Kubernetes

Data Infrastructure Insights

NetApp
January 17, 2025

Sommaire

Kubernetes

Présentation du cluster Kubernetes

L'explorateur Kubernetes de Data Infrastructure Insights est un outil puissant pour afficher l'état et l'utilisation généraux de vos clusters Kubernetes. Il vous permet d'explorer facilement les domaines d'enquête.

Cliquez sur **tableaux de bord > Kubernetes Explorer** pour ouvrir la page liste des clusters Kubernetes. Cette page de présentation contient le tableau des clusters Kubernetes de votre locataire.

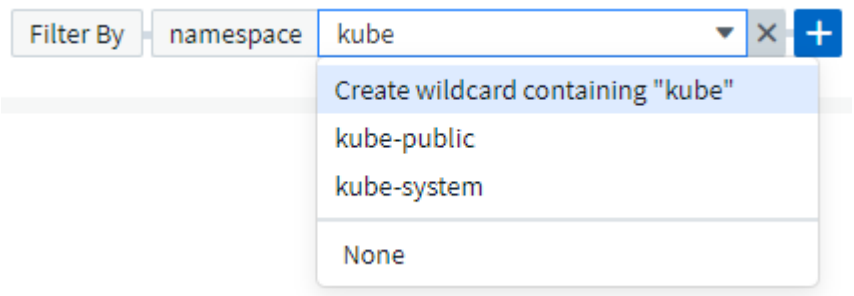
Liste des clusters

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

- Nom * du cluster. Si vous cliquez sur le nom d'un cluster, le système s'ouvrira"page de détails".
- **Pourcentages de saturation**. La saturation globale est la saturation totale la plus élevée du processeur, de la mémoire ou du stockage.
- Nombre de **nœuds** dans le cluster. Cliquez sur ce numéro pour ouvrir la page liste des nœuds.
- Nombre de **Pods** dans le cluster. Cliquez sur ce numéro pour ouvrir la page de liste Pod.
- Nombre de **espaces de noms** dans le cluster. Cliquez sur ce numéro pour ouvrir la page de la liste d'espace de noms.
- Nombre de **charges de travail** dans le cluster. Cliquez sur ce numéro pour ouvrir la page de la liste des charges de travail.

Raffinage du filtre

Lorsque vous filtrez, lorsque vous commencez à taper, vous avez la possibilité de créer un **filtre générique** basé sur le texte en cours. Si vous sélectionnez cette option, tous les résultats correspondant à l'expression de caractère générique seront résélectionnés. Vous pouvez également créer **expressions** à l'aide DE NOT ou ET, ou sélectionner l'option "aucun" pour filtrer les valeurs nulles dans le champ.



Filtres basés sur des caractères génériques ou des expressions (par exemple NON, ET, « aucun », etc.) s'affichent en bleu foncé dans le champ du filtre. Les éléments que vous sélectionnez directement dans la liste s'affichent en bleu clair.

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

Notez que le filtrage des caractères génériques et des expressions fonctionne avec du texte ou des listes, mais pas avec des valeurs numériques, des dates ou des valeurs booléennes.

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 "Opérateur de surveillance Kubernetes".

Composant	Conditions requises
Version Kubernetes	Kubernetes v1.20 et versions ultérieures.
Distributions Kubernetes	AWS Elastic Kubernetes Service (EKS) Azure Kubernetes Service (AKS) Google Kubernetes Engine (GKE) Red Hat OpenShift Rancher Kubernetes Engine (RKE) VMware Tanzu
Linux	Data Infrastructure Insights ne prend pas en charge les nœuds qui s'exécutent avec l'architecture Arm64. Surveillance réseau : doit exécuter le noyau Linux version 4.18.0 ou ultérieure. Le système d'exploitation de photons n'est pas pris en charge.
Étiquettes	Les informations d'infrastructure de données prennent en charge la surveillance des nœuds Kubernetes exécutant Linux, en spécifiant un sélecteur de nœuds Kubernetes qui recherche les étiquettes Kubernetes suivantes sur ces plateformes : Kubernetes v1.20 et versions ultérieures : Kubernetes.io/os = linux Rancher + bétail.io comme plateforme d'orchestration/Kubernetes : bétail.io/os = linux
Commandes	Les commandes curl et kubectl doivent être disponibles.; pour de meilleurs résultats, ajoutez ces commandes au CHEMIN.

Composant	Conditions requises
Connectivité	Kubectl cli est configuré pour communiquer avec le cluster K8s cible et dispose d'une connectivité Internet à votre environnement Data Infrastructure Insights. Si vous êtes derrière un proxy pendant l'installation, suivez les instructions de "Configuration du support de proxy"la section installation de l'opérateur. Pour un audit et un reporting précis des données, synchronisez l'heure sur la machine Agent à l'aide du protocole NTP (Network Time Protocol) ou SNTP (simple Network Time Protocol).
Autre	Si vous utilisez OpenShift 4.6 ou une version supérieure, vous devez suivre le en plus de vous "Instructions OpenShift"assurer que ces conditions préalables sont remplies.
Jeton d'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 d'API ; vous pouvez réutiliser le jeton précédent.

Points importants à noter avant de commencer

Si vous référentiel personnaliséutilisez un proxy, utilisez un , ou OpenShift, lisez attentivement les sections suivantes.

Lisez également à propos de Autorisations.

Configuration du support de proxy

Vous pouvez utiliser un proxy sur votre locataire à deux endroits pour installer l'opérateur de surveillance Kubernetes NetApp. Il peut s'agir de systèmes proxy identiques ou distincts :

- Proxy nécessaire lors de l'exécution de l'extrait de code d'installation (à l'aide de « curl ») pour connecter le système sur lequel l'extrait de code 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'une ou l'autre de ces versions, ou les deux, pour installer le moniteur d'exploitation NetApp Kubernetes, vous devez d'abord vous assurer que votre proxy est configuré pour permettre une bonne communication avec votre environnement Data Infrastructure Insights. Par exemple, dans les serveurs/machines virtuelles depuis lesquels vous souhaitez installer l'opérateur, vous devez être en mesure d'accéder aux informations exploitables de l'infrastructure de données et de télécharger des binaires depuis Data Infrastructure Insights.

Pour le proxy utilisé pour installer le moniteur d'exploitation NetApp Kubernetes, définissez les variables d'environnement *http_proxy/https_proxy* avant d'installer l'opérateur. Pour certains environnements proxy, il peut être nécessaire de définir la variable *no_proxy Environment*.

Pour définir la ou les variables, effectuez les opérations 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 n'a 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 la
commande suivante :
```

```
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 l'opérateur de surveillance Kubernetes NetApp après avoir lu toutes ces instructions.

