# Kubernetes

## Data Infrastructure Insights

NetApp
February 10, 2026

# Sommaire

# Kubernetes

## Présentation du cluster Kubernetes

Data Infrastructure Insights Kubernetes Explorer est un outil puissant permettant d'afficher l'état général et l'utilisation de vos clusters Kubernetes et vous permet d'explorer facilement les domaines d'investigation.

Cliquer sur **Tableaux de bord > Kubernetes Explorer** ouvre la page de liste des clusters Kubernetes. Cette page de présentation contient un tableau des clusters Kubernetes sur votre locataire.

[Page de liste Kubernetes]

**Liste des clusters**

La liste des clusters affiche les informations suivantes pour chaque cluster de votre locataire :

- Cluster **Nom**. Cliquer sur le nom d'un cluster ouvrira le"**page de détail**" pour ce cluster.
- Pourcentages de **saturation**. La saturation globale est la plus élevée des saturations du processeur, de la mémoire ou du stockage.
- Nombre de **nœuds** dans le cluster. Cliquer sur ce numéro ouvrira la page de la liste des nœuds.
- Nombre de **Pods** dans le cluster. Cliquer sur ce numéro ouvrira la page de la liste des pods.
- Nombre d'**espaces de noms** dans le cluster. Cliquer sur ce numéro ouvrira la page de liste des espaces de noms.
- Nombre de **charges de travail** dans le cluster. Cliquer sur ce numéro ouvrira la page de liste de charge de travail.

### Affiner le filtre

Lorsque vous filtrez, lorsque vous commencez à taper, vous avez la possibilité de créer un **filtre générique** basé sur le texte actuel. La sélection de cette option renverra tous les résultats correspondant à l'expression générique. Vous pouvez également créer des **expressions** en utilisant NOT ou AND, ou vous pouvez sélectionner l'option « Aucun » pour filtrer les valeurs nulles dans le champ.

[Filtrage avec caractère générique dans K8S Explorer]

Les filtres basés sur des caractères génériques ou des expressions (par exemple, NOT, AND, « Aucun », etc.) s'affichent en bleu foncé dans le champ de filtre. Les éléments que vous sélectionnez directement dans la liste sont affichés en bleu clair.

[Filtre affichant les éléments génériques et sélectionnés]

Les filtres Kubernetes sont contextuels, ce qui signifie par exemple que si vous êtes sur une page de nœud spécifique, le filtre pod_name répertorie uniquement les pods liés à ce nœud. De plus, si vous appliquez un filtre pour un espace de noms spécifique, le filtre pod_name répertoriera uniquement les pods sur ce nœud *et* dans cet espace de noms.

Notez que le filtrage par caractères génériques et par expressions fonctionne avec du texte ou des listes, mais pas avec des nombres, des dates ou des booléens.

# Avant d’installer ou de mettre à niveau l’opérateur de surveillance NetApp Kubernetes

Lisez ces informations avant d’installer ou de mettre à niveau le"Opérateur de surveillance Kubernetes" .

| Composant | Exigence |
|---|---|
| Version de Kubernetes | Kubernetes v1.20 et supérieur. |
| Distributions Kubernetes | AWS Elastic Kubernetes Service (EKS) Azure Kubernetes Service (AKS) Google Kubernetes Engine (GKE) Red Hat OpenShift Rancher Kubernetes Engine (RKE) VMware Tanzu |
| Système d’exploitation Linux | Data Infrastructure Insights ne prend pas en charge les nœuds exécutés avec l’architecture Arm64. Surveillance du réseau : doit exécuter le noyau Linux version 4.18.0 ou supérieure. Photon OS n’est pas pris en charge. |
| Étiquettes | Data Infrastructure Insights prend en charge la surveillance des nœuds Kubernetes qui exécutent Linux, en spécifiant un sélecteur de nœud Kubernetes qui recherche les étiquettes Kubernetes suivantes sur ces plates-formes : Kubernetes v1.20 et versions ultérieures : Kubernetes.io/os = linux Rancher + cattle.io comme plate-forme d’orchestration/Kubernetes : cattle.io/os = linux |
| Commandes | Les commandes curl et kubectl doivent être disponibles. Pour de meilleurs résultats, ajoutez ces commandes au PATH. |
| Connectivité | kubectl cli est configuré pour communiquer avec le cluster K8s cible et disposer d’une connectivité Internet à votre environnement Data Infrastructure Insights . Si vous êtes derrière un proxy lors de l’installation, suivez les instructions du"Configuration de la prise en charge du proxy" section de l’installation de l’opérateur. Pour un audit et des rapports de données précis, synchronisez l’heure sur la machine Agent à l’aide du protocole NTP (Network Time Protocol) ou du protocole SNTP (Simple Network Time Protocol). |
| Autre | Si vous utilisez OpenShift 4.6 ou supérieur, vous devez suivre les"Instructions OpenShift" en plus de veiller à ce que ces conditions préalables soient respectées. |
| Jeton API | Si vous redéployez l’opérateur (c’est-à-dire que vous le mettez à jour ou le remplacez), il n’est pas nécessaire de créer un nouveau jeton API ; vous pouvez réutiliser le jeton précédent. |

## Choses importantes à noter avant de commencer

Si vous courez avec unprocuration , avoir unréférentiel personnalisé , ou utilisentOpenShift , lisez attentivement les sections suivantes.

Lire aussi à propos deAutorisations .

### Configuration de la prise en charge du proxy

Il existe deux endroits où vous pouvez utiliser un proxy sur votre locataire afin d'installer NetApp Kubernetes Monitoring Operator. Il peut s'agir des mêmes systèmes proxy ou de systèmes proxy distincts :

- Proxy nécessaire lors de l'exécution de l'extrait de code d'installation (à l'aide de « curl ») pour connecter le système où l'extrait est exécuté à votre environnement Data Infrastructure Insights
- Proxy requis par le cluster Kubernetes cible pour communiquer avec votre environnement Data Infrastructure Insights

Si vous utilisez un proxy pour l'un ou les deux, pour installer NetApp Kubernetes Operating Monitor, vous devez d'abord vous assurer que votre proxy est configuré pour permettre une bonne communication avec votre environnement Data Infrastructure Insights . Par exemple, à partir des serveurs/VM à partir desquels vous souhaitez installer l'opérateur, vous devez pouvoir accéder à Data Infrastructure Insights et pouvoir télécharger des binaires depuis Data Infrastructure Insights.

Pour le proxy utilisé pour installer NetApp Kubernetes Operating Monitor, avant d'installer l'opérateur, définissez les variables d'environnement *http_proxy/https_proxy*. Pour certains environnements proxy, vous devrez peut-être également définir la variable d'environnement *no_proxy*.

Pour définir la ou les variables, effectuez les étapes suivantes sur votre système **avant** d'installer NetApp Kubernetes Monitoring Operator :

1. Définissez les variables d'environnement *https_proxy* et/ou *http_proxy* pour l'utilisateur actuel :
   a. Si le proxy en cours de configuration ne dispose pas d'authentification (nom d'utilisateur/mot de passe), exécutez la commande suivante :

```
 export https_proxy=<proxy_server>:<proxy_port>
.. Si le proxy en cours de configuration dispose d'une
authentification (nom d'utilisateur/mot de passe), exécutez cette
commande :
```

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

Pour que le proxy utilisé pour votre cluster Kubernetes communique avec votre environnement Data Infrastructure Insights , installez NetApp Kubernetes Monitoring Operator après avoir lu toutes ces instructions.

Configurez la section proxy d'AgentConfiguration dans operator-config.yaml avant de déployer l'opérateur de surveillance NetApp 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
  ...
...
```

**Utiliser un référentiel Docker personnalisé ou privé**

Par défaut, l'opérateur de surveillance NetApp Kubernetes extrait les images de conteneur du référentiel Data Infrastructure Insights . Si vous disposez d'un cluster Kubernetes utilisé comme cible pour la surveillance et que ce cluster est configuré pour extraire uniquement des images de conteneur à partir d'un référentiel Docker personnalisé ou privé ou d'un registre de conteneurs, vous devez configurer l'accès aux conteneurs nécessaires à l'opérateur de surveillance NetApp Kubernetes.

Exécutez « Image Pull Snippet » à partir de la mosaïque d'installation de NetApp Monitoring Operator. Cette commande se connectera au référentiel Data Infrastructure Insights , extraira toutes les dépendances d'image pour l'opérateur et se déconnectera du référentiel Data Infrastructure Insights . Lorsque vous y êtes invité, saisissez le mot de passe temporaire du référentiel fourni. Cette commande télécharge toutes les images utilisées par l'opérateur, y compris pour les fonctionnalités optionnelles. Voir ci-dessous pour les fonctionnalités pour lesquelles ces images sont utilisées.

Fonctionnalités de l'opérateur principal et surveillance de Kubernetes

- surveillance netapp
- proxy kube-rbac
- métriques d'état de kube
- télégraphe
- utilisateur root sans distribution

Journal des événements

- bit courant

- exportateur d'événements Kubernetes

Performances et carte du réseau

- ci-net-observer

Poussez l'image Docker de l'opérateur vers votre référentiel Docker privé/local/d'entreprise conformément à vos politiques d'entreprise. Assurez-vous que les balises d'image et les chemins d'accès aux répertoires de ces images dans votre référentiel sont cohérents avec ceux du référentiel Data Infrastructure Insights .

Modifiez le déploiement de l'opérateur de surveillance dans operator-deployment.yaml et modifiez toutes les références d'image pour utiliser votre référentiel Docker privé.

```
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>
```

Modifiez AgentConfiguration dans operator-config.yaml pour refléter le nouvel emplacement du référentiel Docker. Créez un nouveau imagePullSecret pour votre référentiel privé, pour plus de détails, consultez *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
```

**Instructions OpenShift**

Si vous utilisez OpenShift 4.6 ou une version ultérieure, vous devez modifier AgentConfiguration dans *operator-config.yaml* pour activer le paramètre *runPrivileged* :

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

Openshift peut implémenter un niveau de sécurité supplémentaire qui peut bloquer l'accès à certains composants Kubernetes.

### Autorisations

Si le cluster que vous surveillez contient des ressources personnalisées qui n'ont pas de ClusterRole,"agrégats à visualiser" , vous devrez accorder manuellement à l'opérateur l'accès à ces ressources pour les surveiller avec les journaux d'événements.

1. Modifiez *operator-additional-permissions.yaml* avant l'installation, ou après l'installation, modifiez la ressource *ClusterRole/<namespace>-additional-permissions*
2. Créez une nouvelle règle pour les apiGroups et ressources souhaités avec les verbes ["get", "watch", "list"]. Voir \ https://kubernetes.io/docs/reference/access-authn-authz/rbac/
3. Appliquez vos modifications au cluster

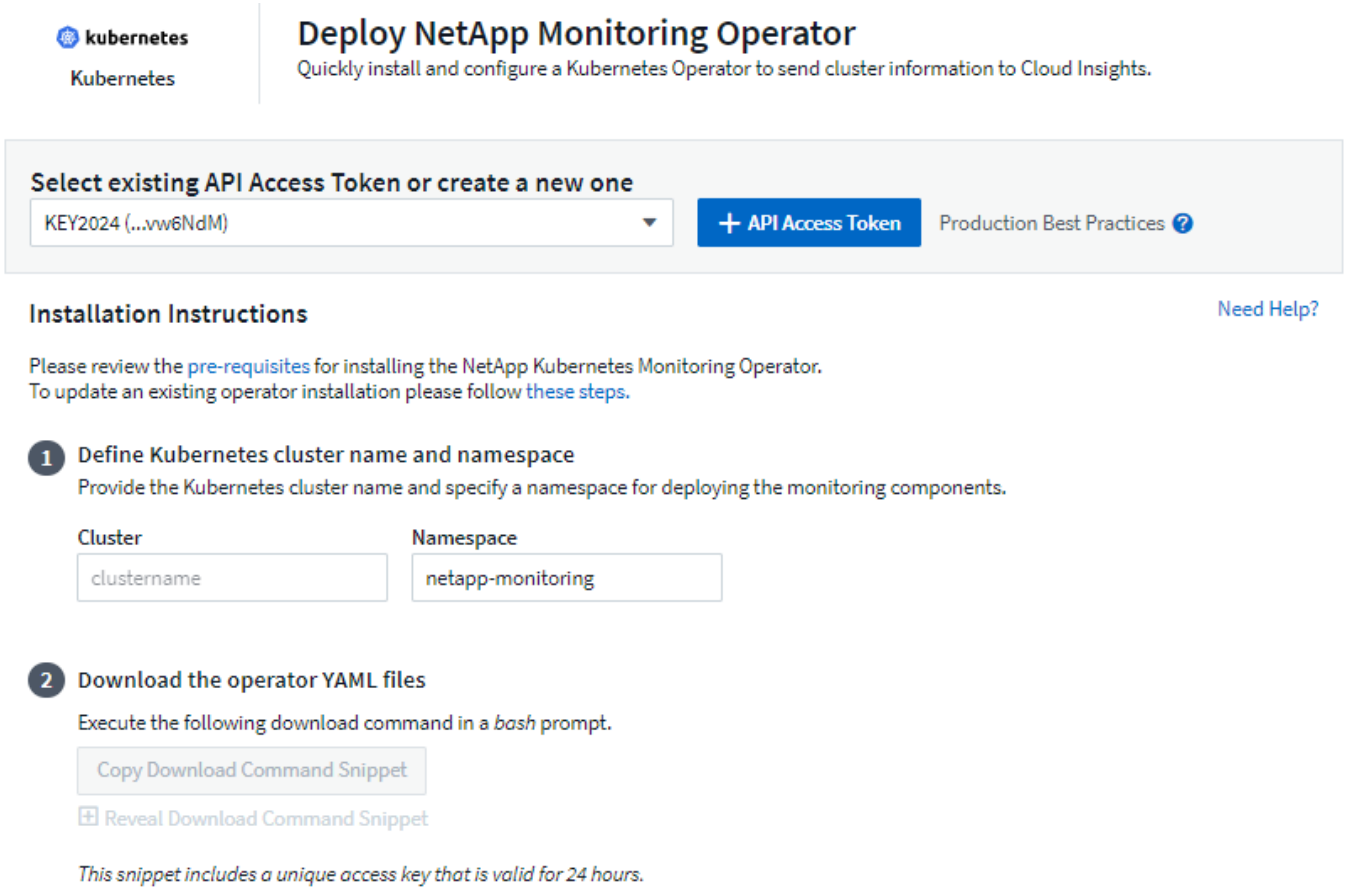
# Installation et configuration de l'opérateur de surveillance Kubernetes

