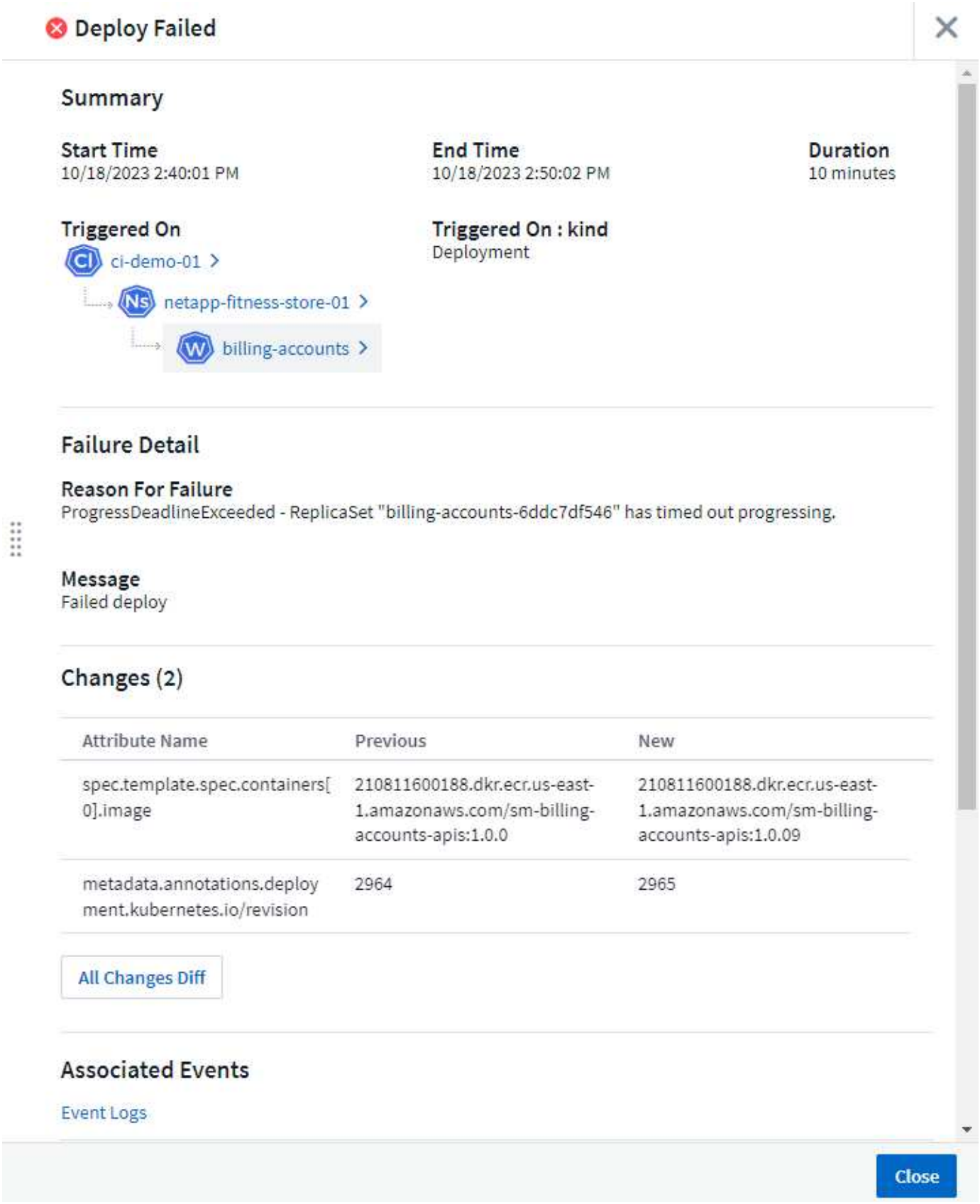
El estado de despliegue también se muestra aquí. De nuevo, el valor por defecto es mostrar el recuento de despliegues *Started*, *Complete* y *Failed*. Para ver solo despliegues *FAILED*, establezca un filtro para “Estado” y seleccione “Fallo”.

Las 3 principales cargas de trabajo con la mayor cantidad de alertas son las siguientes. El número de rojo que aparece 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 a través de 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 un despliegue fallido, se muestra un resumen del despliegue, con horas de inicio y finalización, duración y dónde se activó el despliegue, con enlaces para explorar esos recursos. También muestra el motivo del fallo, los cambios relacionados y los eventos asociados.



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