Configurez la section proxy de 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
  ...
...
```

À l'aide d'un référentiel docker personnalisé ou privé

Par défaut, l'opérateur de surveillance Kubernetes NetApp extrait les images du conteneur du référentiel informations exploitables de l'infrastructure de données. Si vous utilisez un cluster Kubernetes comme cible de surveillance et que ce cluster est configuré pour extraire uniquement les 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 requis par l'opérateur de surveillance NetApp Kubernetes.

Exécutez l'extrait de code image dans la mosaïque d'installation de NetApp Monitoring Operator. Cette commande permet de se connecter au référentiel Data Infrastructure Insights, d'extraire toutes les dépendances d'image pour l'opérateur et de se déconnecter du référentiel Data Infrastructure Insights. Lorsque vous y êtes invité, saisissez le mot de passe temporaire du référentiel fourni. Cette commande permet de télécharger toutes les images utilisées par l'opérateur, y compris pour les fonctions facultatives. Voir ci-dessous pour connaître les caractéristiques auxquelles ces images sont utilisées.

Fonctionnalités centrales de l'opérateur et surveillance Kubernetes

- surveillance netapp
- proxy kube-rbac
- metrics-état-kube
- telegraf
- utilisateur-root-distroless

Journal des événements

- fluent-bit
- exportateur-événements-kubernetes

Performances et carte réseau

- ci-net-observateur

Envoyez l'image de docker de l'opérateur à votre référentiel docker privé, local ou d'entreprise, conformément aux règles de votre entreprise. Assurez-vous que les balises d'image et les chemins de répertoire vers 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 la configuration d'agentConfiguration dans Operator-config.yaml pour refléter le nouvel emplacement docker repo. Créez une nouvelle imagePullSecret pour votre référentiel privé. Pour plus de détails, voir *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 exécutez sur OpenShift 4.6 ou une version ultérieure, vous devez modifier la configuration d'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 contrôlez contient des ressources personnalisées qui n'ont pas de ClusterRole "agrégats à afficher", vous devez 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 les ressources souhaités avec les verbes ["get", "Watch", "list"]. Voir https://kubernetes.io/docs/reference/access-authn-authz/rbac/
3. Appliquez vos modifications au cluster

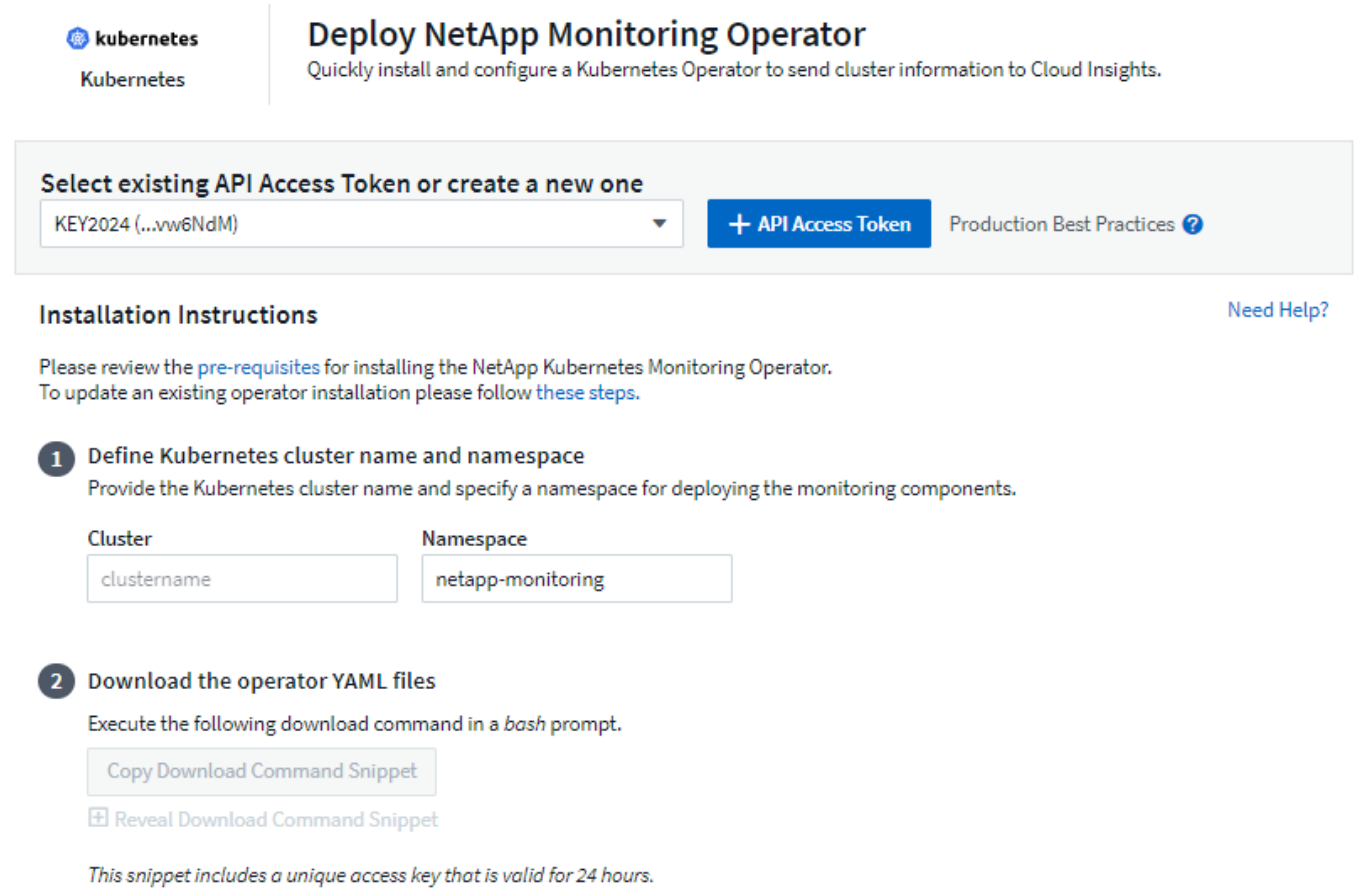
Installation et configuration de l'opérateur de contrôle Kubernetes

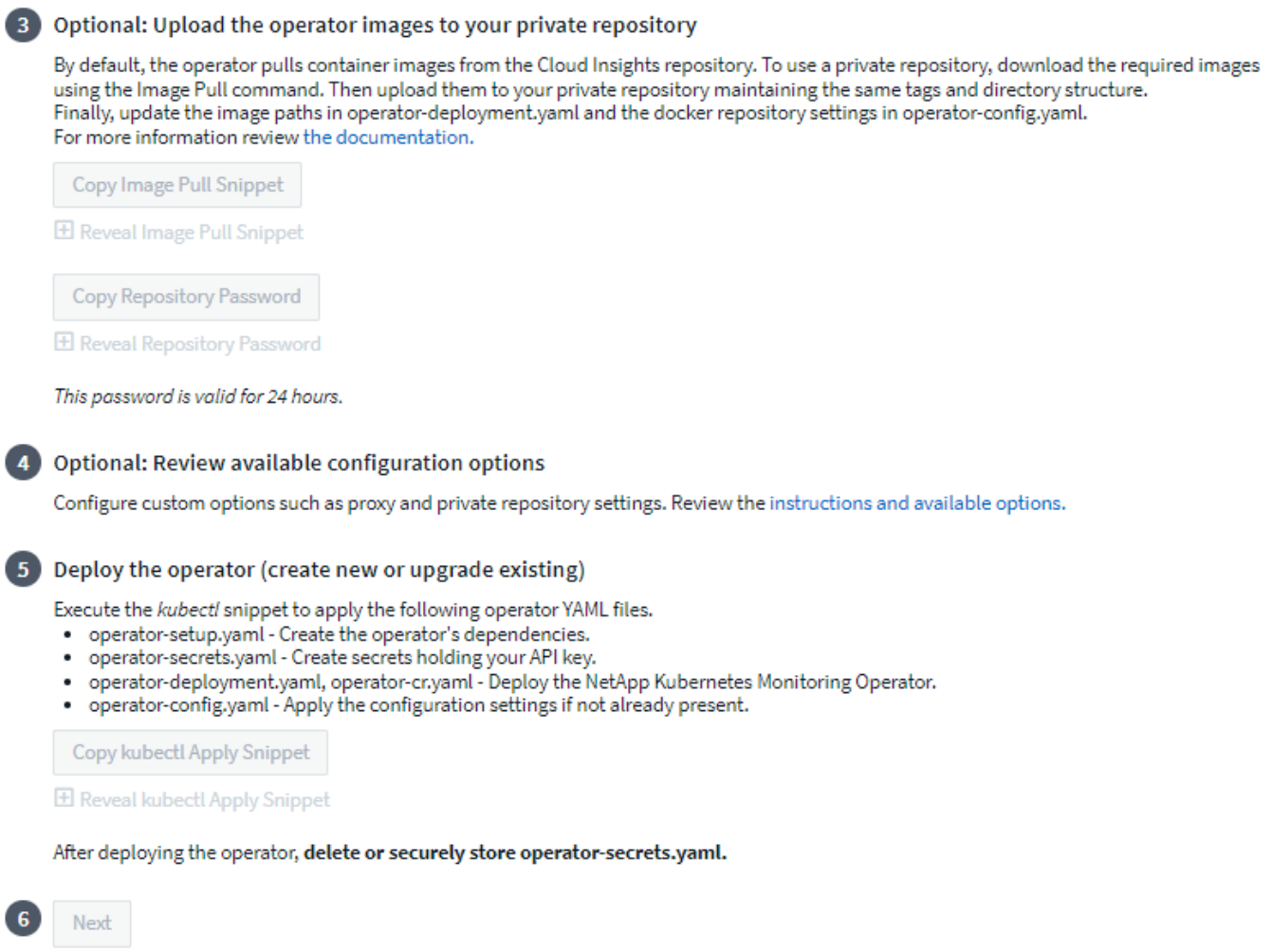
Data Infrastructure Insights propose l'opérateur **Kubernetes Monitoring Operator** pour la collecte Kubernetes. Accédez à **Kubernetes > Collectors > +Kubernetes Collector** pour déployer un nouvel opérateur.

Avant d'installer l'opérateur de surveillance Kubernetes

Consultez "Conditions préalables"la documentation avant d'installer ou de mettre à niveau l'opérateur de surveillance Kubernetes.

Installation de l'opérateur de surveillance Kubernetes





Étapes d'installation de l'agent opérateur de surveillance Kubernetes sur Kubernetes :

1. Entrez un nom de cluster et un espace de noms uniques. Si vous mise à niveauutilisez un opérateur Kubernetes précédent, utilisez le même nom de cluster et le même espace de noms.
2. Une fois ces données saisies, vous pouvez copier le fragment de commande de téléchargement dans le presse-papiers.
3. Collez le fragment dans une fenêtre *bash* et exécutez-le. Les fichiers d'installation de l'opérateur seront téléchargés. Notez que l'extrait de code possède une clé unique et est valide pendant 24 heures.
4. Si vous disposez d'un référentiel personnalisé ou privé, copiez le fragment facultatif image Pull, 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, passez en revue 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 le fragment kubectl Apply, en le téléchargeant et en l'exécutant.
7. L'installation se poursuit 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 utilisez un proxy, lisez à propos de configuration du proxy.

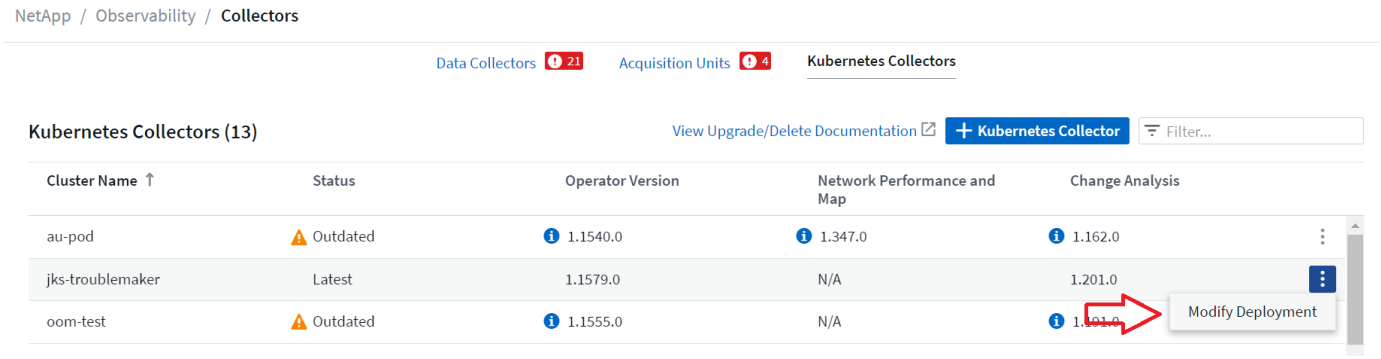
Si vous disposez d'un référentiel personnalisé, lisez à propos de à l'aide d'un référentiel docker personnalisé/privé.

Composants de surveillance Kubernetes

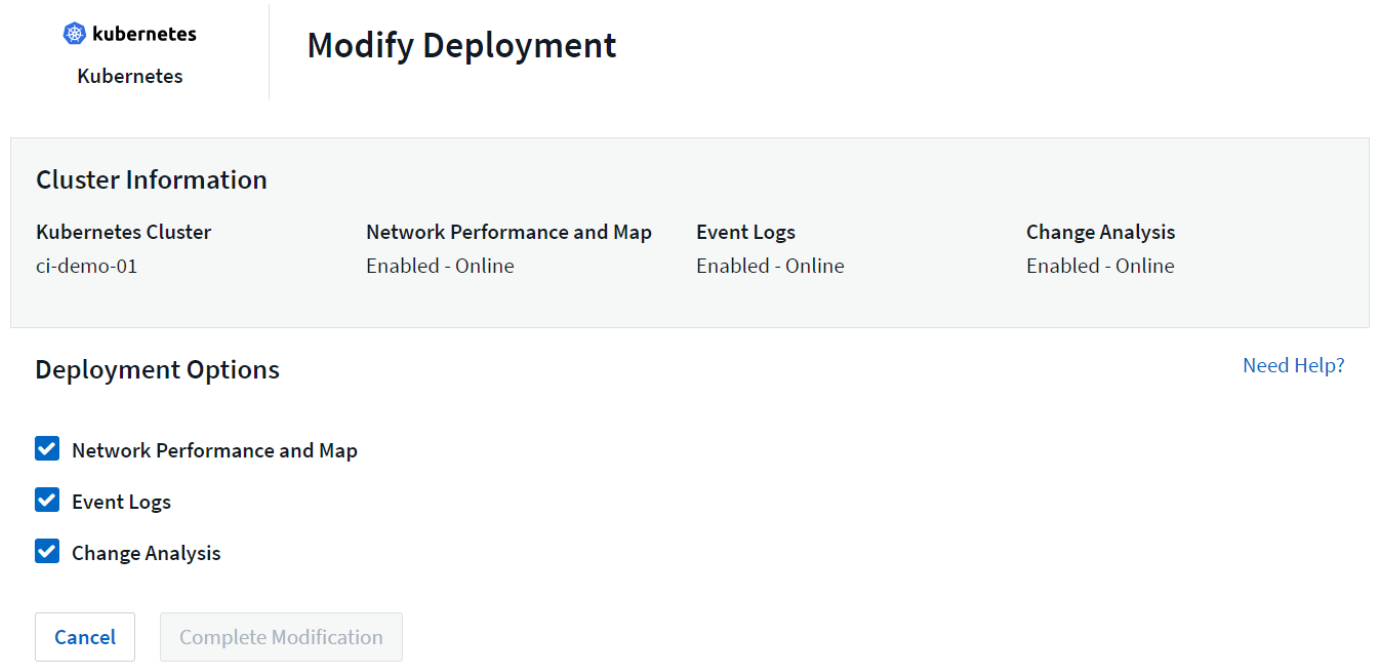
La surveillance Kubernetes de Data Infrastructure Insights comprend quatre composants de surveillance :

- Metrics du cluster
- Carte et performances réseau (en option)
- Journaux d'événements (facultatif)
- Analyse des modifications (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 > Collectors** et en sélectionnant *Modify Deployment* 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 pour ce collecteur selon vos besoins.



Mise à niveau vers le dernier opérateur de surveillance Kubernetes

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