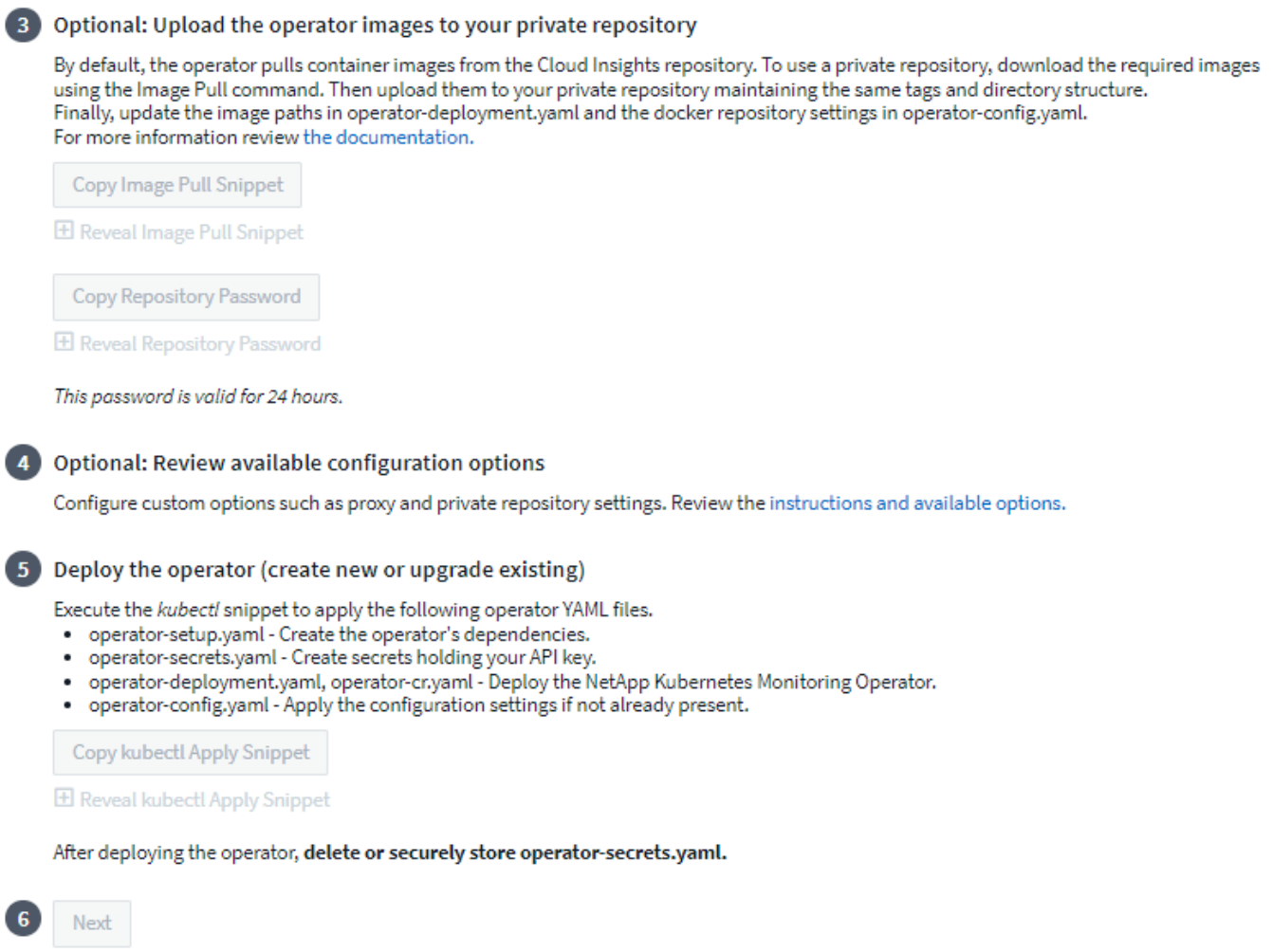
Data Infrastructure Insights propose l'opérateur de surveillance Kubernetes pour la collection Kubernetes. Accédez à **Kubernetes > Collecteurs > +Kubernetes Collector** pour déployer un nouvel opérateur.

## Avant d'installer Kubernetes Monitoring Operator

Voir le"Prérequis" documentation avant d'installer ou de mettre à niveau Kubernetes Monitoring Operator.

## Installation de l'opérateur de surveillance Kubernetes






**Étapes pour installer l'agent Kubernetes Monitoring Operator sur Kubernetes :**

1. Saisissez un nom de cluster et un espace de noms uniques. Si vous êtesmise à niveau à partir d'un opérateur Kubernetes précédent, utilisez le même nom de cluster et le même espace de noms.
2. Une fois ces informations saisies, vous pouvez copier l'extrait de commande de téléchargement dans le presse-papiers.
3. Collez l'extrait dans une fenêtre *bash* et exécutez-le. Les fichiers d'installation de l'opérateur seront téléchargés. Notez que l'extrait possède une clé unique et est valable 24 heures.
4. Si vous disposez d'un référentiel personnalisé ou privé, copiez l'extrait d'image facultatif, collez-le dans un shell *bash* et exécutez-le. Une fois les images extraites, copiez-les dans votre référentiel privé. Assurez-vous de conserver les mêmes balises et la même structure de dossiers. Mettez à jour les chemins dans *operator-deployment.yaml* ainsi que les paramètres du référentiel Docker dans *operator-config.yaml*.
5. Si vous le souhaitez, examinez les options de configuration disponibles telles que les paramètres de proxy ou de référentiel privé. Vous pouvez en savoir plus sur"options de configuration" .
6. Lorsque vous êtes prêt, déployez l'opérateur en copiant l'extrait kubectl Apply, en le téléchargeant et en l'exécutant.
7. L'installation se déroule automatiquement. Une fois terminé, cliquez sur le bouton *Suivant*.
8. Une fois l'installation terminée, cliquez sur le bouton *Suivant*. Assurez-vous également de supprimer ou de stocker en toute sécurité le fichier *operator-secrets.yaml*.

Si vous disposez d'un référentiel personnalisé, lisez à proposen utilisant un référentiel Docker

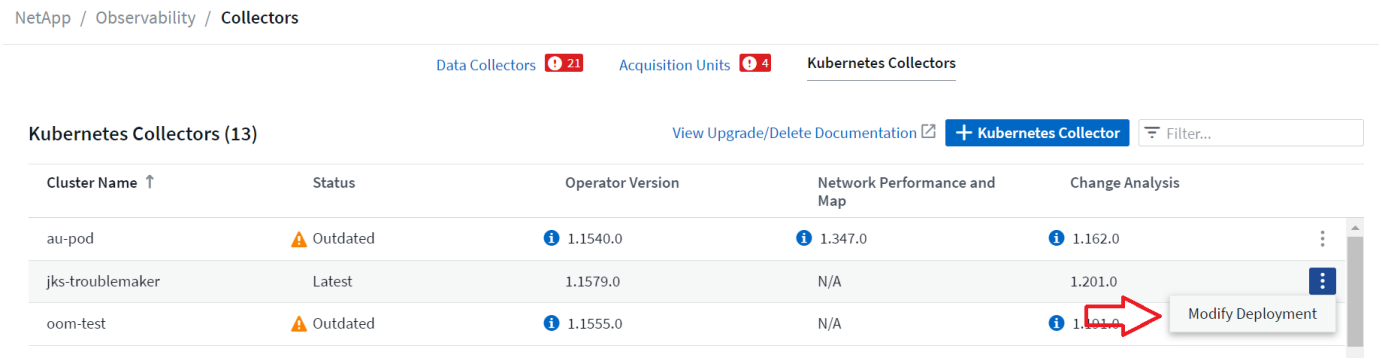
personnalisé/privé .

## Composants de surveillance Kubernetes

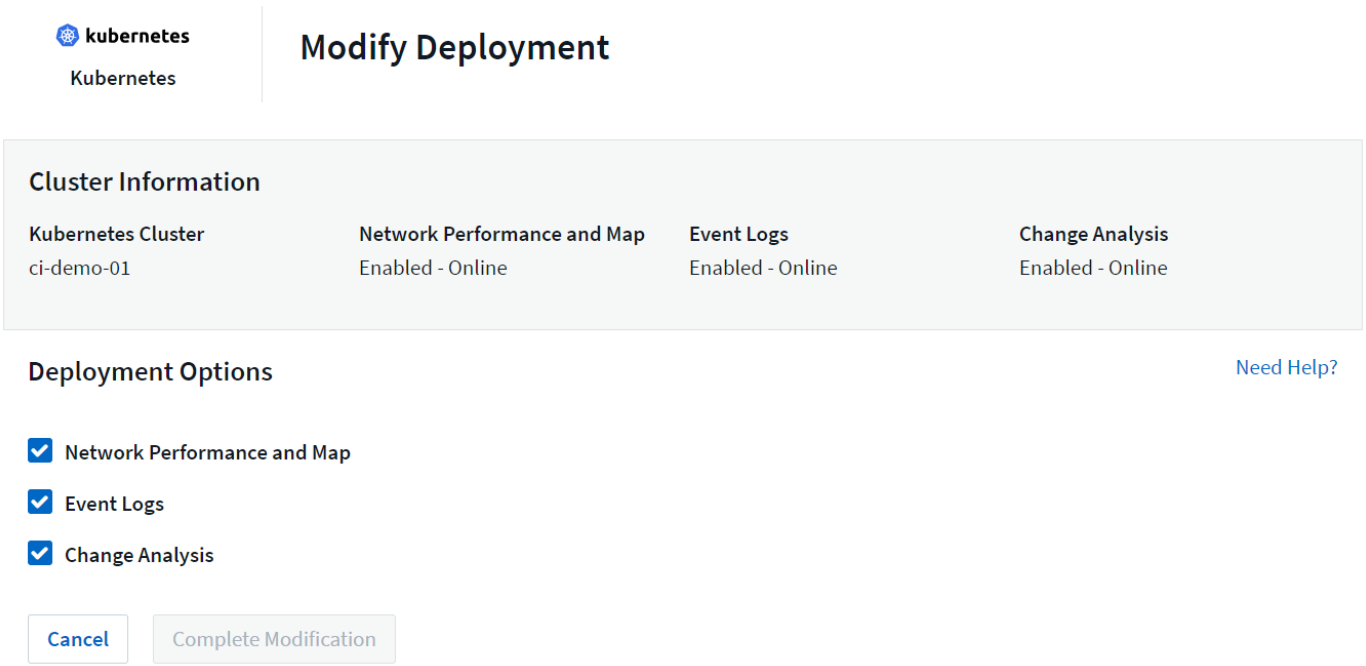
Data Infrastructure Insights Kubernetes Monitoring comprend quatre composants de surveillance :

- Mesures de cluster
- Performances et carte du réseau (facultatif)
- Journaux d'événements (facultatif)
- Analyse des changements (facultatif)

Les composants facultatifs ci-dessus sont activés par défaut pour chaque collecteur Kubernetes ; si vous décidez que vous n'avez pas besoin d'un composant pour un collecteur particulier, vous pouvez le désactiver en accédant à **Kubernetes > Collecteurs** et en sélectionnant *Modifier le déploiement* dans le menu « trois points » du collecteur à droite de l'écran.




L'écran affiche l'état actuel de chaque composant et vous permet de désactiver ou d'activer les composants de ce collecteur selon vos besoins.



## Mise à niveau vers la dernière version de Kubernetes Monitoring Operator

### Mises à niveau du bouton-poussoir DII

Vous pouvez mettre à niveau l'opérateur de surveillance Kubernetes via la page Collecteurs Kubernetes DII. Cliquez sur le menu à côté du cluster que vous souhaitez mettre à niveau et sélectionnez *Mettre à niveau*. L'opérateur vérifiera les signatures d'image, effectuera un instantané de votre installation actuelle et effectuera la mise à niveau. Dans quelques minutes, vous devriez voir l'état de l'opérateur progresser de « Mise à niveau en cours » à « Dernière mise à jour ». Si vous rencontrez une erreur, vous pouvez sélectionner le statut d'erreur pour plus de détails et vous référer au tableau de dépannage des mises à niveau par bouton-poussoir ci-dessous.

#### Mises à niveau par simple pression d'un bouton avec des référentiels privés

Si votre opérateur est configuré pour utiliser un référentiel privé, assurez-vous que toutes les images requises pour exécuter l'opérateur et leurs signatures sont disponibles dans votre référentiel. Si vous rencontrez une erreur pendant le processus de mise à niveau pour des images manquantes, ajoutez-les simplement à votre référentiel et réessayez la mise à niveau. Pour télécharger les signatures d'image dans votre référentiel, veuillez utiliser l'outil de cosignature comme suit, en veillant à télécharger les signatures pour toutes les images spécifiées sous 3 Facultatif : Télécharger les images de l'opérateur dans votre référentiel privé > Extrait d'image

```
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>
```

#### Revenir à une version précédemment exécutée

Si vous avez effectué une mise à niveau à l'aide de la fonction de mise à niveau par simple pression sur un bouton et que vous rencontrez des difficultés avec la version actuelle de l'opérateur dans les sept jours suivant la mise à niveau, vous pouvez rétrograder vers la version précédemment exécutée à l'aide de l'instantané créé pendant le processus de mise à niveau. Cliquez sur le menu à côté du cluster que vous souhaitez restaurer et sélectionnez *Restaurer*.

### Mises à niveau manuelles

Déterminez si une *AgentConfiguration* existe avec l'opérateur existant (si votre espace de noms n'est pas le *netapp-monitoring* par défaut, remplacez-le par l'espace de noms approprié) :

```
 kubectl -n netapp-monitoring get agentconfiguration netapp-ci-monitoring-
configuration
Si une _AgentConfiguration_ existe :
```

- Installationle dernier opérateur sur l'opérateur existant.
    - Assurez-vous que vous êtesextraire les dernières images de conteneurs si vous utilisez un référentiel personnalisé.

Si *AgentConfiguration* n'existe pas :

- Notez le nom de votre cluster tel qu'il est reconnu par Data Infrastructure Insights (si votre espace de noms n'est pas le netapp-monitoring par défaut, remplacez-le par l'espace de noms approprié) :

```
 kubectl -n netapp-monitoring get agent -o
jsonpath='{.items[0].spec.cluster-name}'
* Créez une sauvegarde de l'opérateur existant (si votre espace de noms
n'est pas le netapp-monitoring par défaut, remplacez-le par l'espace de
noms approprié) :
```

```
 kubectl -n netapp-monitoring get agent -o yaml > agent_backup.yaml
* <<to-remove-the-kubernetes-monitoring-
operator,Désinstaller>>l'opérateur existant.
* <<installing-the-kubernetes-monitoring-operator,Installation>>le
dernier opérateur.
```

  - Utilisez le même nom de cluster.
  - Après avoir téléchargé les derniers fichiers YAML de l'opérateur, reportez toutes les personnalisations trouvées dans *agent_backup.yaml* vers le fichier *operator-config.yaml* téléchargé avant le déploiement.
  - Assurez-vous que vous êtesextraire les dernières images de conteneurs si vous utilisez un référentiel personnalisé.

## Arrêt et démarrage de l'opérateur de surveillance Kubernetes

Pour arrêter l'opérateur de surveillance Kubernetes :

```
 kubectl -n netapp-monitoring scale deploy monitoring-operator
--replicas=0
Pour démarrer l'opérateur de surveillance Kubernetes :
```

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

## Désinstallation

### Pour supprimer l'opérateur de surveillance Kubernetes

Notez que l'espace de noms par défaut pour l'opérateur de surveillance Kubernetes est « netapp-monitoring ». Si vous avez défini votre propre espace de noms, remplacez cet espace de noms dans ces commandes et fichiers et dans tous les suivants.