```
 kubectl -n netapp-monitoring get agentconfiguration netapp-monitoring-
configuration
Si une configuration d'agentConfiguration existe :
```

- Installez Dernier opérateur par rapport à l'opérateur existant.
  - Assurez-vous que vous extraction des dernières images du conteneurutilisez un référentiel personnalisé.

Si AgentConfiguration n'existe pas :

- Notez le nom de votre cluster tel qu'il a été reconnu par les informations d'infrastructure de données (si votre namespace n'est pas le contrôle NetApp par défaut, remplacez l'espace de noms approprié) :

```
 kubectl -n netapp-monitoring get agent -o
jsonpath='{.items[0].spec.cluster-name}'
* Créer une sauvegarde de l'opérateur existant (si votre namespace n'est
pas la surveillance netapp par défaut, remplacez le namespace 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,Installez>> Le dernier
opérateur.
```

  - Utilisez le même nom de cluster.
  - Après avoir téléchargé les derniers fichiers Operator YAML, porter toutes les personnalisations trouvées dans agent_backup.yaml à l'opérateur-config.yaml téléchargé avant le déploiement.
  - Assurez-vous que vous extraction des dernières images du conteneurutilisez 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 de l'opérateur de surveillance Kubernetes est « netapp-monitoring ». Si vous avez défini votre propre espace de noms, remplacez-le dans ces commandes et tous les fichiers suivants.

Les nouvelles versions de l'opérateur de surveillance peuvent être désinstallées à l'aide des 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, supprimer l'espace de noms :

```
 kubectl delete ns <NAMESPACE>
Si la première commande renvoie "aucune ressource trouvée", suivez les
instructions ci-dessous pour désinstaller les anciennes versions de
l'opérateur de surveillance.
```

Exécutez chacune des commandes suivantes dans l'ordre indiqué. 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 des indicateurs Kube-State

L'opérateur de surveillance NetApp Kubernetes installe ses propres metrics kube-State pour éviter les conflits avec d'autres instances.

Pour plus d'informations sur Kube-State-Metrics, reportez-vous à "cette page"la section .

Configuration/personnalisation de l'opérateur

Ces sections contiennent des informations sur la personnalisation de la configuration de votre opérateur, l'utilisation du 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 contient des exemples de paramètres commentés. Voir la liste des pour la version la plus récente de "paramètres disponibles"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 du support de proxy

Vous pouvez utiliser un proxy sur votre locataire à deux endroits pour installer l'opérateur Kubernetes Monitoring. Il peut s'agir de systèmes proxy identiques ou distincts :

- Proxy nécessaire lors de l'exécution de l'extrait de code d'installation (à l'aide de « curl ») pour connecter le système sur lequel l'extrait de code 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'une ou l'autre de ces opérations, ou pour les deux, 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 que vous 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 le moniteur d'exploitation Kubernetes, avant d'installer l'opérateur, définissez les variables d'environnement *http_proxy/https_proxy*. Pour certains environnements proxy, il peut être nécessaire de définir la variable *no_proxy Environment*.

Pour définir la ou les variable(s), effectuez les opérations suivantes sur votre système **avant** installation de l'opérateur de surveillance Kubernetes :

1. Définissez les variables d'environnement *https_proxy* et/ou *http_proxy* pour l'utilisateur actuel :
   a. Si le proxy en cours de configuration n'a 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 la
commande suivante :
```

```
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 l'opérateur de surveillance Kubernetes après avoir lu toutes ces instructions.

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

À l'aide d'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 utilisez un cluster Kubernetes comme cible pour la surveillance et que ce cluster est configuré pour extraire uniquement les 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 requis par l'opérateur de surveillance Kubernetes.

Exécutez l'extrait de code image dans la mosaïque d'installation de NetApp Monitoring Operator. Cette commande permet de se connecter au référentiel Data Infrastructure Insights, d'extraire toutes les dépendances d'image pour l'opérateur et de se déconnecter du référentiel Data Infrastructure Insights. Lorsque vous y êtes invité, saisissez le mot de passe temporaire du référentiel fourni. Cette commande permet de télécharger toutes les images utilisées par l'opérateur, y compris pour les fonctions facultatives. Voir ci-dessous pour connaître les caractéristiques auxquelles ces images sont utilisées.

Fonctionnalités centrales de l'opérateur et surveillance Kubernetes

- surveillance netapp
- proxy ci-kube-rbac
- ci-ksm
- ci-telegraf
- utilisateur-root-distroless

Journal des événements

- bit fluide ci

- ci-kubernetes-exportateur-événements

Performances et carte réseau

- ci-net-observateur

Envoyez l'image de docker de l'opérateur à votre référentiel docker privé, local ou d'entreprise, conformément aux règles de votre entreprise. Assurez-vous que les balises d'image et les chemins de répertoire vers 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 la configuration d'agentConfiguration dans Operator-config.yaml pour refléter le nouvel emplacement docker repo. Créez une nouvelle imagePullSecret pour votre référentiel privé. Pour plus de détails, voir *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
```

Instructions OpenShift

Si vous exécutez sur OpenShift 4.6 ou une version ultérieure, vous devez modifier la configuration d'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 tainations

Les *netapp-ci-telegraf-ds*, *netapp-ci-Fluent-bit-ds* et *netapp-ci-net-observateur-l4-ds* Demonsets 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 certains **taints** bien connus. Si vous avez configuré des fichiers d'accès personnalisés sur vos nœuds, empêchant ainsi les modules de s'exécuter sur chaque nœud, vous pouvez créer une **tolérance** pour ces fichiers d'accès "Dans *AgentConfiguration*". Si vous avez appliqué des rejets personnalisés 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 opérateur.

En savoir plus sur Kubernetes "Teintes et tolérances".

Revenir au **"Page installation de l'opérateur de surveillance NetApp Kubernetes"**

Remarque sur les secrets

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

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

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'option analyse des modifications est activée, modifiez *AgentConfiguration* ou *Operator-config.yaml* pour annuler le commentaire de la section de gestion des modifications et incluez *kindsToIgnoreFromWatch: '"secrets"'* dans la section de gestion des modifications. 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'images 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 co-signer 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 : télécharger les images de l'opérateur dans votre référentiel privé > clé publique de signature d'image*

Pour vérifier manuellement une signature d'image, effectuez les opérations suivantes :