Les versions plus récentes de l'opérateur de surveillance peuvent être désinstallées avec les commandes suivantes :

```
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 l'opérateur de surveillance a été déployé dans son propre espace de noms dédié, supprimez l'espace de noms :

```
 kubectl delete ns <NAMESPACE>
Remarque : si la première commande renvoie « Aucune ressource trouvée »,
utilisez les instructions suivantes pour désinstaller les anciennes
versions de l'opérateur de surveillance.
```

Exécutez chacune des commandes suivantes dans l'ordre. Selon votre installation actuelle, certaines de ces commandes peuvent renvoyer des messages « objet non trouvé ». Ces messages peuvent être ignorés en toute sécurité.

```
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 une contrainte de contexte de sécurité a été créée précédemment :

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

## À propos de Kube-state-metrics

L'opérateur de surveillance NetApp Kubernetes installe ses propres métriques d'état Kube pour éviter tout conflit avec d'autres instances.

Pour plus d'informations sur Kube-State-Metrics, voir"cette page" .

## Configuration/Personnalisation de l'opérateur

Ces sections contiennent des informations sur la personnalisation de la configuration de votre opérateur, l'utilisation d'un proxy, l'utilisation d'un référentiel Docker personnalisé ou privé ou l'utilisation d'OpenShift.

### Options de configuration

Les paramètres les plus fréquemment modifiés peuvent être configurés dans la ressource personnalisée *AgentConfiguration*. Vous pouvez modifier cette ressource avant de déployer l'opérateur en modifiant le fichier *operator-config.yaml*. Ce fichier comprend des exemples de paramètres commentés. Voir la liste des"paramètres disponibles" pour la version la plus récente de l'opérateur.

Vous pouvez également modifier cette ressource après le déploiement de l'opérateur à l'aide de la commande suivante :

```
 kubectl -n netapp-monitoring edit AgentConfiguration
Pour déterminer si votre version déployée de l'opérateur prend en charge
_AgentConfiguration_, exécutez la commande suivante :
```

```
 kubectl get crd agentconfigurations.monitoring.netapp.com
Si vous voyez un message « Erreur du serveur (NotFound) », votre opérateur
doit être mis à niveau avant de pouvoir utiliser AgentConfiguration.
```

### Configuration de la prise en charge du proxy

Il existe deux endroits où vous pouvez utiliser un proxy sur votre locataire afin d'installer Kubernetes Monitoring Operator. Il peut s'agir des mêmes systèmes proxy ou de systèmes proxy distincts :

- Proxy nécessaire lors de l'exécution de l'extrait de code d'installation (à l'aide de « curl ») pour connecter le système où l'extrait est exécuté à votre environnement Data Infrastructure Insights
- Proxy requis par le cluster Kubernetes cible pour communiquer avec votre environnement Data Infrastructure Insights

Si vous utilisez un proxy pour l'un ou les deux, afin d'installer Kubernetes Operating Monitor, vous devez d'abord vous assurer que votre proxy est configuré pour permettre une bonne communication avec votre environnement Data Infrastructure Insights . Si vous disposez d'un proxy et pouvez accéder à Data Infrastructure Insights à partir du serveur/de la machine virtuelle à partir duquel vous souhaitez installer l'opérateur, votre proxy est probablement configuré correctement.

Pour le proxy utilisé pour installer Kubernetes Operating Monitor, avant d'installer l'opérateur, définissez les variables d'environnement *http_proxy/https_proxy*. Pour certains environnements proxy, vous devrez peut-être également définir la variable d'environnement *no_proxy*.

Pour définir la ou les variables, effectuez les étapes suivantes sur votre système **avant** d'installer Kubernetes Monitoring Operator :

1. Définissez les variables d'environnement *https_proxy* et/ou *http_proxy* pour l'utilisateur actuel :
    a. Si le proxy en cours de configuration ne dispose pas d'authentification (nom d'utilisateur/mot de passe), exécutez la commande suivante :

```
 export https_proxy=<proxy_server>:<proxy_port>
.. Si le proxy en cours de configuration dispose d'une
authentification (nom d'utilisateur/mot de passe), exécutez cette
commande :
```

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

Pour que le proxy utilisé pour votre cluster Kubernetes communique avec votre environnement Data Infrastructure Insights , installez Kubernetes Monitoring Operator après avoir lu toutes ces instructions.

Configurez la section proxy de *AgentConfiguration* dans *operator-config.yaml* avant de déployer le Kubernetes Monitoring Operator.

```
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
  ...
...
```

**Utiliser un référentiel Docker personnalisé ou privé**

Par défaut, l'opérateur de surveillance Kubernetes extrait les images de conteneur du référentiel Data Infrastructure Insights . Si vous disposez d'un cluster Kubernetes utilisé comme cible pour la surveillance et que ce cluster est configuré pour extraire uniquement des images de conteneur à partir d'un référentiel Docker personnalisé ou privé ou d'un registre de conteneurs, vous devez configurer l'accès aux conteneurs nécessaires à l'opérateur de surveillance Kubernetes.

Exécutez « Image Pull Snippet » à partir de la mosaïque d'installation de NetApp Monitoring Operator. Cette commande se connectera au référentiel Data Infrastructure Insights , extraira toutes les dépendances d'image pour l'opérateur et se déconnectera du référentiel Data Infrastructure Insights . Lorsque vous y êtes invité, saisissez le mot de passe temporaire du référentiel fourni. Cette commande télécharge toutes les images utilisées par l'opérateur, y compris pour les fonctionnalités optionnelles. Voir ci-dessous pour les fonctionnalités pour lesquelles ces images sont utilisées.

Fonctionnalités de l'opérateur principal et surveillance de Kubernetes

- surveillance netapp
- proxy ci-kube-rbac
- ci-ksm
- ci-telegraf
- utilisateur root sans distribution

Journal des événements

- ci-fluent-bit
- exportateur d'événements ci-kubernetes

Performances et carte du réseau

- ci-net-observer

Poussez l'image Docker de l'opérateur vers votre référentiel Docker privé/local/d'entreprise conformément à vos politiques d'entreprise. Assurez-vous que les balises d'image et les chemins d'accès aux répertoires de ces images dans votre référentiel sont cohérents avec ceux du référentiel Data Infrastructure Insights .

Modifiez le déploiement de l'opérateur de surveillance dans operator-deployment.yaml et modifiez toutes les références d'image pour utiliser votre référentiel Docker privé.

```
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>
```

Modifiez *AgentConfiguration* dans *operator-config.yaml* pour refléter le nouvel emplacement du dépôt docker. Créez un nouveau imagePullSecret pour votre dépôt privé, pour plus de détails, consultez *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
```

#### Jeton d'accès API pour les mots de passe à long terme

Certains environnements (par exemple, les dépôts proxy) exigent des mots de passe à long terme pour le dépôt docker de Data Infrastructure Insights. Le mot de passe fourni dans l'interface utilisateur lors de l'installation n'est valable que 24 heures. Au lieu de cela, on peut utiliser un jeton d'accès API comme mot de passe du dépôt docker. Ce mot de passe sera valable aussi longtemps que le jeton d'accès API est valide. On peut générer un nouveau jeton d'accès API à cette fin ou en utiliser un existant.

"Lire ici" pour obtenir des instructions sur la création d'un nouveau jeton d'accès API.

Pour extraire un jeton d'accès API existant à partir d'un fichier *operator-secrets.yaml* téléchargé, les utilisateurs peuvent exécuter la commande suivante :

```
grep '\.dockerconfigjson' operator-secrets.yaml |sed 's/.*\.dockerconfigjson:
//g' |base64 -d |jq
```

Pour extraire un jeton d'accès API existant d'une installation d'opérateur en cours d'exécution, les utilisateurs peuvent exécuter la commande suivante :

```
kubectl -n netapp-monitoring get secret netapp-ci-docker -o
jsonpath='{.data.\.dockerconfigjson}' |base64 -d |jq
```

### Instructions OpenShift

Si vous utilisez OpenShift 4.6 ou une version ultérieure, vous devez modifier *AgentConfiguration* dans *operator-config.yaml* pour activer le paramètre *runPrivileged* :

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

Openshift peut implémenter un niveau de sécurité supplémentaire qui peut bloquer l'accès à certains composants Kubernetes.

### Tolérances et souillures

Les DaemonSets *netapp-ci-telegraf-ds*, *netapp-ci-fluent-bit-ds* et *netapp-ci-net-observer-l4-ds* doivent planifier un pod sur chaque nœud de votre cluster afin de collecter correctement les données sur tous les nœuds.

L'opérateur a été configuré pour tolérer certaines **souillures** bien connues. Si vous avez configuré des taches personnalisées sur vos nœuds, empêchant ainsi les pods de s'exécuter sur chaque nœud, vous pouvez créer une **tolérance** pour ces taches"dans *AgentConfiguration*" . Si vous avez appliqué des taches personnalisées à tous les nœuds de votre cluster, vous devez également ajouter les tolérances nécessaires au déploiement de l'opérateur pour permettre la planification et l'exécution du pod de l'opérateur.

En savoir plus sur Kubernetes"Souillures et tolérances" .

Retour à la"* Page d'installation de l'opérateur de surveillance NetApp Kubernetes*"

## Une note sur les secrets

Pour supprimer l'autorisation permettant à l'opérateur de surveillance Kubernetes d'afficher les secrets à l'échelle du cluster, supprimez les ressources suivantes du fichier *operator-setup.yaml* avant l'installation :

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

S'il s'agit d'une mise à niveau, supprimez également les ressources de votre cluster :

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

Si l'analyse des changements est activée, modifiez *AgentConfiguration* ou *operator-config.yaml* pour supprimer le commentaire de la section de gestion des changements et inclure *kindsToIgnoreFromWatch: '"secrets"'* sous la section de gestion des changements. Notez la présence et la position des guillemets simples et doubles dans cette ligne.

```
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"'
  ...
```

## Vérification des signatures d'image de l'opérateur de surveillance Kubernetes

L'image de l'opérateur et toutes les images associées qu'il déploie sont signées par NetApp. Vous pouvez vérifier manuellement les images avant l'installation à l'aide de l'outil de cosignature ou configurer un contrôleur d'admission Kubernetes. Pour plus de détails, veuillez consulter le"Documentation Kubernetes" .

La clé publique utilisée pour vérifier les signatures d'image est disponible dans la mosaïque d'installation de l'opérateur de surveillance sous *Facultatif : téléchargez les images de l'opérateur dans votre référentiel privé >*

*Clé publique de signature d'image*

Pour vérifier manuellement une signature d'image, procédez comme suit :

1. Copiez et exécutez l'extrait d'image
2. Copiez et saisissez le mot de passe du référentiel lorsque vous y êtes invité
3. Stocker la clé publique de signature d'image (dii-image-signing.pub dans l'exemple)
4. Vérifiez les images à l'aide de cosign. Reportez-vous à l'exemple suivant d'utilisation de cosignature

```
$ 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}]
```

## Dépannage

Quelques éléments à essayer si vous rencontrez des problèmes lors de la configuration de l'opérateur de surveillance Kubernetes :

| Problème: | Essayez ceci: |
| --- | --- |
| Je ne vois pas d'hyperlien/connexion entre mon volume persistant Kubernetes et le périphérique de stockage back-end correspondant. Mon volume persistant Kubernetes est configuré à l'aide du nom d'hôte du serveur de stockage. | Suivez les étapes pour désinstaller l'agent Telegraf existant, puis réinstaller le dernier agent Telegraf. Vous devez utiliser Telegraf version 2.0 ou ultérieure et votre stockage de cluster Kubernetes doit être activement surveillé par Data Infrastructure Insights. |

| Problème: | Essayez ceci: |
| --- | --- |
| Je vois des messages dans les journaux ressemblant à ce qui suit : E0901 15:21:39.962145 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: Failed to list *v1.MutatingWebhookConfiguration: the server could not find the requested resource E0901 15:21:43.168161 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: Failed to list *v1.Lease: the server could not find the requested resource (get leases.coordination.k8s.io) etc. | Ces messages peuvent se produire si vous exécutez kube-state-metrics version 2.0.0 ou supérieure avec des versions de Kubernetes inférieures à 1.20. Pour obtenir la version de Kubernetes : *kubectl version* Pour obtenir la version de kube-state-metrics : *kubectl get deploy/kube-state-metrics -o jsonpath='{..image}'* Pour éviter que ces messages ne se produisent, les utilisateurs peuvent modifier leur déploiement kube-state-metrics pour désactiver les baux suivants : *mutatingwebhookconfigurations validatingwebhookconfigurations volumeattachments resources* Plus précisément, ils peuvent utiliser l'argument CLI suivant : resources=certificatesigningrequests,configmaps,cronjobs,daemonsets, deployments,endpoints,horizontalpodautoscalers,ingresses,jobs,limitranges, namespaces,networkpolicies,nodes,persistentvolumeclaims,persistentvolumes, poddisruptionbudgets,pods,replicasets,replicationcontrollers,resourcequotas, La liste de ressources par défaut est : « certificatesigningrequests,configmaps,cronjobs,daemonsets,deployments,endpoints,horizontalpodautoscalers,ingresses,jobs,leases,limitranges,mutatingwebhookconfigurations,namespaces,networkpolicies,nodes, persistentvolumeclaims,persistentvolumes,poddisruptionbudgets,pods,replicasets,replicationcontrollers,resourcequotas,secrets,services,statefulsets,storageclasses,validatingwebhookconfigurations,volumeattachments » |
| Je vois des messages d'erreur de Telegraf ressemblant à ce qui suit, mais Telegraf démarre et s'exécute : 11 oct. 14:23:41 ip-172-31-39-47 systemd[1] : Démarré L'agent serveur piloté par plug-in pour la création de rapports de métriques dans InfluxDB. 11 oct. 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="échec de la création du répertoire de cache. /etc/telegraf/.cache/snowflake, err : mkdir /etc/telegraf/.ca che : permission refusée. ignoré\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" Oct 11 14:23:41 ip-172-31-39-47 telegraf[1827] : time="2021-10-11T14:23:41Z" level=error msg="échec d'ouverture. Ignoré. ouvrez /etc/telegraf/.cache/snowflake/ocsp_response_cache.json : aucun fichier ou répertoire de ce type\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 Je ! Démarrage de Telegraf 1.19.3 | Il s'agit d'un problème connu. Se référer à"Cet article GitHub" pour plus de détails. Tant que Telegraf est opérationnel, les utilisateurs peuvent ignorer ces messages d'erreur. |

| Problème: | Essayez ceci: |
| --- | --- |
| Sur Kubernetes, mes pods Telegraf signalent l'erreur suivante : « Erreur lors du traitement des informations mountstats : échec d'ouverture du fichier mountstats : /hostfs/proc/1/mountstats, erreur : ouverture de /hostfs/proc/1/mountstats : autorisation refusée » | Si SELinux est activé et appliqué, il empêche probablement les pods Telegraf d'accéder au fichier /proc/1/mountstats sur le nœud Kubernetes. Pour surmonter cette restriction, modifiez la configuration de l'agent et activez le paramètre runPrivileged. Pour plus de détails, reportez-vous aux instructions OpenShift. |
| Sur Kubernetes, mon pod Telegraf ReplicaSet signale l'erreur suivante : [inputs.prometheus] Erreur dans le plugin : impossible de charger la paire de clés /etc/kubernetes/pki/etcd/server.crt:/etc/kubernetes/pki/etcd/server.key : ouvrir /etc/kubernetes/pki/etcd/server.crt : aucun fichier ou répertoire de ce type | Le pod Telegraf ReplicaSet est destiné à s'exécuter sur un nœud désigné comme maître ou pour etcd. Si le pod ReplicaSet n'est pas en cours d'exécution sur l'un de ces nœuds, vous obtiendrez ces erreurs. Vérifiez si vos nœuds maître/etcd sont contaminés. Si c'est le cas, ajoutez les tolérances nécessaires au Telegraf ReplicaSet, telegraf-rs. Par exemple, modifiez le ReplicaSet… kubectl edit rs telegraf-rs … et ajoutez les tolérances appropriées à la spécification. Ensuite, redémarrez le pod ReplicaSet. |
| J'ai un environnement PSP/PSA. Cela affecte-t-il mon opérateur de surveillance ? | Si votre cluster Kubernetes s'exécute avec la stratégie de sécurité des pods (PSP) ou l'admission de sécurité des pods (PSA) en place, vous devez effectuer une mise à niveau vers la dernière version de Kubernetes Monitoring Operator. Suivez ces étapes pour mettre à niveau vers l'opérateur actuel avec prise en charge de PSP/PSA : 1. Désinstaller l'opérateur de surveillance précédent : 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. Installation la dernière version de l'opérateur de surveillance. |
| J'ai rencontré des problèmes lors du déploiement de l'opérateur et j'utilise PSP/PSA. | 1. Modifiez l'agent à l'aide de la commande suivante : kubectl -n <name-space> edit agent 2. Marquer « security-policy-enabled » comme « faux ». Cela désactivera les politiques de sécurité des pods et l'admission de sécurité des pods et permettra à l'opérateur de se déployer. Confirmez en utilisant les commandes suivantes : kubectl get psp (devrait indiquer que la politique de sécurité du pod a été supprimée) kubectl get all -n <namespace> |
| grep -i psp (devrait indiquer que rien n'est trouvé) | Erreurs « ImagePullBackoff » observées |
| Ces erreurs peuvent être observées si vous disposez d'un référentiel Docker personnalisé ou privé et que vous n'avez pas encore configuré l'opérateur de surveillance Kubernetes pour le reconnaître correctement. En savoir plus à propos de la configuration pour un référentiel personnalisé/privé. | J'ai un problème avec mon déploiement d'opérateur de surveillance et la documentation actuelle ne m'aide pas à le résoudre. |

| Problème: | Essayez ceci: |
| --- | --- |
| Capturez ou notez autrement la sortie des commandes suivantes et contactez l'équipe de support technique.<br><br>```<br>kubectl -n netapp-monitoring get all<br>kubectl -n netapp-monitoring describe all<br>kubectl -n netapp-monitoring logs <monitoring-operator-pod> --all-containers=true<br>kubectl -n netapp-monitoring logs <telegraf-pod> --all-containers=true<br>``` | Les pods net-observer (Workload Map) dans l'espace de noms Operator sont dans CrashLoopBackOff |
| Ces pods correspondent au collecteur de données Workload Map pour l'observabilité du réseau. Essayez ceci : • Vérifiez les journaux de l'un des pods pour confirmer la version minimale du noyau. Par exemple : ---- {"ci-tenant-id":"votre-id-de-tenant","collector-cluster":"votre-nom-de-cluster-k8s","environment":"prod","level":"error","msg":"échec de validation. Raison : la version du noyau 3.10.0 est inférieure à la version minimale du noyau 4.18.0","time":"2022-11-09T08:23:08Z"} ---- • Les pods Net-observer nécessitent que la version du noyau Linux soit au moins 4.18.0. Vérifiez la version du noyau à l'aide de la commande « uname -r » et assurez-vous qu'elle est >= 4.18.0 | Les pods s'exécutent dans l'espace de noms de l'opérateur (par défaut : netapp-monitoring), mais aucune donnée n'est affichée dans l'interface utilisateur pour la carte de charge de travail ou les métriques Kubernetes dans les requêtes. |
| Vérifiez le réglage de l'heure sur les nœuds du cluster K8S. Pour un audit et des rapports de données précis, il est fortement recommandé de synchroniser l'heure sur la machine Agent à l'aide du protocole Network Time Protocol (NTP) ou du protocole Simple Network Time Protocol (SNTP). | Certains des pods d'observateur réseau dans l'espace de noms Operator sont en état d'attente |
| Net-observer est un DaemonSet et exécute un pod dans chaque nœud du cluster k8s. • Notez le pod qui est en état d'attente et vérifiez s'il rencontre un problème de ressources pour le processeur ou la mémoire. Assurez-vous que la mémoire et le processeur requis sont disponibles dans le nœud. | Je vois ce qui suit dans mes journaux immédiatement après l'installation de Kubernetes Monitoring Operator : [inputs.prometheus] Erreur dans le plugin : erreur lors de la requête HTTP vers http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics : obtenir http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics : numérotation TCP : recherche kube-state-metrics.<namespace>.svc.cluster.local : aucun hôte de ce type |

| Problème: | Essayez ceci: |
| --- | --- |
| Ce message n'apparaît généralement que lorsqu'un nouvel opérateur est installé et que le pod *telegraf-rs* est opérationnel avant le pod *ksm*. Ces messages devraient cesser une fois que tous les pods sont en cours d'exécution. | Je ne vois aucune métrique collectée pour les CronJobs Kubernetes qui existent dans mon cluster. |
| Vérifiez votre version de Kubernetes (c'est-à-dire `kubectl version` ). S'il s'agit de la version 1.20.x ou d'une version antérieure, il s'agit d'une limitation attendue. La version kube-state-metrics déployée avec l'opérateur de surveillance Kubernetes prend uniquement en charge la version v1.CronJob. Avec Kubernetes 1.20.x et versions antérieures, la ressource CronJob se trouve dans v1beta.CronJob. Par conséquent, kube-state-metrics ne peut pas trouver la ressource CronJob. | Après l'installation de l'opérateur, les pods telegraf-ds entrent dans CrashLoopBackOff et les journaux des pods indiquent « su : échec d'authentification ». |
| Modifiez la section telegraf dans *AgentConfiguration*, et définissez *dockerMetricCollectionEnabled* sur false. Pour plus de détails, consultez le "options de configuration" de l'opérateur. … spec: … telegraf: … - name: docker run-mode: - DaemonSet substitutions: - key: DOCKER_UNIX_SOCK_PLACEHOLDER value: unix:///run/docker.sock … … | Je vois des messages d'erreur répétés ressemblant à ce qui suit dans mes journaux Telegraf : E! [agent] Erreur lors de l'écriture dans outputs.http : Post "https://<tenant_url>/rest/v1/lake/ingest/influxdb" : délai de contexte dépassé (Client.Timeout dépassé lors de l'attente des en-têtes) |
| Modifiez la section Telegraf dans *AgentConfiguration* et augmentez *outputTimeout* à 10 s. Pour plus de détails, reportez-vous au manuel de l'opérateur."options de configuration" . | Il me manque des données *involvedobject* pour certains journaux d'événements. |
| Assurez-vous d'avoir suivi les étapes décrites dans la"Autorisations" section ci-dessus. | Pourquoi est-ce que je vois deux pods d'opérateurs de surveillance en cours d'exécution, l'un nommé netapp-ci-monitoring-operator-<pod> et l'autre nommé monitoring-operator-<pod> ? |
| À compter du 12 octobre 2023, Data Infrastructure Insights a remanié l'opérateur pour mieux servir nos utilisateurs ; pour que ces changements soient pleinement adoptés, vous devezsupprimer l'ancien opérateur etinstaller le nouveau . | Mes événements Kubernetes ont cessé de manière inattendue d'être signalés à Data Infrastructure Insights. |
| Récupérer le nom du pod exportateur d'événements :<br>`` `kubectl -n netapp-monitoring get pods `` | grep event-exporter |

| Problème: | Essayez ceci: |
| --- | --- |
| awk '{print $1}' | sed 's/event-exporter./event-exporter/'`<br>Il doit s'agir soit de « netapp-ci-event-exporter » soit de « event-exporter ». Ensuite, modifiez l'agent de surveillance `kubectl -n netapp-monitoring edit agent` et définissez la valeur de LOG_FILE pour refléter le nom de pod d'exportation d'événements approprié trouvé à l'étape précédente. Plus précisément, LOG_FILE doit être défini sur « /var/log/containers/netapp-ci-event-exporter.log » ou « /var/log/containers/event-exporter*.log »<br><br>….<br>fluent-bit:<br>…<br>- name: event-exporter-ci<br>substitutions:<br>- key: LOG_FILE<br>values:<br>- /var/log/containers/netapp-ci-event-exporter*.log<br>…<br>….<br>Alternativement, on peut aussidésinstaller etréinstaller l'agent. |
| Je vois que des pods déployés par l'opérateur de surveillance Kubernetes se bloquent en raison de ressources insuffisantes. | Consultez l'opérateur de surveillance Kubernetes"options de configuration" pour augmenter les limites du processeur et/ou de la mémoire selon les besoins. |
| Une image manquante ou une configuration non valide a entraîné l'échec du démarrage ou de la préparation des pods netapp-ci-kube-state-metrics. Désormais, le StatefulSet est bloqué et les modifications de configuration ne sont pas appliquées aux pods netapp-ci-kube-state-metrics. | Le StatefulSet est dans un"cassé" État. Après avoir résolu les problèmes de configuration, faites rebondir les pods netapp-ci-kube-state-metrics. |
| Les pods netapp-ci-kube-state-metrics ne parviennent pas à démarrer après l'exécution d'une mise à niveau de l'opérateur Kubernetes, générant ErrImagePull (échec de l'extraction de l'image). | Essayez de réinitialiser les pods manuellement. |
| Les messages « Événement rejeté car plus ancien que maxEventAgeSeconds » sont observés pour mon cluster Kubernetes sous Analyse des journaux. | Modifiez l'opérateur *agentconfiguration* et augmentez *event-exporter-maxEventAgeSeconds* (c'est-à-dire à 60 s), *event-exporter-kubeQPS* (c'est-à-dire à 100) et *event-exporter-kubeBurst* (c'est-à-dire à 500). Pour plus de détails sur ces options de configuration, consultez le"options de configuration" page. |

| Problème: | Essayez ceci: |
| --- | --- |
| Telegraf avertit ou plante à cause d'une mémoire verrouillable insuffisante. | Essayez d'augmenter la limite de mémoire verrouillable pour Telegraf dans le système d'exploitation/nœud sous-jacent. Si l'augmentation de la limite n'est pas une option, modifiez la configuration de l'agent NKMO et définissez *unprotected* sur *true*. Cela indiquera à Telegraf de ne pas tenter de réserver des pages de mémoire verrouillées. Bien que cela puisse présenter un risque de sécurité, car les secrets déchiffrés peuvent être échangés sur le disque, cela permet l'exécution dans des environnements où la réservation de mémoire verrouillée n'est pas possible. Pour plus de détails sur les options de configuration *non protégées*, reportez-vous à la"options de configuration" page. |
| Je vois des messages d'avertissement de Telegraf ressemblant à ce qui suit : *W! [inputs.diskio] Impossible de récupérer le nom du disque pour « vdc » : erreur de lecture de /dev/vdc : aucun fichier ou répertoire de ce type* | Pour l'opérateur de surveillance Kubernetes, ces messages d'avertissement sont sans conséquence et peuvent être ignorés. Vous pouvez également modifier la section telegraf dans AgentConfiguration et définir *runDsPrivileged* sur true. Pour plus de détails, consultez le "options de configuration de l'opérateur". |

| Problème: | Essayez ceci: |
| --- | --- |
| Mon pod Fluent-bit échoue avec les erreurs suivantes : [2024/10/16 14:16:23] [error] [/src/fluent-bit/plugins/in_tail/tail_fs_inotify.c:360 errno=24] Trop de fichiers ouverts [2024/10/16 14:16:23] [error] échec d'initialisation de l'entrée tail.0 [2024/10/16 14:16:23] [error] [engine] échec d'initialisation de l'entrée | Essayez de modifier vos paramètres *fsnotify* dans votre cluster :<br><br>`sudo sysctl fs.inotify.max_user_instances (take note of setting)`<br><br>`sudo sysctl fs.inotify.max_user_instances=<something larger than current setting>`<br><br>`sudo sysctl fs.inotify.max_user_watches (take note of setting)`<br><br>`sudo sysctl fs.inotify.max_user_watches=<something larger than current setting>`<br><br>Redémarrez Fluent-bit.<br><br>Remarque : pour que ces paramètres soient persistants après chaque redémarrage du nœud, vous devez placer les lignes suivantes dans */etc/sysctl.conf*<br><br>`fs.inotify.max_user_instances=<something larger than current setting>`<br>`fs.inotify.max_user_watches=<something larger than current setting>` |

| Problème: | Essayez ceci: |
| --- | --- |
| Les pods DS de Telegraf signalent des erreurs liées au fait que le plug-in d’entrée Kubernetes ne parvient pas à effectuer des requêtes HTTP en raison de l’impossibilité de valider le certificat TLS. Par exemple : E! [inputs.kubernetes] Erreur dans le plugin : erreur lors de la requête HTTP vers"<a href="https://<kubelet_IP>:10250/stats/summary": " class="bare">https://<kubelet_IP>:10250/stats/summary":</a> Obtenir"<a href="https://<kubelet_IP>:10250/stats/summary": " class="bare">https://<kubelet_IP>:10250/stats/summary":</a> tls : échec de vérification du certificat : x509 : impossible de valider le certificat pour <kubelet_IP> car il ne contient aucun SAN IP | Cela se produit si le kubelet utilise des certificats auto-signés et/ou si le certificat spécifié n'inclut pas le <kubelet_IP> dans la liste *Subject Alternative Name* des certificats. Pour résoudre ce problème, l'utilisateur peut modifier le"configuration de l'agent" , et définissez *telegraf:insecureK8sSkipVerify* sur *true*. Cela configurera le plugin d'entrée Telegraf pour ignorer la vérification. Alternativement, l'utilisateur peut configurer le kubelet pour"serveurTLSBootstrap" , ce qui déclenchera une demande de certificat à partir de l'API « certificates.k8s.io ». |
| Je rencontre l'erreur suivante dans les pods Fluent-bit et le pod ne peut pas démarrer : [2026/01/12 20:20:32] [error] [sqldb] error=unable to open database file [2026/01/12 20:20:32] [error] [input:tail:tail.0] db: could not create 'in_tail_files' table [2026/01/12 20:20:32] [error] [input:tail:tail.0] could not open/create database [2026/01/12 20:20:32] [error] failed initialize input tail.0 [2026/01/12 20:20:32] [error] [engine] input initialization failed | Assurez-vous que le répertoire hôte dans lequel le fichier DB réside dispose des permissions de lecture/écriture appropriées. Plus précisément, le répertoire hôte doit accorder des permissions de lecture/écriture aux utilisateurs non-root. L'emplacement par défaut du fichier DB est /var/log/ sauf si cela est remplacé par l'option fluent-bit-dbFile *agentconfiguration*. Si SELinux est activé, essayez de définir l'option fluent-bit-seLinuxOptionsType agentconfiguration sur 'spc_t'. |

Des informations complémentaires peuvent être trouvées à partir du"Support" page ou dans le"Matrice de support du collecteur de données" .

# Options de configuration de l'opérateur de surveillance Kubernetes

Le "Opérateur de surveillance Kubernetes" Offre de nombreuses options de personnalisation via le fichier AgentConfiguration. Vous pouvez configurer les limites de ressources, les intervalles de collecte, les paramètres de proxy, les tolérances et les paramètres spécifiques aux composants afin d'optimiser la surveillance de votre environnement Kubernetes. Utilisez ces options pour personnaliser Telegraf, Kube-State-Metrics, la collecte des journaux, le mappage des charges de travail, la gestion des modifications et d'autres composants de surveillance.

## Exemple de fichier AgentConfiguration

Vous trouverez ci-dessous un exemple de fichier *AgentConfiguration*, avec la description de chaque option.