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 du cosigne. Reportez-vous à l'exemple suivant d'utilisation des coenseignes

```
$ 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

Voici quelques points à essayer en cas de problème lors de la configuration de l'opérateur de surveillance Kubernetes :

Problème :	**Essayer :**
Je ne vois pas de lien hypertexte/connexion entre mon volume persistant Kubernetes et le périphérique de stockage back-end correspondant. Mon volume persistant Kubernetes est configuré en utilisant le nom d'hôte du serveur de stockage.	Procédez comme suit pour désinstaller l'agent Telegraf existant, puis réinstaller l'agent Telegraf le plus récent. Vous devez utiliser Telegraf version 2.0 ou ultérieure et le stockage de votre cluster Kubernetes doit être activement surveillé par Data Infrastructure Insights.

Problème :	Essayer :
Je vois des messages dans les journaux qui ressemblent à ce qui suit : E0901 15 352:21:39.962145 178 1 Reflector.Go:178] k8s.io/kube-state-metrics/Internal/store/Builder.Go:43.168161 : échec de la liste *v1.MutatingWebhookio Configuration : le serveur n'a pas pu trouver la ressource demandée E0901 15:21/352/Reflector.s.Go.so	Ces messages peuvent se produire si vous exécutez des metrics d'état kube version 2.0.0 ou supérieure avec les versions Kubernetes inférieures à 1.20. Pour obtenir la version Kubernetes : *kubectl version* pour obtenir la version kube-state-metrics : *kubectl get deployment/kube-state-metrics -o jsonpath='{..image}'* pour éviter que ces messages se produisent, les utilisateurs peuvent modifier leur déploiement de metrics kube-state-metrics pour désactiver les baux suivants : *hookingwebconfigurations*. Ressources=certificats,demandes persistantes,configmaps,cronjobs,demonets, déploiements,noeuds finaux,horizontalepodpodscalers,ingresources,details, resuts,undats,depositionsstatees,depositigmats,defiees,resottes,depositionssecuts,defiees,dees,depositionunedats,delimantees,delimantees,deficedats,dees,delimantees,delimantees,delimantees,deficedats,delimantees,deficedats,delimantees,deficedats,deficedats,dees,delimantees,delimantees,dees,delimantees,deficedats,dees,delimantees,delimantees,delimantees,delimantees,de vaillewebconfiguration,v'
Je vois des messages d'erreur de Telegraf ressemblant aux messages suivants, mais Telegraf démarre et s'exécute : oct 11 14:23:41 ip-172-31-39-47 systemd[1] : lancé l'agent serveur piloté par des plug-ins pour signaler des mesures dans InfluxDB. Oct 11 14:23:41 ip-172-31-39-47 telegraf[1827] : heure="2021-10-11T14:23:41Z" level=erreur msg="Impossible de créer le répertoire de cache. /Etc/telegraf/.cache/flocon de neige, err : mkdir /etc/telegraf/.ca che : permission refusée. Ignored\n » func="nowgosflake.(*defaultLogger).Errorf" file="log.Go:10" Oct 1827 23:2021:39-47 ip-172-31-41 telegraf[11 14] : échec de l'ouverture:23:120. Ignoré. Ouvrir /etc/telegraf/.cache/flocon/ocsp_Response_cache.json : pas de fichier ou répertoire\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.Go:120 10" Oct 23:2021:39-47 ip-1827-31 telegraf[172]: 23-41-11 14:11Z! Démarrage de Telegraf 1.19.3	Il s'agit d'un problème connu. Voir "Article GitHub"pour plus de détails. Tant que Telegraf est opérationnel, les utilisateurs peuvent ignorer ces messages d'erreur.
Sur Kubernetes, mes coffee pad(s) Telegraf ont signalé l'erreur suivante : "erreur lors du traitement des informations de mountstats : échec de l'ouverture du fichier mountstats: /Hostfs/proc/1/mountstats, erreur: Ouvrir /hostfs/proc/1/mountstats: Permission refusée"	Si SELinux est activé et applique, il empêche probablement le ou les pod(s) Telegraf d'accéder au fichier /proc/1/mountstats sur le nœud Kubernetes. Pour contourner cette restriction, modifiez la configuration d'agentconfiguration et activez le paramètre runPrivileged. Pour plus de détails, reportez-vous au "Instructions OpenShift".

Problème :	Essayer :
Sur Kubernetes, mon pod Telegraf ReplicaSet signale l'erreur suivante : [inputs.prometheus] erreur dans le plug-in : impossible de charger keypair /etc/kubernetes/pki/ETcd/Server.crt:/etc/kubernetes/pki/ETcd/Server.key : ouvrir /etc/kubernetes/pki/ETcd/Server.crt : aucun fichier ni répertoire	Le pod Télégraf ReplicaSet est conçu pour 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 obtenez ces erreurs. Vérifiez si vos nœuds maître/ETCD ont des astuces sur eux. S'ils le font, ajoutez les tolérances nécessaires à Telegraf ReplicaSet, telegraf-RS. Par exemple, modifiez le ReplicaSet… kubectl edit RS telegraf-RS …et ajoutez les tolérances appropriées à la spécification. Redémarrez ensuite 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 règle de sécurité Pod (PSP) ou l'admission de sécurité Pod (PSA) sur place, vous devez effectuer la mise à niveau vers l'opérateur de surveillance Kubernetes le plus récent. Procédez comme suit pour mettre à niveau vers l'opérateur actuel avec la prise en charge de PSP/PSA : 1. Désinstaller le précédent opérateur de surveillance : kubectl delete agent-monitoring-NetApp -n NetApp-monitoring kubectl delete ns NetApp-monitoring kubectl delete crd agents.monitoring.NetApp.com kubectl delete clusterrole agent-manager-role agent-proxy-role agent-metrics-reader kubectl delete clusterrolebinding agent-manager-rolebinding agent-cluster-agent-roleadmin-binding-cluster-2-agent-binding. Installez dernière version de l'opérateur de surveillance.
J'ai rencontré des problèmes lors de la tentative de déploiement de l'opérateur, et j'ai utilisé PSP/PSA.	1. Modifiez l'agent à l'aide de la commande suivante : kubectl -n <name-space> edit agent 2. Marquez « Security-policy-enabled » comme « false ». Ceci désactivera les stratégies de sécurité du Pod et l'admission de sécurité du Pod et permettra à l'opérateur de déployer. Confirmez en utilisant les commandes suivantes : kubectl get psp (devrait afficher Pod Security Policy supprimé) kubectl get all -n <namespace>
grep -i psp (doit montrer que rien n'a été trouvé)	Erreurs « ImagePullBackoff » détectées
Ces erreurs peuvent se produire si vous disposez d'un référentiel docker personnalisé ou privé et que vous n'avez pas encore configuré l'opérateur de surveillance Kubernetes pour qu'il le reconnaisse correctement. En savoir plus a propos de la configuration pour 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 :	Essayer :

Capturer ou noter le résultat des commandes suivantes et contacter l'équipe de support technique.
`kubectl -n netapp-monitoring get all`
`kubectl -n netapp-monitoring describe all`
`kubectl -n netapp-monitoring logs <monitoring-operator-pod> --all-containers=true`
`kubectl -n netapp-monitoring logs <telegraf-pod> --all-containers=true`
	Les pods net-observateur (Workload Map) de l'espace de noms de l'opérateur se trouvent dans CrashLoopBackOff

Ces pods correspondent au collecteur de données Workload Map pour l'observabilité réseau. Essayez : • Vérifiez les journaux de l'un des modules pour confirmer la version minimale du noyau. Par exemple : --- {"ci-tenant-ID":"votre-tenant-ID","collectionneur-cluster":"votre-k8s-cluster-name","Environment":"prod","level":"error","msg":"échec de la validation. Raison : la version 3.10.0 du noyau est inférieure à la version minimale du noyau de 4.18.0","Time":"2022-11-09T08:23:08Z"} ---- • les pods Net-observateur requièrent 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'ils sont >= 4.18.0	Les pods s'exécutent dans l'espace de noms Operator (par défaut : surveillance netapp), mais aucune donnée n'est affichée dans l'interface pour la carte des workloads ou les metrics Kubernetes dans les requêtes
Vérifiez le réglage de l'heure sur les nœuds du cluster K8S. Pour un audit et un reporting précis des données, il est vivement recommandé de synchroniser l'heure sur l'ordinateur de l'agent à l'aide du protocole NTP (Network Time Protocol) ou SNTP (simple Network Time Protocol).	Certains des pods net-observateur dans l'espace de noms de l'opérateur sont à l'état en attente
Net-observateur est un DemonSet et exécute un pod dans chaque nœud du cluster k8s. • Notez le pod qui est à l'état en attente et vérifiez s'il rencontre un problème de ressource 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 l'opérateur de surveillance Kubernetes : [inputs.prometheus] erreur dans le plug-in : erreur lors de la demande HTTP vers http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics : get http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics : Dial tcp: kube-state-metrics.<namespace>.svc.cluster.local : pas de recherche d'hôte

Problème :	Essayer :
Ce message n'apparaît généralement que lorsqu'un nouvel opérateur est installé et que le module *telegraf-RS* est en marche avant que le module *ksm* ne soit en marche. Ces messages doivent s'arrêter une fois que tous les modules sont en cours d'exécution.	Je ne vois aucun indicateur collecté pour les cronjobs Kubernetes qui existent dans mon cluster.