```
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
    ##
```

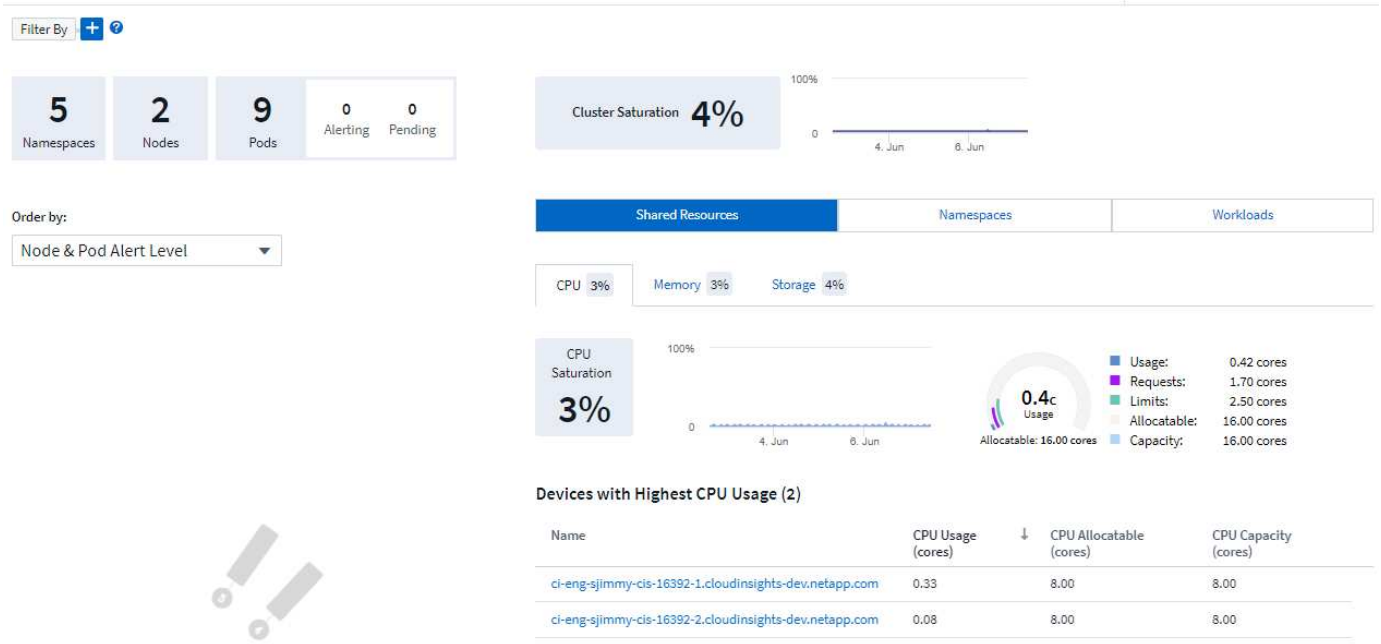
```
    # cpuLimit: '1'
    # memLimit: '1Gi'
    # cpuRequest: '500m'
    # memRequest: '500Mi'

    ##
    ## 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: ''
```

# Page de détails du cluster Kubernetes

La page de détails du cluster Kubernetes affiche un aperçu détaillé de votre cluster Kubernetes.




## Nombre d’espaces de noms, de nœuds et de pods

Les comptes en haut de la page vous indiquent le nombre total d’espaces de noms, de nœuds et de pods dans le cluster, ainsi que le nombre de popds actuellement en alerte et en attente.

## Ressources partagées et saturation

En haut à droite de la page de détails se trouve la saturation de votre cluster sous forme de pourcentage actuel ainsi qu’un graphique montrant la tendance récente au fil du temps. La saturation du cluster correspond à la saturation la plus élevée du processeur, de la mémoire ou du stockage à chaque instant.

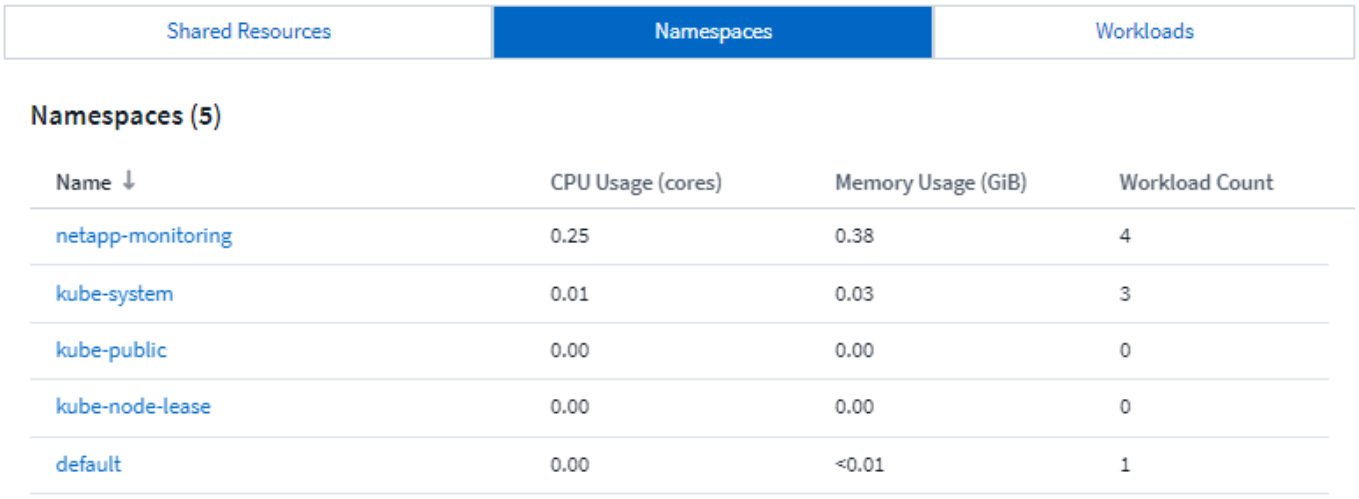
En dessous, la page affiche par défaut l’utilisation des **Ressources partagées**, avec des onglets pour le processeur, la mémoire et le stockage. Chaque onglet affiche le pourcentage de saturation et la tendance au fil du temps, avec des détails d’utilisation supplémentaires. Pour le stockage, la valeur affichée est la plus élevée entre la saturation du backend et du système de fichiers, qui sont calculées indépendamment.

Les appareils les plus utilisés sont indiqués dans un tableau en bas. Cliquez sur n’importe quel lien pour explorer ces appareils.

## Espaces de noms

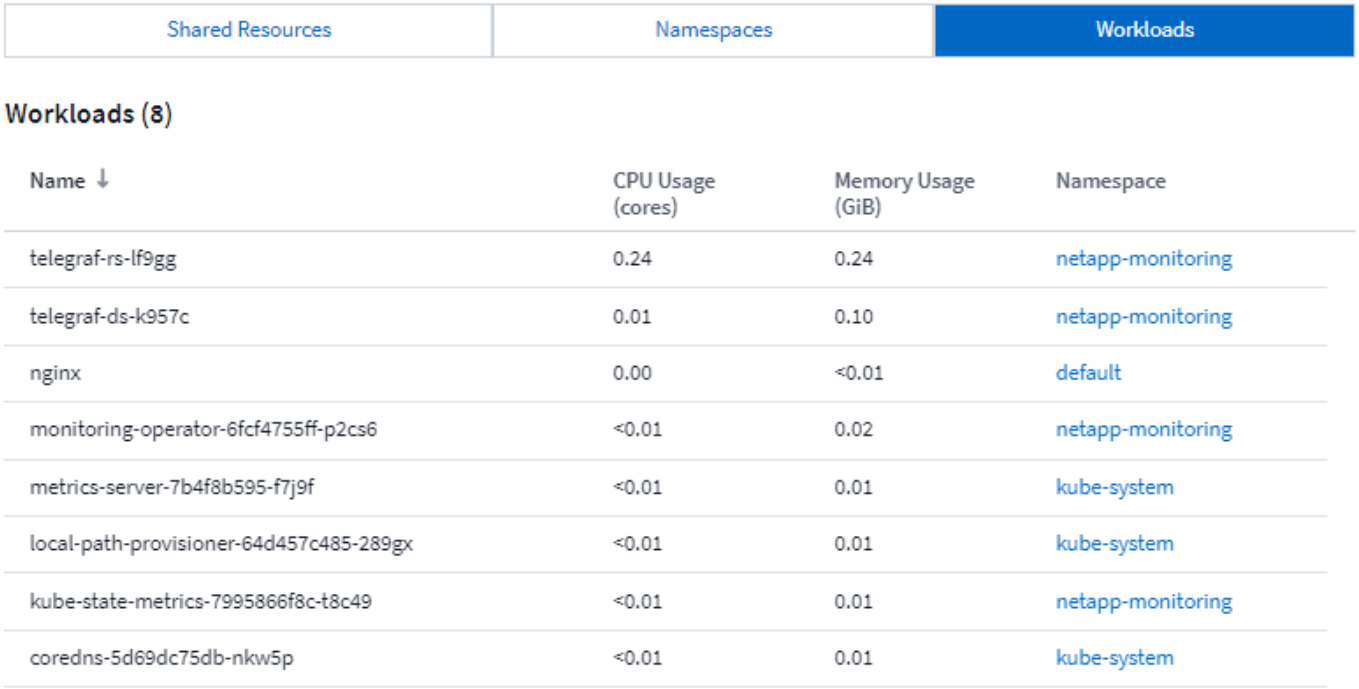
L’onglet Espaces de noms affiche une liste de tous les espaces de noms de votre environnement Kubernetes,

indiquant l'utilisation du processeur et de la mémoire ainsi qu'un nombre de charges de travail dans chaque espace de noms. Cliquez sur les liens Nom pour explorer chaque espace de noms.




## Charges de travail

De même, l'onglet Charges de travail affiche une liste des charges de travail dans chaque espace de noms, indiquant à nouveau l'utilisation du processeur et de la mémoire. En cliquant sur les liens de l'espace de noms, vous accédez à chacun d'eux.



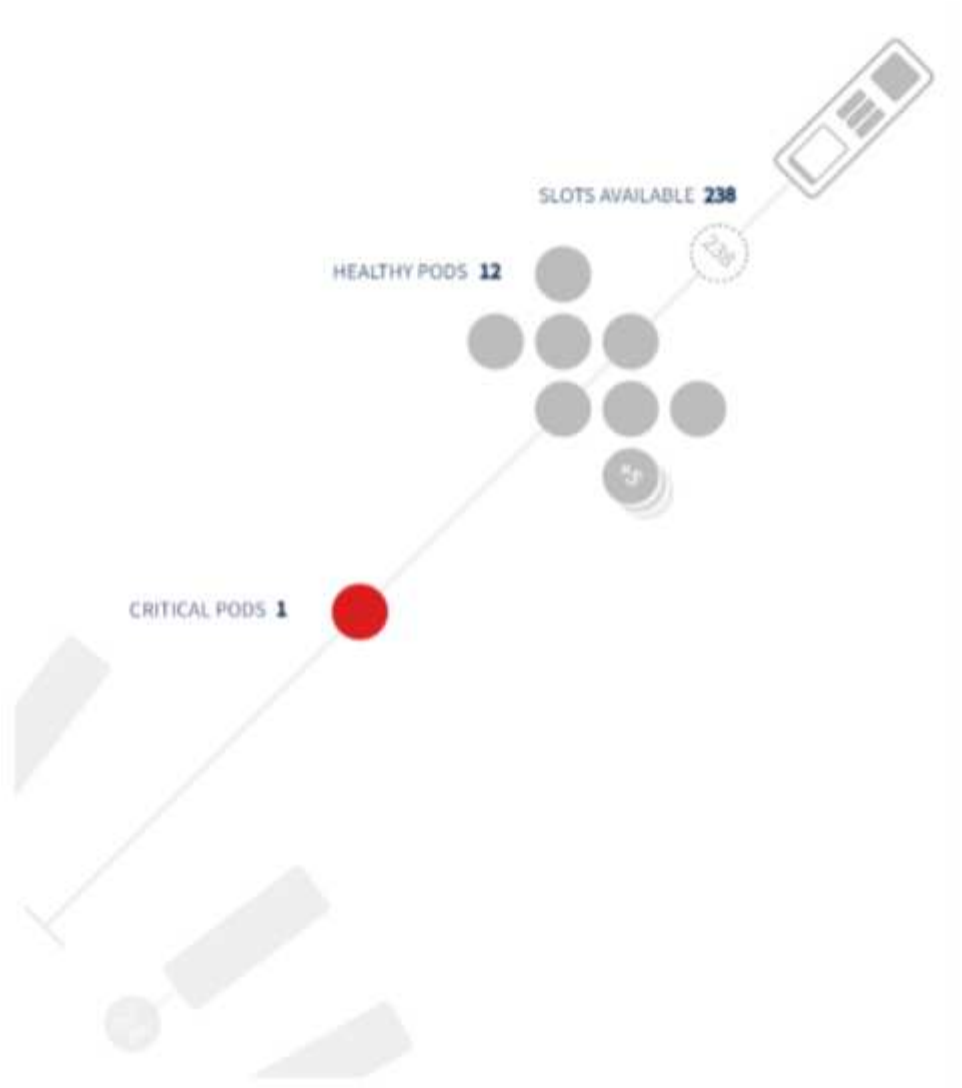

## La « roue » du cluster


La section « Roue » du cluster fournit un aperçu de l'état des nœuds et des pods, que vous pouvez explorer pour obtenir plus d'informations. Si votre cluster contient plus de nœuds que ce qui peut être affiché dans cette zone de la page, vous pourrez tourner la roue à l'aide des boutons disponibles.

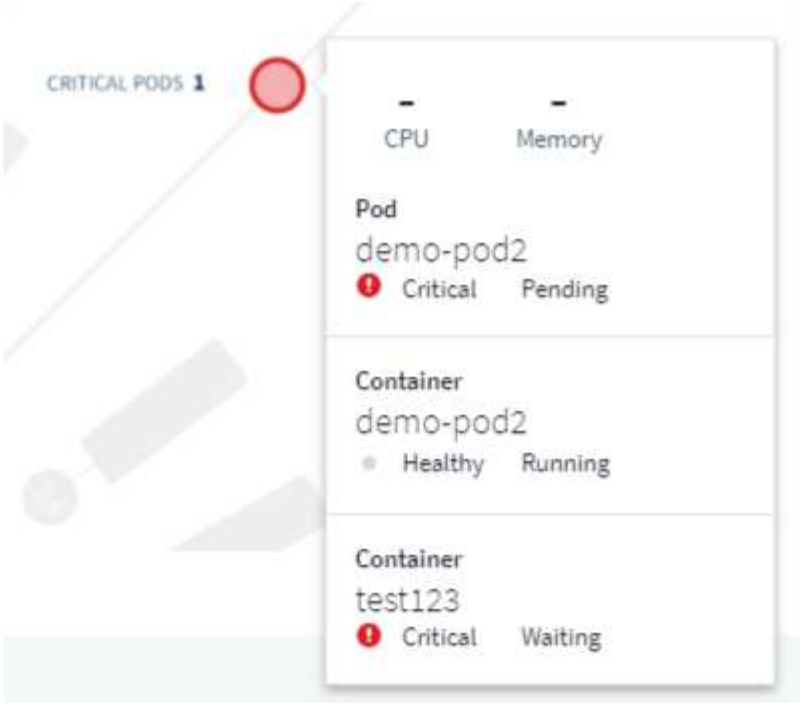
Les pods ou nœuds d'alerte sont affichés en rouge. Les zones « Avertissement » sont affichées en orange. Les pods non planifiés (c'est-à-dire non attachés) s'afficheront dans le coin inférieur de la « Roue » du cluster.

Survoler un pod (cercle) ou un nœud (barre) étendra la vue du nœud.

Cliquer sur le pod ou le nœud dans cette vue effectuera un zoom avant sur la vue du nœud développée.




À partir de là, vous pouvez survoler un élément pour afficher les détails de cet élément. Par exemple, survoler le pod critique dans cet exemple affiche les détails sur ce pod.




Vous pouvez afficher les informations sur le système de fichiers, la mémoire et le processeur en survolant les éléments du nœud.


### Une note sur les jauges

Les jauges de mémoire et de processeur affichent trois couleurs, car elles indiquent la capacité utilisée par rapport à la capacité allouable et à la capacité totale.

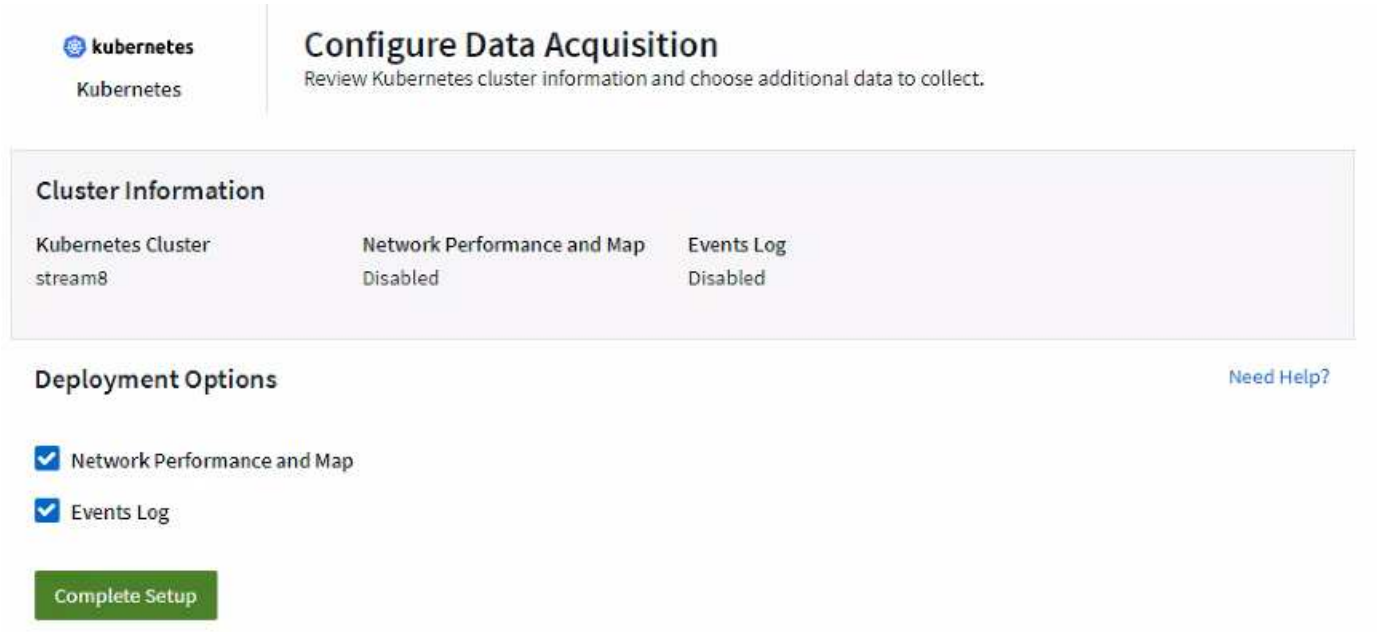
# Surveillance et cartographie des performances du réseau Kubernetes

La fonctionnalité Kubernetes Network Performance Monitoring and Map simplifie le dépannage en mappant les dépendances entre les services (également appelés charges de travail) et offre une visibilité en temps réel sur les latences et les anomalies des performances du réseau pour identifier les problèmes de performances avant qu'ils n'affectent les utilisateurs. Cette capacité aide les organisations à réduire les coûts globaux en analysant et en auditant les flux de trafic Kubernetes.

Principales fonctionnalités : • La carte de charge de travail présente les dépendances et les flux de charge de travail de Kubernetes et met en évidence les problèmes de réseau et de performances. • Surveillez le trafic réseau entre les pods, les charges de travail et les nœuds Kubernetes ; identifiez la source des problèmes de trafic et de latence. • Réduisez les coûts globaux en analysant le trafic réseau entrant, sortant, interrégional et interzone.