Vérifiez votre version de Kubernetes (c'est-à-dire `kubectl version`). S'il est v1.20.x ou inférieur, il s'agit d'une limitation attendue. La version de kube-state-metrics déployée avec l'opérateur de surveillance Kubernetes ne prend en charge que v1.cronjob. Avec Kubernetes 1.20.x et versions antérieures, la ressource cronjob est à v1beta.cronjob. Par conséquent, les indicateurs d'état kube ne peuvent pas trouver la ressource cronjob.	Après l'installation de l'opérateur, les modules telegraf-ds entrent dans CrashLoopBackOff et les journaux du pod indiquent « su: Authentication failure ».
Modifiez la section telegraf dans *AgentConfiguration* et définissez *dockerMetricCollectionEnabled* sur FALSE. Pour plus de détails, reportez-vous au "options de configuration"manuel de l'opérateur . … spec: … telegraf: … - Nom: docker run-mode : - DemonSet substitutions: - Key: DOCKER_UNIX_SOCK_PLACEHOLDER valeur: unix:///run/docker.sock … …	Je vois des messages d'erreur récurrents ressemblant à ce qui suit dans mes journaux Telegraf: E! [Agent] erreur d'écriture dans outputs.http: Post "https://<tenant_url>/REST/v1/Lake/iningt/influxdb": Délai de contexte dépassé (client. Dépassement du délai d'attente des en-têtes)
Modifiez la section telegraf dans *AgentConfiguration* et augmentez *outputTimeout* à 10 s. Pour plus de détails, reportez-vous au "options de configuration"manuel de l'opérateur .	Il me manque des données *involvedobject* pour certains journaux d'événements.
Assurez-vous d'avoir suivi les étapes de la "Autorisations" section ci-dessus.	Pourquoi deux modules d'opérateurs de surveillance s'exécutent, l'un nommé netapp-ci-monitoring-Operator-<pod> et l'autre Monitoring-Operator-<pod> ?
Depuis le 12 octobre 2023, Data Infrastructure Insights a été décidé de réorganiser l'opérateur pour mieux répondre aux besoins de nos utilisateurs. Pour que ces changements soient entièrement adoptés, vous devez retirez l'ancien opérateur et installez le nouveau.	Mes événements kubernetes ont cessé de générer des rapports à Data Infrastructure Insights de manière inattendue.
Récupérer le nom du pod Event-exportateur : `` `kubectl -n netapp-monitoring get pods ``	grep event-exporter

Problème :	Essayer :
awk '{print $1}'	sed 's/event-exporter./event-exporter/'` Il doit être « netapp-ci-event-exportatrice » ou « event-exportatrice ». Ensuite, modifiez l'agent de surveillance `kubectl -n netapp-monitoring edit agent` et définissez la valeur de LOG_FILE pour qu'elle reflète 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-exportatrice.log » ou «/var/log/containers/event-exportatrice*.log ». …. fluent-bit: … - name: event-exporter-ci substitutions: - key: LOG_FILE values: - /var/log/containers/netapp-ci-event-exporter*.log … …. Sinon, on peut aussi désinstaller et réinstallez l'agent.
J'constate que le ou les pods déployés par l'opérateur de surveillance Kubernetes sont en panne en raison de ressources insuffisantes.	Reportez-vous à l'opérateur de surveillance Kubernetes "options de configuration"pour augmenter les limites de processeur et/ou de mémoire selon les besoins.
Si une image manquante ou une configuration non valide a entraîné l'échec du démarrage ou de la préparation des pods de metrics d'état de netapp-ci-kube. L'état StatefulSet est bloqué et les modifications de configuration ne sont pas appliquées aux pods de metrics netapp-ci-kube-state.	StatefulSet est dans un "cassé" état. Après avoir résolu tout problème de configuration, utilisez les pods de metrics netapp-ci-kube-état.
les pods de metrics d'état-ci-kube-netapp ne parviennent pas à démarrer après l'exécution d'une mise à niveau d'opérateur Kubernetes, et lancent ErrImagePull (échec de l'extraction de l'image).	Essayez de réinitialiser les modules manuellement.
Des messages « événement ignoré comme étant plus ancien que maxEventAgeSeconds » sont observés pour mon cluster Kubernetes sous analyse du journal.	Modifiez l'opérateur *agentconfiguration* et augmentez les valeurs *event-exportatrice-maxEventAgeSeconds* (c.-à-d. à 60 s), *event-exportatrice-kubeQPS* (c.-à-d. à 100) et *event-exportatrice-kubeBurst* (c.-à-d. à 500). Pour plus de détails sur ces options de configuration, reportez-vous à la "options de configuration" page.

Problème :	Essayer :
Telegraf avertit ou se bloque en raison 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'agentNKMO et définissez *Unprotected* sur *true*. Cela indique 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 disque, il permet une exécution dans des environnements où il est impossible de réserver de la mémoire verrouillée. Pour plus de détails sur les options de configuration *Unprotected*, 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 lors de la lecture de /dev/vdc : pas de fichier ou de répertoire	Pour l'opérateur de surveillance Kubernetes, ces messages d'avertissement sont bénins et peuvent être ignorés en toute sécurité. Vous pouvez également modifier la section telegraf dans AgentConfiguration et définir *runDsPrivileged* sur TRUE. Pour plus de détails, reportez-vous au "options de configuration de l'opérateur".

Problème :	Essayer :

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

Problème :	Essayer :
Les pods telegraf DS signalent des erreurs liées au plug-in d’entrée kubernetes qui ne parviennent pas à faire de requêtes HTTP en raison de l’incapacité à valider le certificat TLS. Par exemple : E! [Inputs.kubernetes] erreur dans le plug-in : erreur lors de la demande HTTP pour "<a href="https://<kubelet_IP>:10250/stats/summary": " class="bare">https://<kubelet_IP>:10250/stats/summary": obtenir "<a href="https://<kubelet_IP>:10250/stats/summary": " class="bare">https://<kubelet_IP>:10250/stats/summary": tls : échec de la vérification du certificat : x509 : impossible de valider le certificat pour <kubelet_IP> car il ne contient pas de 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 des certificats *Subject alternative Name*. Pour résoudre ce problème, l'utilisateur peut modifier le "configuration de l'agent", et définir *telegraf:insecureK8sSkipVerify* sur *true*. Cela va configurer le plug-in d'entrée telegraf pour ignorer la vérification. Sinon, l'utilisateur peut configurer le kubelet pour "ServerTLSBootstrap", qui déclenchera une demande de certificat à partir de l'API 'certificates.k8s.io'.

Des informations supplémentaires sont disponibles sur la "Assistance" page ou dans le "Matrice de prise en charge du Data Collector".

Options de configuration de l'opérateur de surveillance Kubernetes

La "Opérateur de surveillance Kubernetes" configuration peut être personnalisée.

Le tableau ci-dessous répertorie les options possibles pour le fichier *AgentConfiguration* :

Composant	Option	Description
agent		Options de configuration communes à tous les composants que l'opérateur peut installer. Ces options peuvent être considérées comme des options « globales ».
	DockerRepo	Remplacement dockerRepo pour extraire des images des repos docker privés des clients par rapport à Data Infrastructure Insights docker repo. La valeur par défaut est Data Infrastructure Insights docker repo
	DockerImagePullSecret	Facultatif : un secret pour les clients repo privés
	Nom du cluster	Champ de texte libre qui identifie de manière unique un cluster dans tous les clusters de clients. Cette fonctionnalité doit être unique dans un locataire Data Infrastructure Insights. La valeur par défaut correspond à ce que le client saisit dans l'interface utilisateur pour le champ Cluster Name

Composant	Option	Description
	Proxy format: Proxy: Server: Port: Nom d'utilisateur: Mot de passe: NoProxy: IsTelegrafProxyEnabled: IsAuProxyEnabled: IsFluentbitProxyEnabled: IsCollectorProxyEnabled:	Facultatif pour définir le proxy. Il s'agit généralement du proxy d'entreprise du client.
telegraf		Options de configuration qui peuvent personnaliser l'installation de telegraf de l'opérateur
	Intervalle de collection	Intervalle de collecte des metrics, en secondes (max=60 s)
	DsCpuLimit	Limite CPU pour les telegraf ds
	DsMemLimit	Limite de mémoire pour les télégraf
	DsCpuRequest	Demande CPU pour les telegraf ds
	DsMemRequest	Demande de mémoire pour les télégraf
	RsCpuLimit	Limite CPU pour les RS telegraf
	RsMemLimit	Limite de mémoire pour les RS telegraf
	RsCpuRequest	Demande CPU pour les RS telegraf
	RsMemRequest	Demande de mémoire pour les RS telegraf
	Avec privilèges d'exécution	Exécutez le conteneur *telegraf-mountstats-poller* de telegraf DemonSet en mode privilégié. Définissez cette valeur sur true si SELinux est activé sur vos nœuds Kubernetes.
	RunDsPrivileged	Définissez runDsPrivileged sur true pour exécuter le conteneur telegraf de telegraf DemonSet en mode privilégié.
	Taille de la batchSize	Voir "Documentation de configuration de Telegraf"
	BufferLimit	Voir "Documentation de configuration de Telegraf"
	Interval	Voir "Documentation de configuration de Telegraf"