## Prérequis

Avant de pouvoir utiliser la surveillance et la cartographie des performances du réseau Kubernetes, vous devez avoir configuré le"Opérateur de surveillance NetApp Kubernetes" pour activer cette option. Lors du déploiement de l'opérateur, sélectionnez la case à cocher « Performances et carte du réseau » pour l'activer. Vous pouvez également activer cette option en accédant à une page de destination Kubernetes et en sélectionnant « Modifier le déploiement ».



## Moniteurs

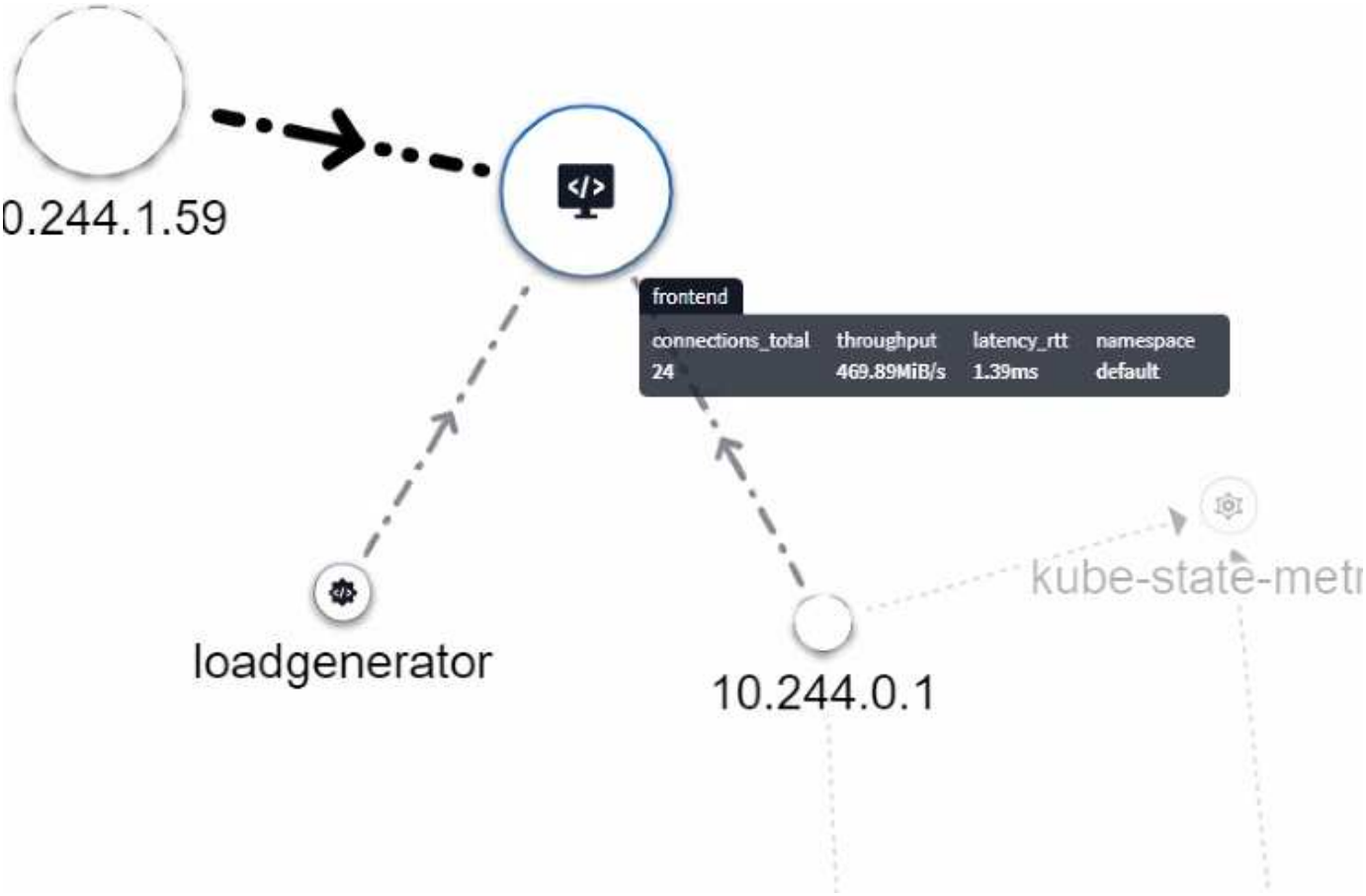
La carte de charge de travail utilise"moniteurs" pour obtenir des informations. Data Infrastructure Insights fournit un certain nombre de moniteurs Kubernetes par défaut (notez que ceux-ci peuvent être *en pause* par défaut). Vous pouvez *Reprendre* (c'est-à-dire activer) les moniteurs que vous souhaitez, ou vous pouvez créer des moniteurs personnalisés pour les objets Kubernetes, que Workload Map utilisera également.

Vous pouvez créer des alertes de métrique Data Infrastructure Insights sur n'importe lequel des types d'objets ci-dessous. Assurez-vous que les données sont regroupées par type d'objet par défaut.

- charge de travail Kubernetes
- ensemble de démons Kubernetes
- déploiement de Kubernetes
- kubernetes.cronjob
- kubernetes.job
- kubernetes.replicaset
- kubernetes.statefulset
- kubernetes.pod
- kubernetes.network_traffic_l4

## La carte

La carte montre les services/charges de travail et leurs relations entre eux. Les flèches indiquent les directions de circulation. Le survol d'une charge de travail affiche des informations récapitulatives pour cette charge de travail, comme vous pouvez le voir dans cet exemple :



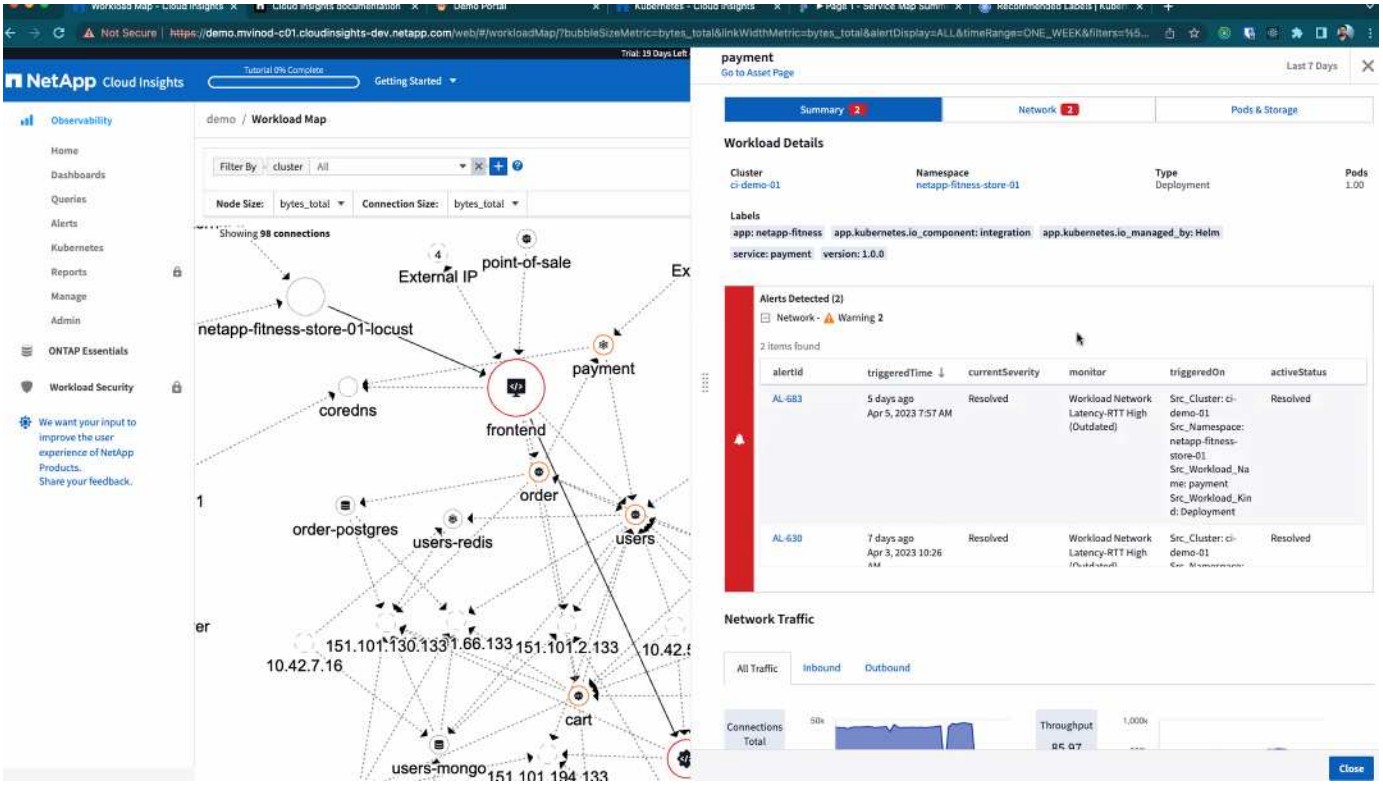

Les icônes dans les cercles représentent différents types de services. Notez que les icônes ne sont visibles que si les objets sous-jacents ontétiquettes .


La taille de chaque cercle indique la taille du nœud. Notez que ces tailles sont relatives, le niveau de zoom de votre navigateur ou la taille de votre écran peuvent affecter les tailles réelles des cercles. De la même manière, le style de ligne de trafic vous donne un aperçu rapide de la taille de la connexion ; les lignes continues en gras correspondent à un trafic élevé, tandis que les lignes pointillées claires correspondent à un trafic plus faible.

Les nombres à l'intérieur des cercles indiquent le nombre de connexions externes actuellement traitées par le service.

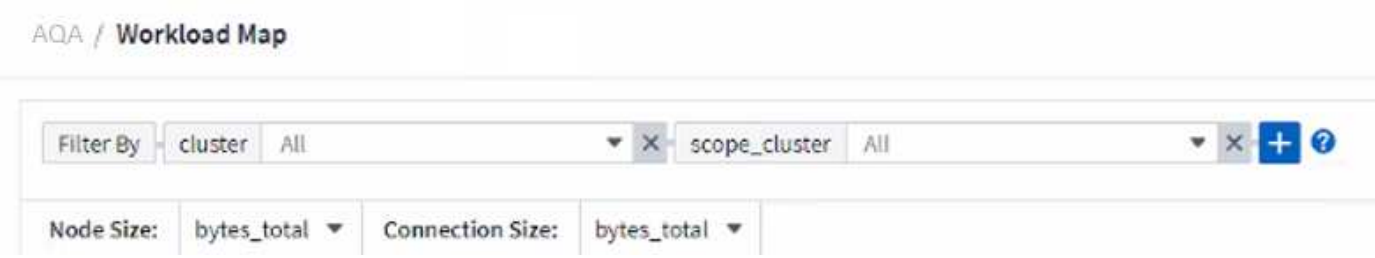

## Détails et alertes de charge de travail

Les cercles affichés en couleur indiquent une alerte de niveau d'avertissement ou critique pour la charge de travail. Passez la souris sur le cercle pour obtenir un résumé du problème ou cliquez sur le cercle pour ouvrir un panneau coulissant avec plus de détails.

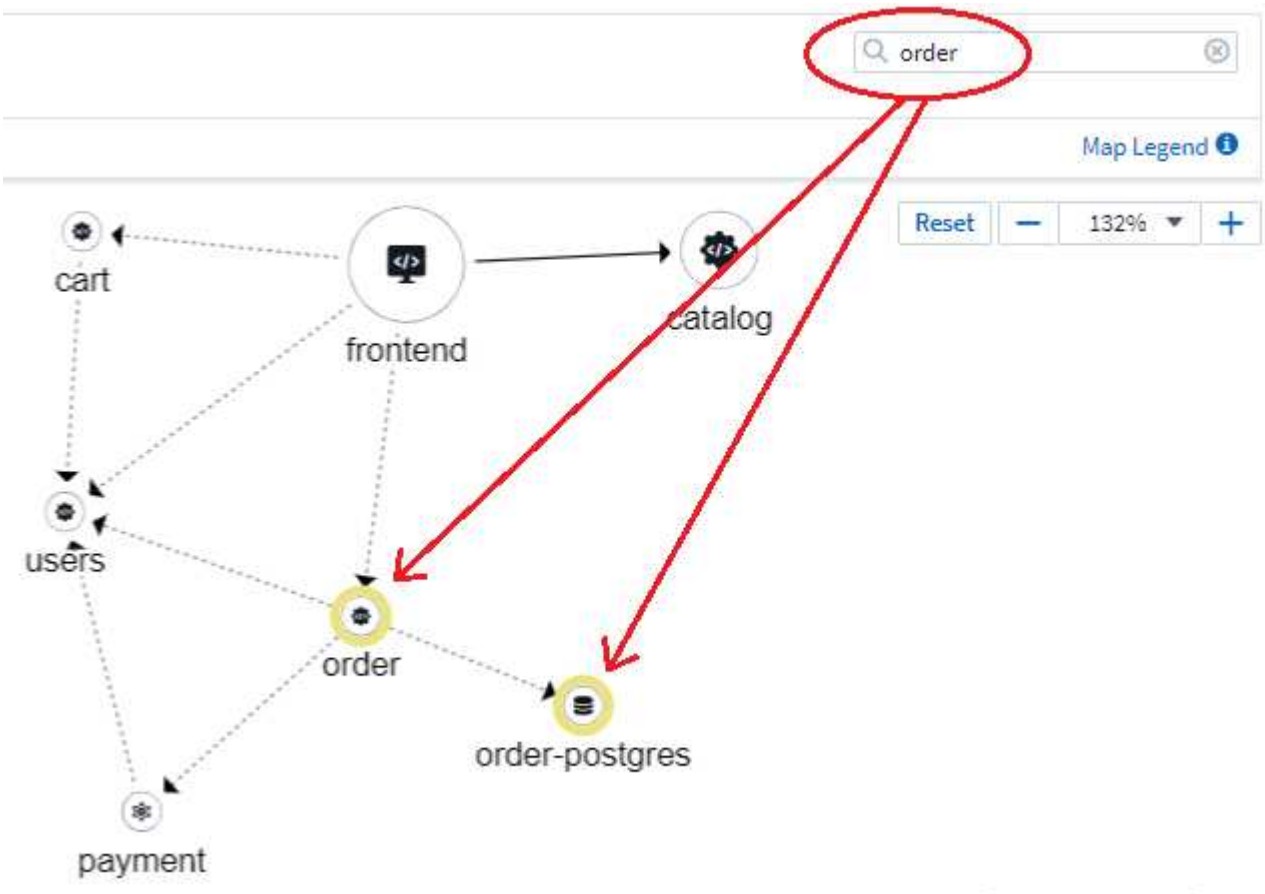



## Recherche et filtrage

Comme avec d'autres fonctionnalités de Data Infrastructure Insights , vous pouvez facilement définir des filtres pour vous concentrer sur les objets spécifiques ou les attributs de charge de travail souhaités.




De même, la saisie d'une chaîne dans le champ *Rechercher* mettra en évidence les charges de travail correspondantes.




## Étiquettes de charge de travail

Les étiquettes de charge de travail sont nécessaires si vous souhaitez que la carte identifie les types de charges de travail affichées (c'est-à-dire les icônes en forme de cercle). Les étiquettes sont dérivées comme suit :

- Nom du service/de l'application exécuté en termes génériques
- Si la source est un pod :
  - L'étiquette est dérivée de l'étiquette de charge de travail du pod
  - Étiquette attendue sur la charge de travail : app.kubernetes.io/component
  - Référence du nom de l'étiquette : https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels/
  - Étiquettes recommandées :

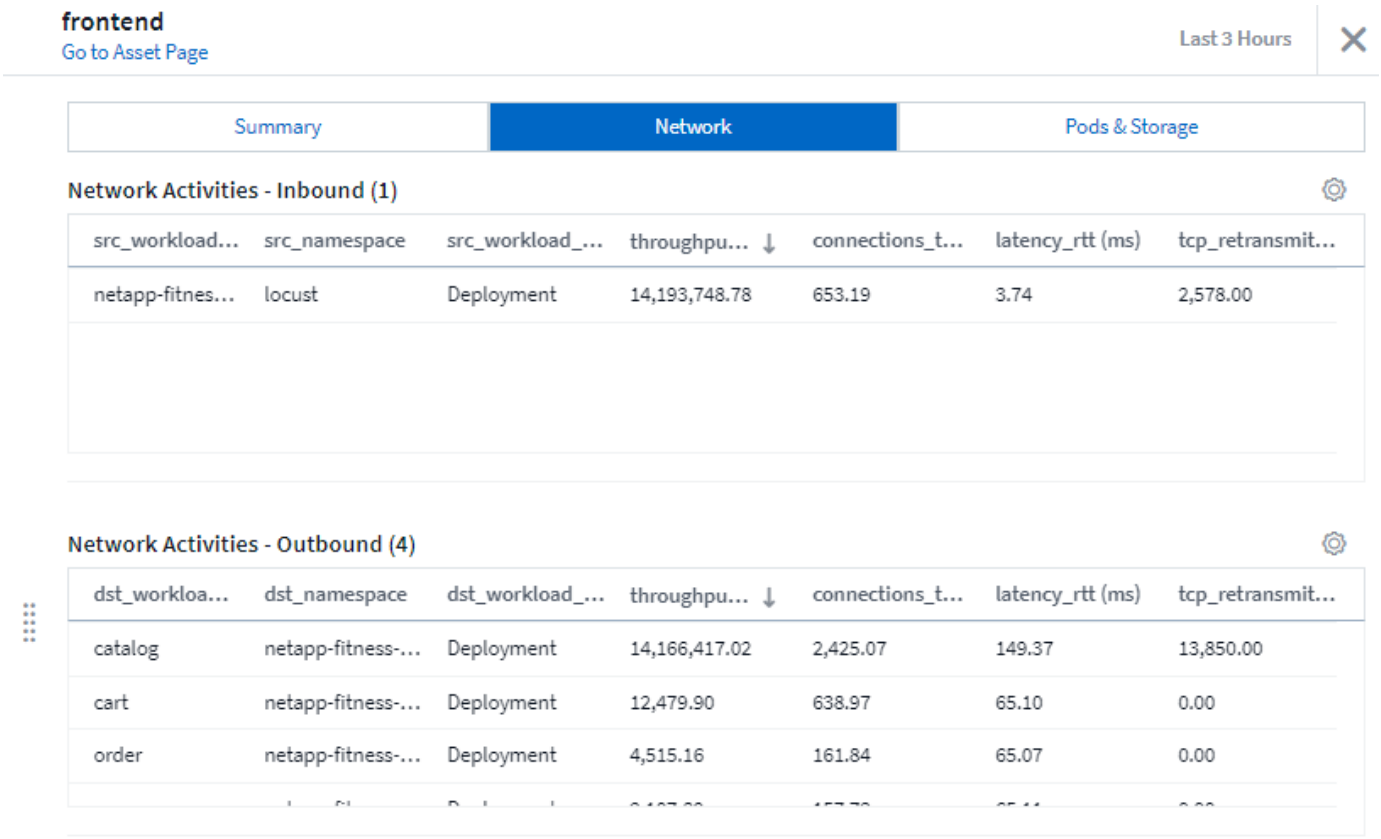
        - l'extrémité avant
        - backend
        - base de données
        - cache
        - file d'attente
        - Kafka
- Si la source est externe au cluster Kubernetes :
    - Data Infrastructure Insights tentera d'analyser le nom résolu DNS pour extraire le type de service.

      Par exemple, avec un nom DNS résolu de *s3.eu-north-1.amazonaws.com*, le nom résolu est analysé pour obtenir *s3* comme type de service.