	CollectionJitter	Voir "Documentation de configuration de Telegraf"
	précision	Voir "Documentation de configuration de Telegraf"
	Intervalle de rinçage	Voir "Documentation de configuration de Telegraf"
	Scintillement	Voir "Documentation de configuration de Telegraf"
	OutputTimeout	Voir "Documentation de configuration de Telegraf"
	Tolérances de type dsTolerations	télégraf-ds tolérances supplémentaires.
	RsTolerations	tolérances supplémentaires de telegraf-rs.

Composant	Option	Description
	SkipProcessorsAfterAggregators	Voir "Documentation de configuration de Telegraf"
	non protégé	ee ce "Problème connu de Telegraf". Le paramètre *Unprotected* indique à l'opérateur de surveillance Kubernetes d'exécuter Telegraf avec l' `--unprotected`indicateur.
	insecureK8sSkipVerify	Si telegraf ne parvient pas à vérifier le certificat en raison de l'absence de SAN IP, essayez d'activer le saut de vérification
metrics-état-kube		Options de configuration permettant de personnaliser l'installation des mesures d'état kube de l'opérateur
	CpuLimit	Limite CPU pour le déploiement des indicateurs d'état kube
	MemLimit	Limite MEM pour le déploiement de mesures kube-state
	CpuRequest	Demande de processeur pour le déploiement de metrics d'état kube
	MemRequest	Demande MEM pour le déploiement de mesures d'état kube
	ressources	une liste de ressources séparées par des virgules à capturer. exemple : cronjobs,demonsets,déploiements,ingresses,travaux, espaces de noms,nœuds,perstentvolumeclaims, perstentvolumes,pods,réplicasets,resourcetas,service s,statefulsets
	tolérances	tolérances supplémentaires des indicateurs d'état kube.
	étiquettes	une liste de ressources séparées par des virgules que les indicateurs d'état kube doivent capturer + exemple : cronjobs=**[],demonsets=[**],deployments=**[],ingresses=[**],jobs=**[],namespaces=[**],nodes=**[],resenteclaims=[**],despotstatsets**[][][]**[],desp odeslotstatedespods=**[][][]**
journaux		Options de configuration permettant de personnaliser la collecte des journaux et l'installation de l'opérateur
	ReadFromHead	vrai/faux, doit couramment lire le journal à partir de la tête
	délai dépassé	délai, en secondes
	DnsMode	TCP/UDP, mode pour DNS
	tolérances fluentes-bit	fluent-bit-ds tolérances supplémentaires.
	tolérance-exportateur-événement	tolérances supplémentaires de l'exportateur d'événements.

Composant	Option	Description
	Event-exportateur-maxEventAgeSeconds	âge max. de l'événement de l'exportateur d'événement. Voir https://github.com/jkroepke/resmoio-kubernetes-event-exporter
mappe des charges de travail		Options de configuration permettant de personnaliser la collection de cartes de charge de travail et l'installation de l'opérateur.
	CpuLimit	Limite CPU pour net observateur ds
	MemLimit	limite mem pour les observateurs nets
	CpuRequest	Demande CPU pour net observateur ds
	MemRequest	demande mem pour net observateur ds
	Intervalle d'Aggregationde métadonnées	intervalle d'agrégation des mesures, en secondes
	BpfPollInterval	Intervalle d'interrogation BPF, en secondes
	EnableDNSLookup	Vrai/faux, activer la recherche DNS
	tolérances l4	net-observateur-l4-ds tolérances supplémentaires.
	Avec privilèges d'exécution	True/FALSE : définissez runPrivileged sur TRUE si SELinux est activé sur vos nœuds Kubernetes.
gestion des modifications		Options de configuration de Kubernetes change Management and Analysis
	CpuLimit	Limite CPU pour change-observateur-Watch-RS
	MemLimit	Limite MEM pour change-observateur-Watch-RS
	CpuRequest	Demande CPU pour change-observateur-Watch-RS
	MemRequest	demande mem pour changement-observateur-watch-rs
	FailureDeclationIntervalMins	Intervalle en minutes après lequel un déploiement non réussi d'une charge de travail sera marqué comme ayant échoué
	DeployAggrIntervalSeconds	Fréquence à laquelle les événements de déploiement de charge de travail en cours sont envoyés
	Non WorkloadAggrIntervalSeconds	Fréquence à laquelle les déploiements sans charge de travail sont combinés et envoyés
	TermsToRedact	Un ensemble d'expressions régulières utilisées dans les noms env et les cartes de données dont la valeur sera bifftée. Exemples de termes : « pwd », « mot de passe », « jeton », « apikey », « api-key », « jwt »
	AdditionalKindsToWatch	Liste séparée par des virgules de types supplémentaires à surveiller par rapport à l'ensemble de types par défaut surveillés par le collecteur

Composant	Option	Description
	KindsToIgnoreFromWatch	Liste de types séparés par une virgule à ignorer de l'ensemble de types par défaut surveillés par le collecteur
	LogRecordAggrIntervalSeconds	Fréquence à laquelle les enregistrements de journal sont envoyés à l'EC à partir du collecteur
	tolérances de surveillance	change-observateur-watch-ds tolérances supplémentaires. Format abrégé à une seule ligne uniquement. Exemple : '{key: Taint1, operator: Exists, effect: NoSchedule},{key: Taint2, operator: Exists, effect: NoExecute}'

Exemple de fichier AgentConfiguration

Vous trouverez ci-dessous un exemple de fichier *AgentConfiguration*.

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

spec:
  # # You can modify the following fields to configure the operator.
  # # Optional settings are commented out and include default values for
reference
  # #   To update them, uncomment the line, change the value, and apply
the updated AgentConfiguration.
  agent:
    # # [Required Field] A uniquely identifiable user-friendly
clustername.
    # # clusterName must be unique across all clusters in your Data
Infrastructure Insights environment.
    clusterName: "my_cluster"

    # # Proxy settings. The proxy that the operator should use to send
metrics to Data Infrastructure Insights.
    # # Please see documentation here: https://docs.netapp.com/us-
en/cloudinsights/task_config_telegraf_agent_k8s.html#configuring-proxy-
support
    # proxy:
    #   server:
    #   port:
    #   noproxy:
```

```
    #   username:
    #   password:
    #   isTelegrafProxyEnabled:
    #   isFluentbitProxyEnabled:
    #   isCollectorsProxyEnabled:

    # # [Required Field] By default, the operator uses the CI repository.
    # # To use a private repository, change this field to your repository
name.
    # # Please see documentation here: https://docs.netapp.com/us-
en/cloudinsights/task_config_telegraf_agent_k8s.html#using-a-custom-or-
private-docker-repository
    dockerRepo: 'docker.c01.cloudinsights.netapp.com'
    # # [Required Field] The name of the imagePullSecret for dockerRepo.
    # # If you are using a private repository, change this field from
'netapp-ci-docker' to the name of your secret.
    dockerImagePullSecret: 'netapp-ci-docker'

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

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

    # # The default time telegraf will wait between inputs for all plugins
(interval). Max=60
    # collectionInterval: '60s'
    # # Maximum number of records per output that telegraf will write in
one batch (metric_batch_size).
    # batchSize: '10000'
    # # Maximum number of records per output that telegraf will cache
pending a successful write (metric_buffer_limit).
    # bufferLimit: '150000'
    # # Collect metrics on multiples of interval (round_interval).
    # roundInterval: 'true'
    # # Each plugin waits a random amount of time between the scheduled
collection time and that time + collection_jitter before collecting inputs
(collection_jitter).
```

```
    # collectionJitter: '0s'
    # # Collected metrics are rounded to the precision specified. When set
to "0s" precision will be set by the units specified by interval
(precision).
    # precision: '0s'
    # # Time telegraf will wait between writing outputs (flush_interval).
Max=collectionInterval
    # flushInterval: '60s'
    # # Each output waits a random amount of time between the scheduled
write time and that time + flush_jitter before writing outputs
(flush_jitter).
    # flushJitter: '0s'
    # # Timeout for writing to outputs (timeout).
    # outputTimeout: '5s'

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

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

    # # Skip second run of processors after aggregators
    # skipProcessorsAfterAggregators: 'true'

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


    # If telegraf warns of insufficient lockable memory, try increasing
the limit of lockable memory for Telegraf in the underlying operating
system/node.  If increasing the limit is not an option, set this to true
to instruct Telegraf to not attempt to reserve locked memory pages.  While
```

```
this might pose a security risk as decrypted secrets might be swapped out
to disk, it allows for execution in environments where reserving locked
memory is not possible.
    # unprotected: 'false'

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

    # # event-exporter client-side throttling
    # # Set kubeBurst to roughly match your events per minute and
```