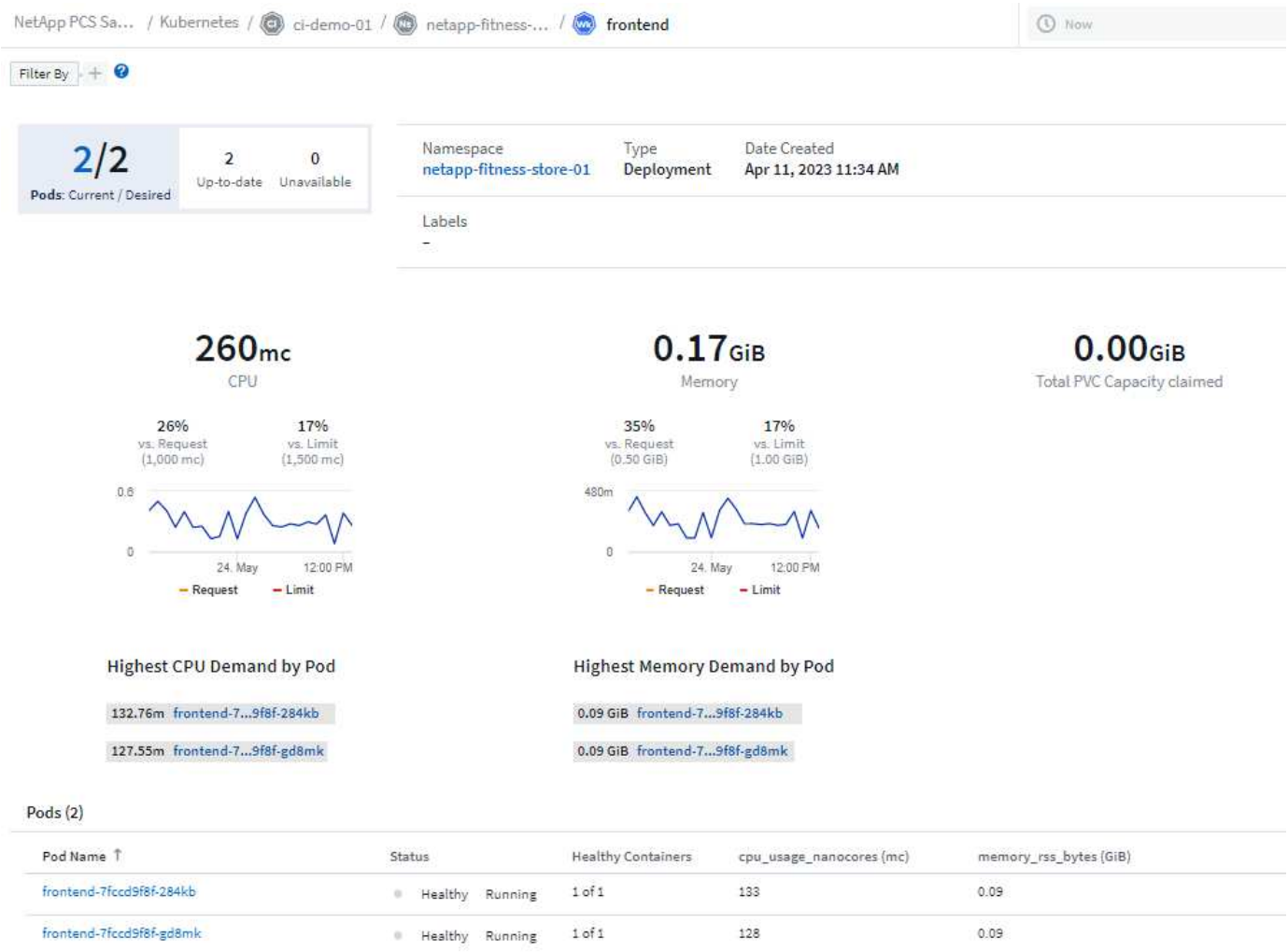
## Plongez en profondeur

Un clic droit sur une charge de travail vous présente des options supplémentaires à explorer plus en détail. Par exemple, à partir d'ici, vous pouvez zoomer pour afficher les connexions pour cette charge de travail.


Vous pouvez également ouvrir le panneau coulissant des détails pour afficher directement l'onglet *Résumé*, *Réseau* ou *Pod et stockage*.




Enfin, la sélection de *Accéder à la page des ressources* ouvrira la page de destination détaillée des ressources pour la charge de travail.



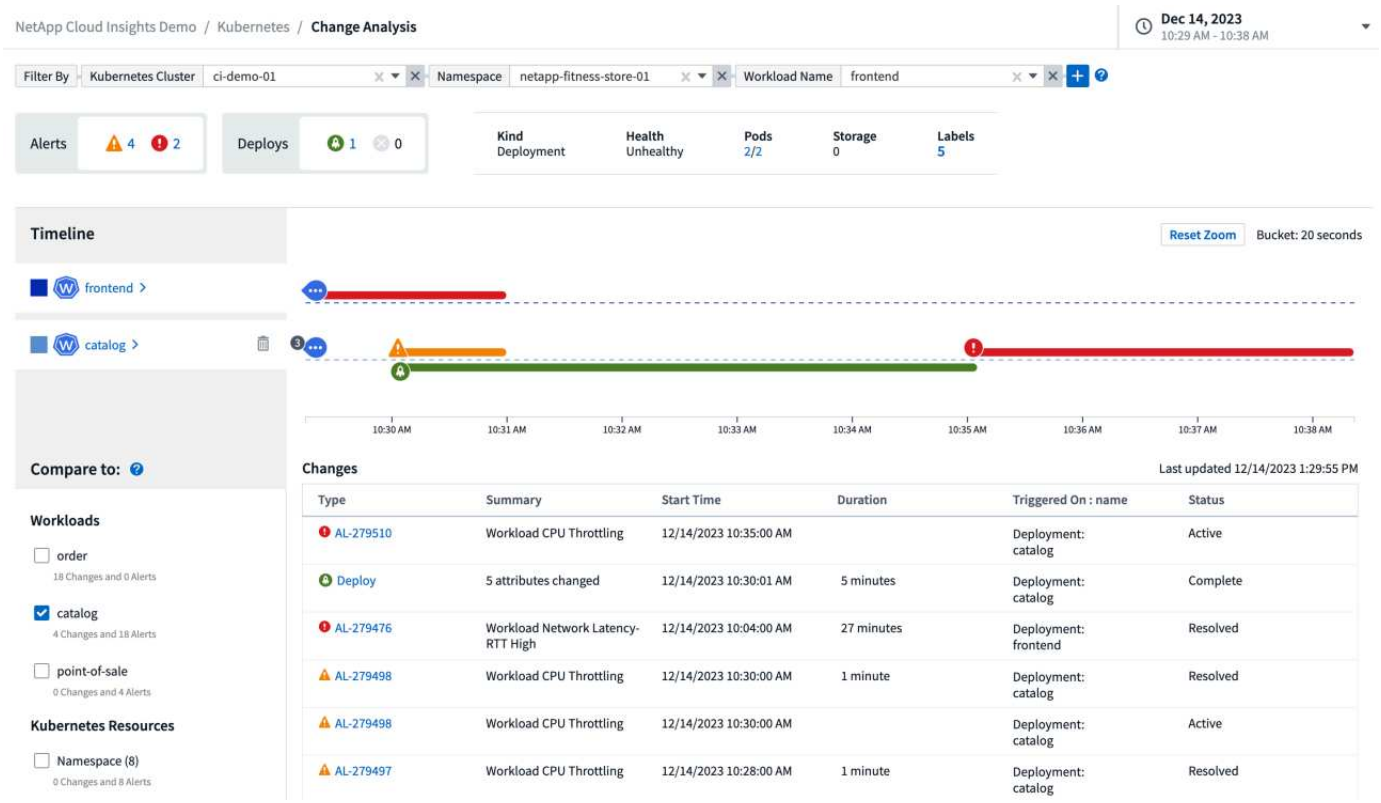

# Analyse des changements Kubernetes

Kubernetes Change Analytics vous offre une vue tout-en-un des modifications récentes apportées à votre environnement K8s. Les alertes et l'état du déploiement sont à portée de main. Avec Change Analytics, vous pouvez suivre chaque changement de déploiement et de configuration et le corréler avec l'état et les performances des services, de l'infrastructure et des clusters K8.

Comment l'analyse du changement aide-t-elle ?

- Dans les environnements Kubernetes multi-locataires, des pannes peuvent survenir en raison de modifications mal configurées. Change Analytics vous aide à y parvenir en fournissant un volet unique pour visualiser et corréler l'état des charges de travail et les modifications de configuration. Cela peut aider à résoudre les problèmes des environnements Kubernetes dynamiques.

Pour afficher Kubernetes Change Analytics, accédez à **Kubernetes > Analyse des changements**.



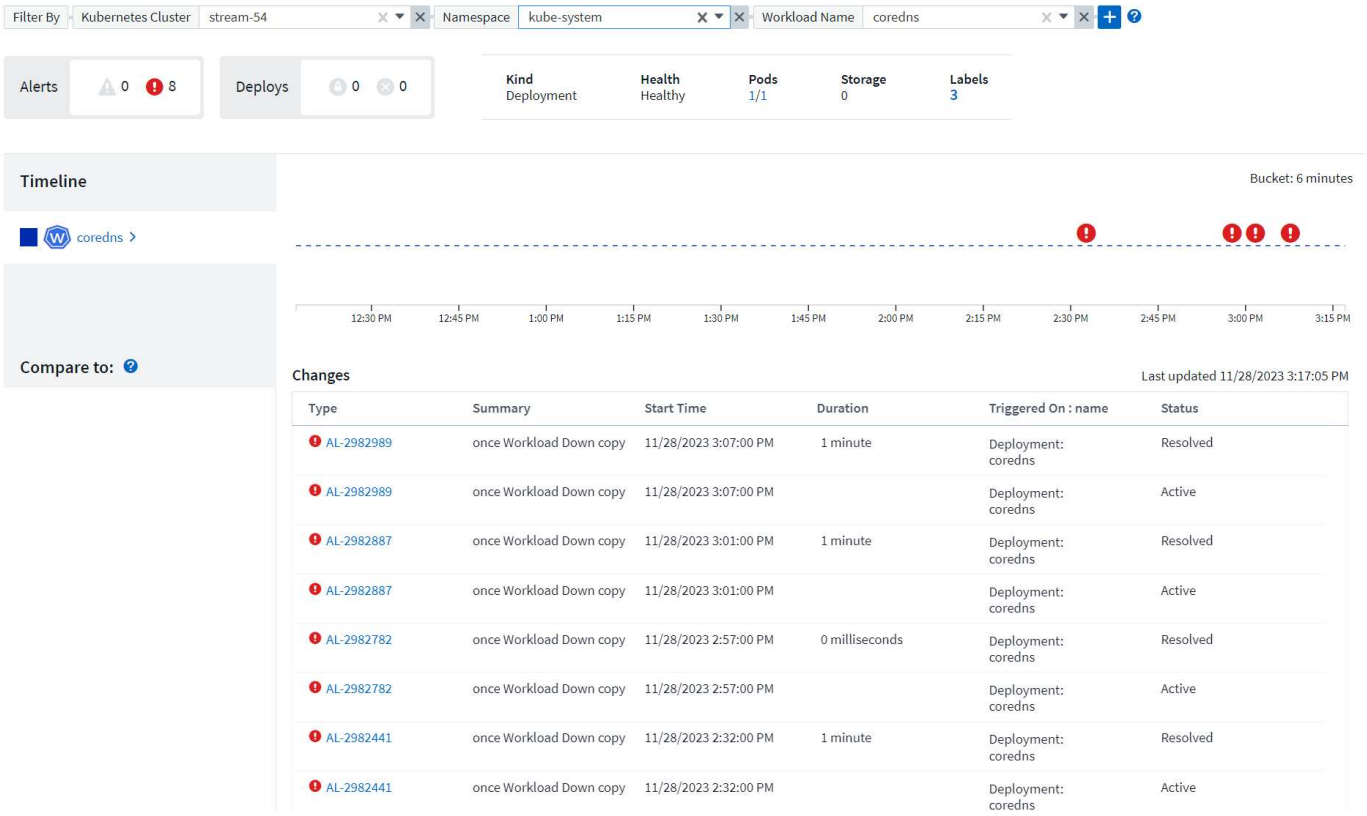

.

La page s’actualise automatiquement en fonction de la plage horaire Data Infrastructure Insights actuellement sélectionnée. Des plages de temps plus courtes signifient un rafraîchissement de l’écran plus fréquent.

## Filtration

Comme pour toutes les fonctionnalités de Data Infrastructure Insights, le filtrage de la liste des modifications est intuitif : en haut de la page, saisissez ou sélectionnez des valeurs pour votre cluster Kubernetes, votre espace de noms ou votre charge de travail, ou ajoutez vos propres filtres en sélectionnant le bouton {+].

Lorsque vous filtrez jusqu’à un cluster, un espace de noms et une charge de travail spécifiques (ainsi que tous les autres filtres que vous définissez), une chronologie des déploiements et des alertes pour cette charge de travail dans cet espace de noms sur ce cluster s’affiche. Effectuez un zoom avant supplémentaire en cliquant et en faisant glisser le graphique pour vous concentrer sur une plage horaire plus spécifique.

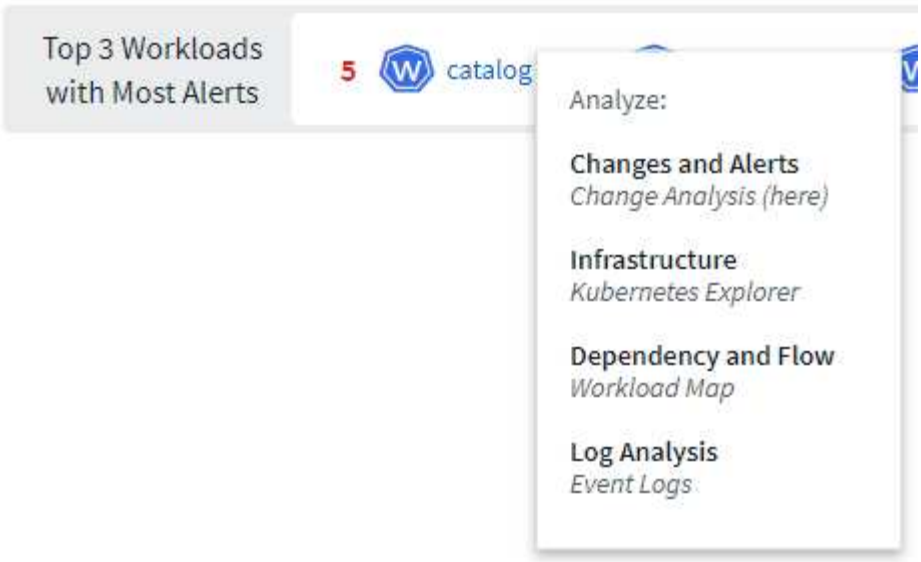



## Statut rapide

Sous la zone de filtrage se trouvent un certain nombre d'indicateurs de haut niveau. Sur la gauche se trouve le nombre d'alertes (Avertissement et Critique). Ce nombre inclut les alertes *Actives* ainsi que *Résolues*. Pour voir uniquement les alertes *Actives*, définissez un filtre pour « Statut » et choisissez « Actif ».

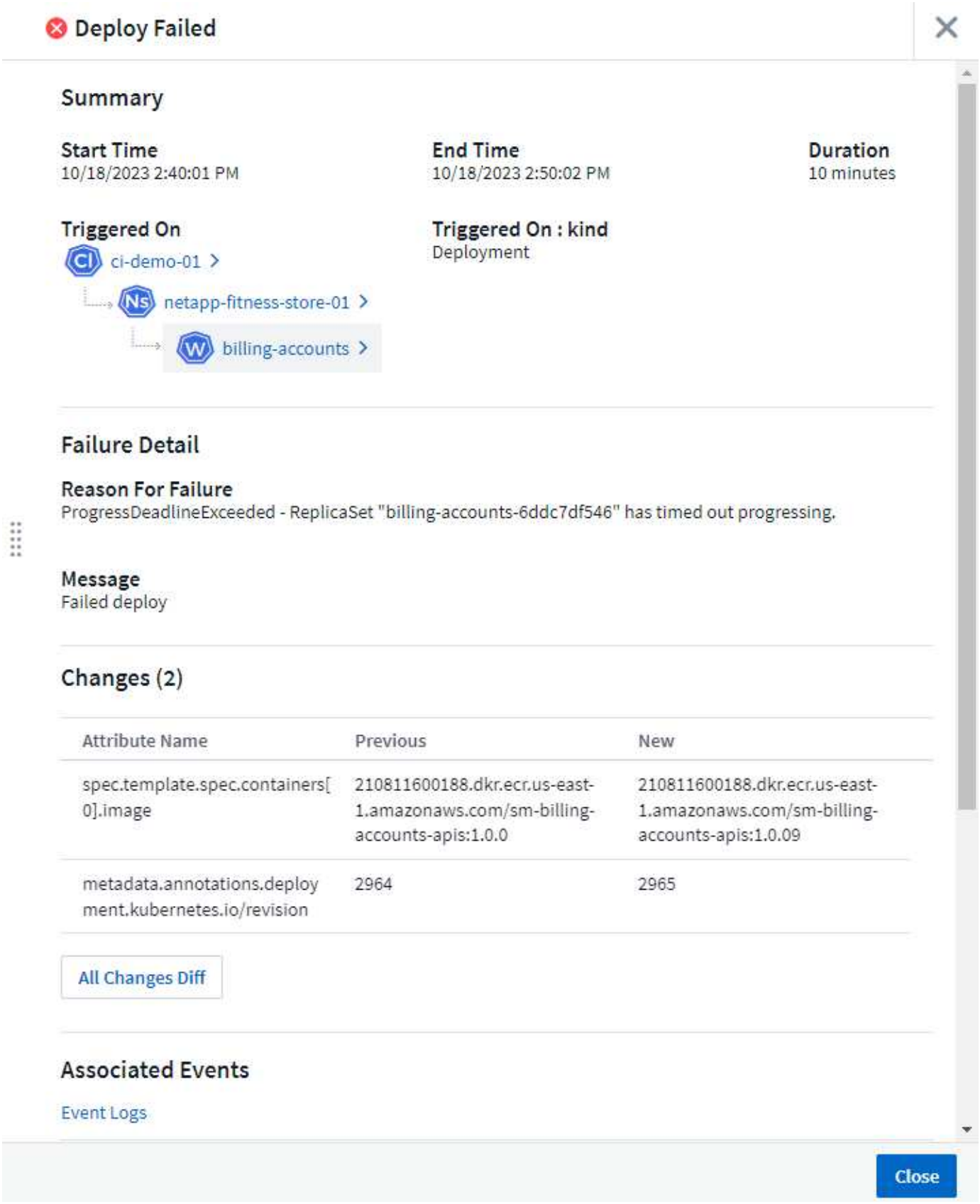

L'état de déploiement est également affiché ici. Encore une fois, la valeur par défaut est d'afficher le nombre de déploiements *Démarrés*, *Terminés* et *Échoués*. Pour voir uniquement les déploiements *Échoués*, définissez un filtre pour « Statut » et sélectionnez « Échoué ».


Les 3 charges de travail avec le plus d'alertes sont les suivantes. Le numéro en rouge à côté de chaque charge de travail indique le nombre d'alertes liées à cette charge de travail. Cliquez sur le lien de charge de travail pour explorer votre infrastructure (Kubernetes Explorer), vos dépendances (carte de charge de travail) ou votre analyse des journaux (journaux des événements).




## Panneau de détails

La sélection d'une modification dans la liste ouvre un panneau décrivant la modification plus en détail. Par exemple, la sélection d'un déploiement ayant échoué affiche un résumé du déploiement, avec les heures de début et de fin, la durée et le lieu où le déploiement a été déclenché, avec des liens pour explorer ces ressources. Il affiche également la raison de l'échec, les modifications associées et tous les événements associés.




La sélection d'une alerte fournit également des détails sur l'alerte, y compris le moniteur qui a déclenché l'alerte ainsi qu'un graphique affichant une chronologie visuelle de l'alerte.