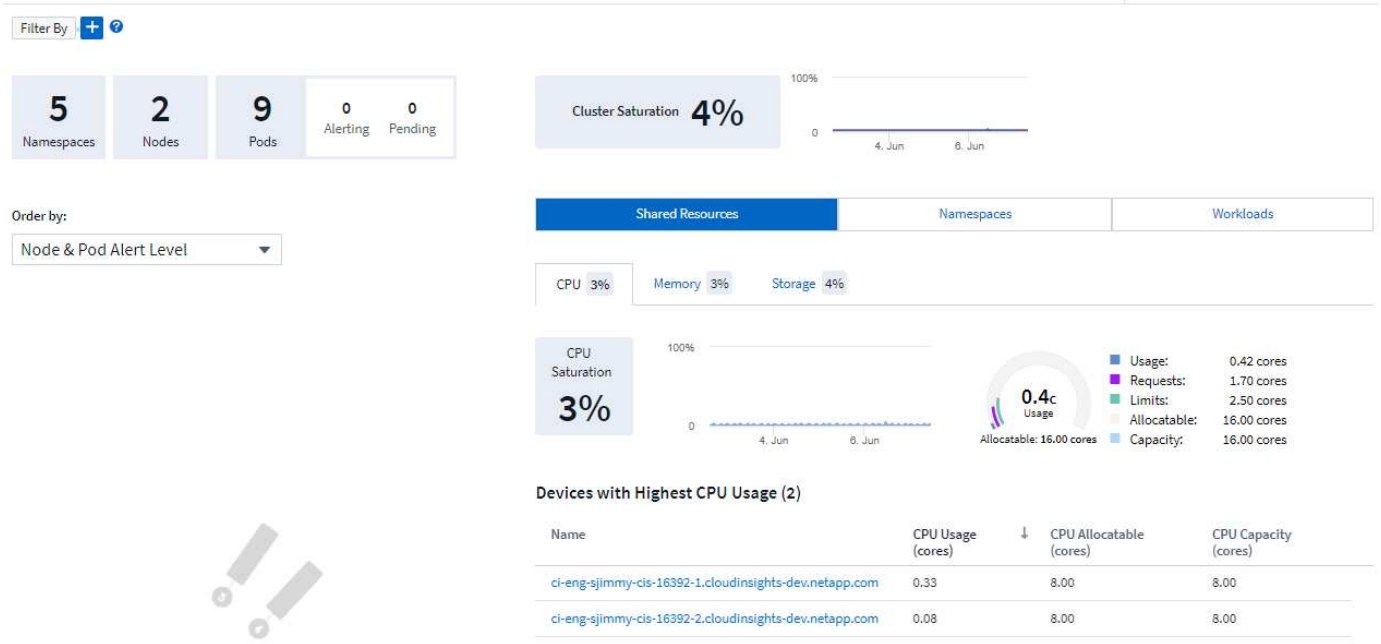
```
kubeQPS=kubeBurst/5
    # # See https://github.com/resmoio/kubernetes-event-
exporter#troubleshoot-events-discarded-warning
    # event-exporter-kubeQPS: 20
    # event-exporter-kubeBurst: 100

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

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

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

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

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

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

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

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

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

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

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

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

    # # A comma separated list of additional kinds to watch from the
default set of kinds watched by the collector
    # # Each kind will have to be prefixed by its apigroup
    # # Example: '"authorization.k8s.io.subjectaccessreviews"'
    # additionalKindsToWatch: ''

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

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

    # # Frequency with which log records are sent to CI from the collector
```

```
    # logRecordAggrIntervalSeconds: '20'

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

Page des détails du cluster Kubernetes

La page des détails du cluster Kubernetes affiche une présentation détaillée du cluster Kubernetes.



L’espace de noms, les nœuds et le Pod

Le nombre en haut de la page indique le nombre total d’espaces de noms, de nœuds et de pods dans le cluster, ainsi que le nombre de fenêtres d’alerte en cours et en attente.

Ressources partagées et saturation

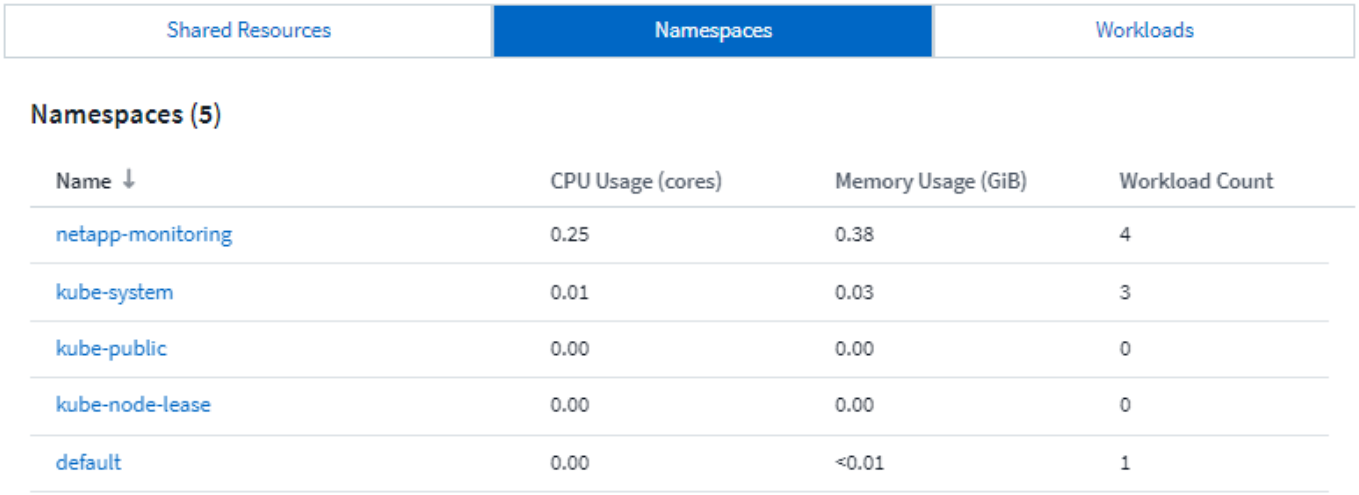
En haut à droite de la page de détails se trouve la saturation de votre cluster en tant que pourcentage actuel, ainsi qu’un graphique affichant la tendance récente dans le temps. La saturation du cluster est la plus élevée du CPU, de la mémoire ou du stockage à chaque point du temps.

Ci-dessous, la page affiche par défaut **Shared Resources**, avec des onglets pour CPU, mémoire et stockage. Chaque onglet affiche le pourcentage de saturation et la tendance dans le temps, avec des détails d’utilisation supplémentaires. Pour le stockage, la valeur indiquée est le plus grand nombre de saturation du système de fichiers back-end et du système de fichiers, qui sont calculés indépendamment.

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

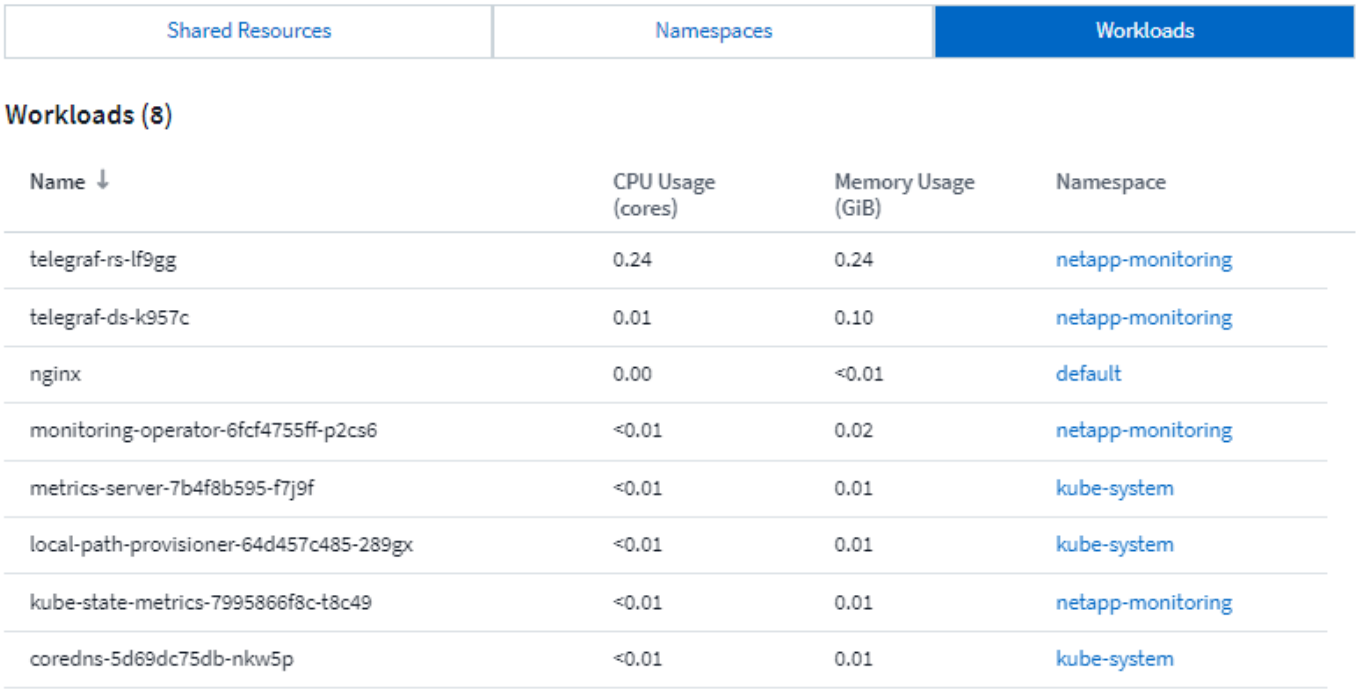
Espaces de noms

L'onglet espaces de noms affiche la liste de tous les espaces de noms de votre environnement Kubernetes, indiquant l'utilisation du CPU et de la mémoire, ainsi que le nombre de workloads dans chaque espace de noms. Cliquez sur les liens Nom pour explorer chaque espace de noms.



Charges de travail

De la même manière, l'onglet charges de travail affiche la liste des charges de travail de chaque namespace, avec pour nouveau l'affichage de l'utilisation du processeur et de la mémoire. Un clic sur les liens de l'espace de noms permet d'accéder à chacun d'eux



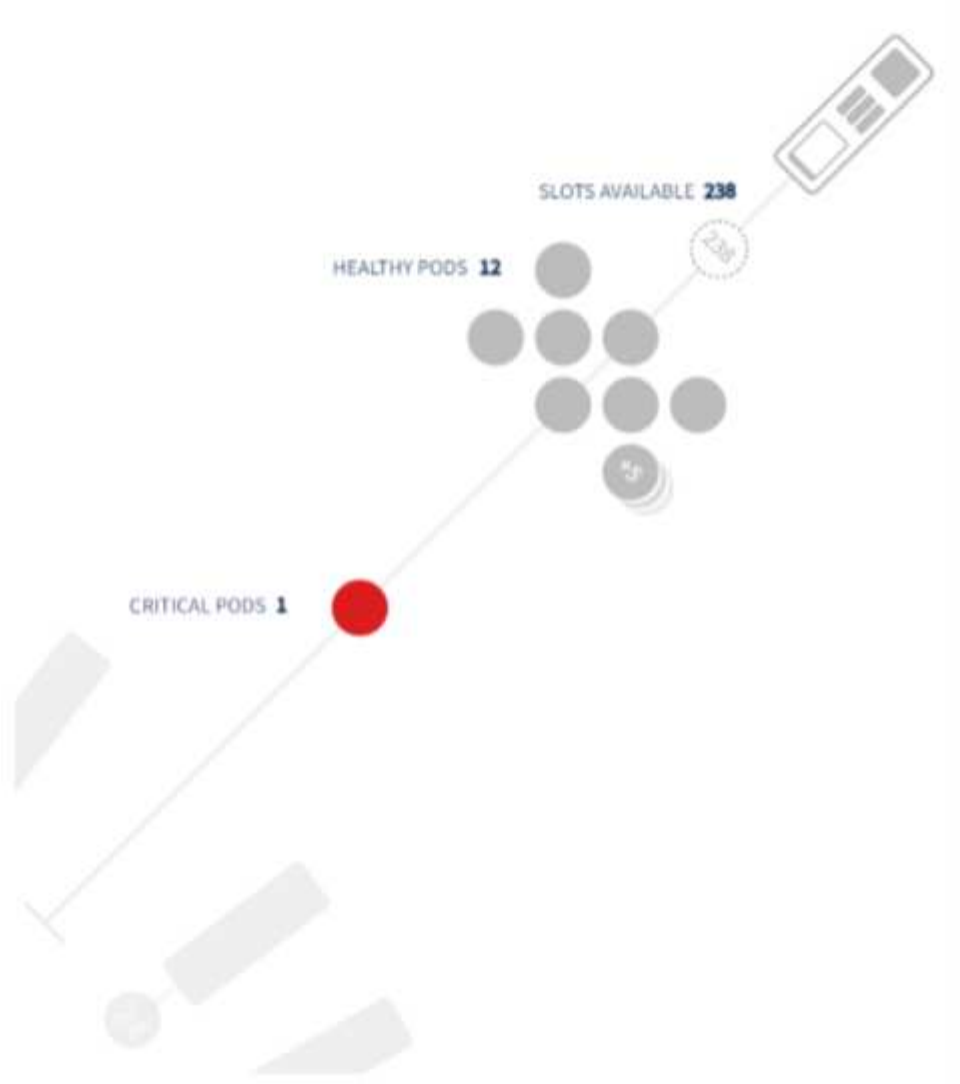
Le groupe d’instruments « roue »

La section roue du cluster fournit un aperçu de l’état des nœuds et des pods pour en savoir plus. 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.

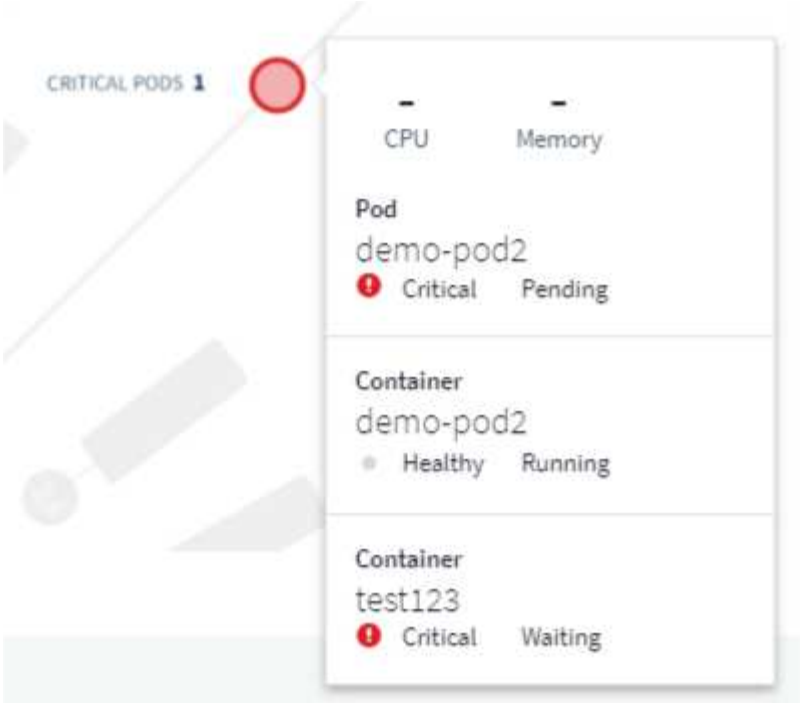
Les pods ou les nœuds d’alerte s’affichent en rouge. Les zones « Avertissement » s’affichent en orange. Les modules qui ne sont pas programmés (c’est-à-dire non connectés) s’affichent dans le coin inférieur du bloc d’instruments.

Le fait de passer la souris sur un pod (cercle) ou un nœud (barre) permet d’étendre la vue du nœud.

Cliquez sur le pod ou le nœud dans cette vue pour effectuer un zoom avant sur la vue développée du nœud.



À partir de là, vous pouvez passer le curseur sur un élément pour afficher des détails sur cet élément. Par exemple, passez la souris sur le pod critique dans cet exemple pour afficher des détails sur ce pod.



Vous pouvez afficher les informations relatives au système de fichiers, à la mémoire et à la CPU en passant le pointeur de la souris sur les éléments du nœud.

Une note sur les jauges

Les jauges mémoire et CPU affichent trois couleurs, puisqu'elles indiquent *used* par rapport à *allocatable Capacity* et *total Capacity*.

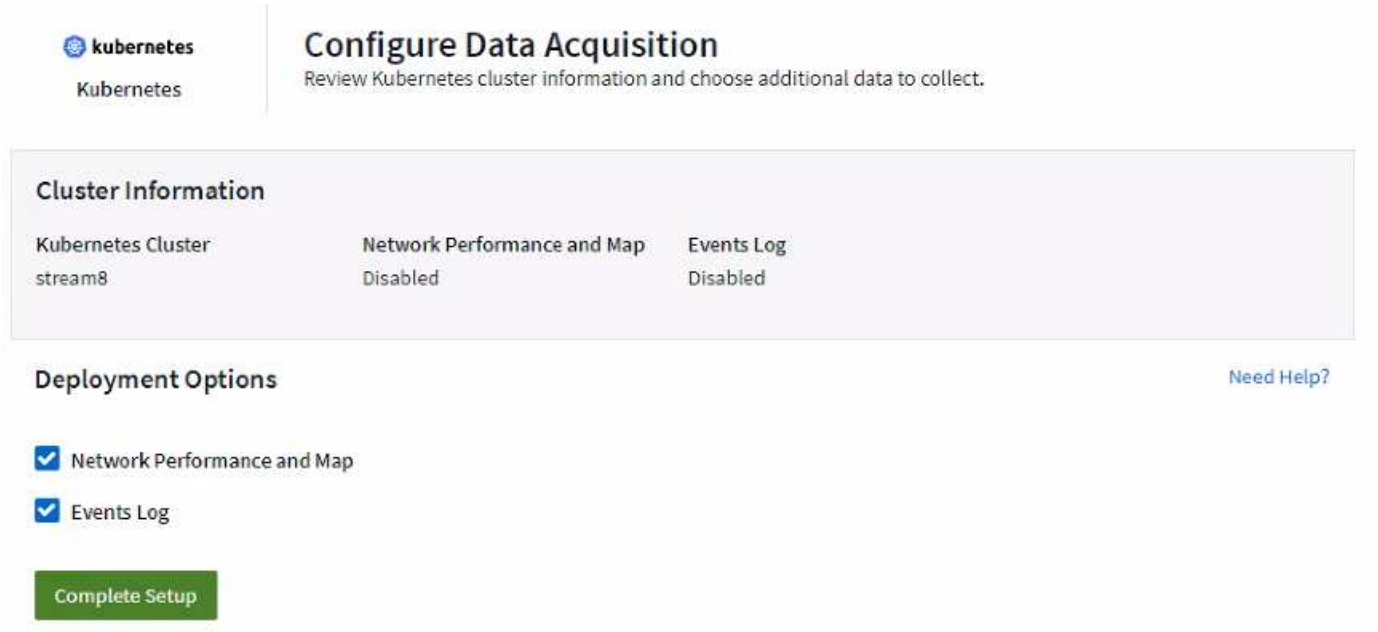
Surveillance et mappage des performances du réseau Kubernetes

La fonctionnalité de surveillance et de mappage des performances réseau Kubernetes simplifie la résolution de problèmes en mappant les dépendances entre les services (également appelés workloads). Elle fournit une visibilité en temps réel sur les latences des performances réseau et les anomalies pour identifier les problèmes de performance avant qu'ils n'affectent les utilisateurs. Cette fonctionnalité aide les entreprises à réduire les coûts globaux grâce à l'analyse et à l'audit des flux de trafic Kubernetes.

Principales fonctionnalités : • la carte des workloads présente les dépendances et les flux des workloads Kubernetes, et souligne les problèmes de réseau et de performance. • Surveiller le trafic réseau entre les pods Kubernetes, les workloads et les nœuds ; identifier la source des problèmes de trafic et de latence. • Réduire les coûts globaux en analysant les entrées, les sorties, le trafic interrégional et le trafic de réseau interzone.

Conditions préalables

Avant de pouvoir utiliser le contrôle et le mappage des performances du réseau Kubernetes, vous devez avoir configuré le "Opérateur de surveillance NetApp Kubernetes" pour activer cette option. Pendant le déploiement de l'opérateur, cochez la case « performances du réseau et carte » pour l'activer. Vous pouvez également activer cette option en accédant à une page d'accueil Kubernetes et en sélectionnant « Modifier le déploiement ».



Moniteurs

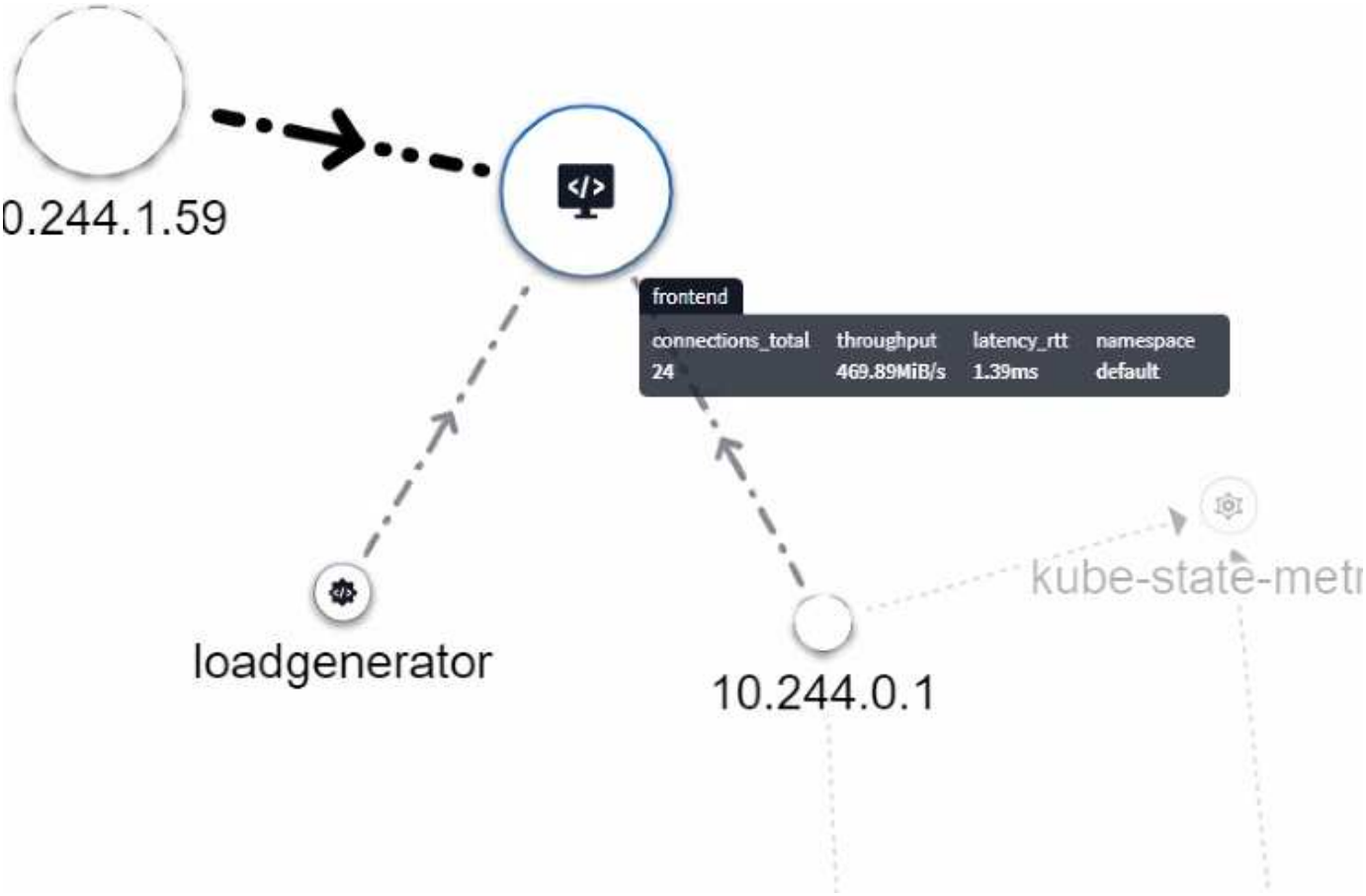
La carte des charges de travail utilise "moniteurs" pour dériver des informations. Data Infrastructure Insights fournit plusieurs moniteurs Kubernetes par défaut (ceux-ci peuvent être *utilisés* par défaut). Vous pouvez *reprendre* (c'est-à-dire activer) les moniteurs de votre choix), ou vous pouvez créer des moniteurs personnalisés pour les objets kubernetes que Workload Map utilisera également.

Vous pouvez créer des alertes de metric Data Infrastructure Insights sur l'un des types d'objet ci-dessous. Assurez-vous que les données sont regroupées par type d'objet par défaut.

- workloads kubernetes
- kubernetes.demonset
- kubernetes.deployment
- kubernetes.cronjob
- kubernetes.travail
- kubernetes.repliaset
- kubernetes.statefulset
- pod kubernetes
- kubernetes.network_traffic_l4

La carte

La carte montre les services/charges de travail et leurs relations les uns avec les autres. Les flèches indiquent l'itinéraire de la circulation. Le survol d'une charge de travail affiche un récapitulatif des informations relatives à cette charge, comme vous pouvez le voir dans cet exemple :



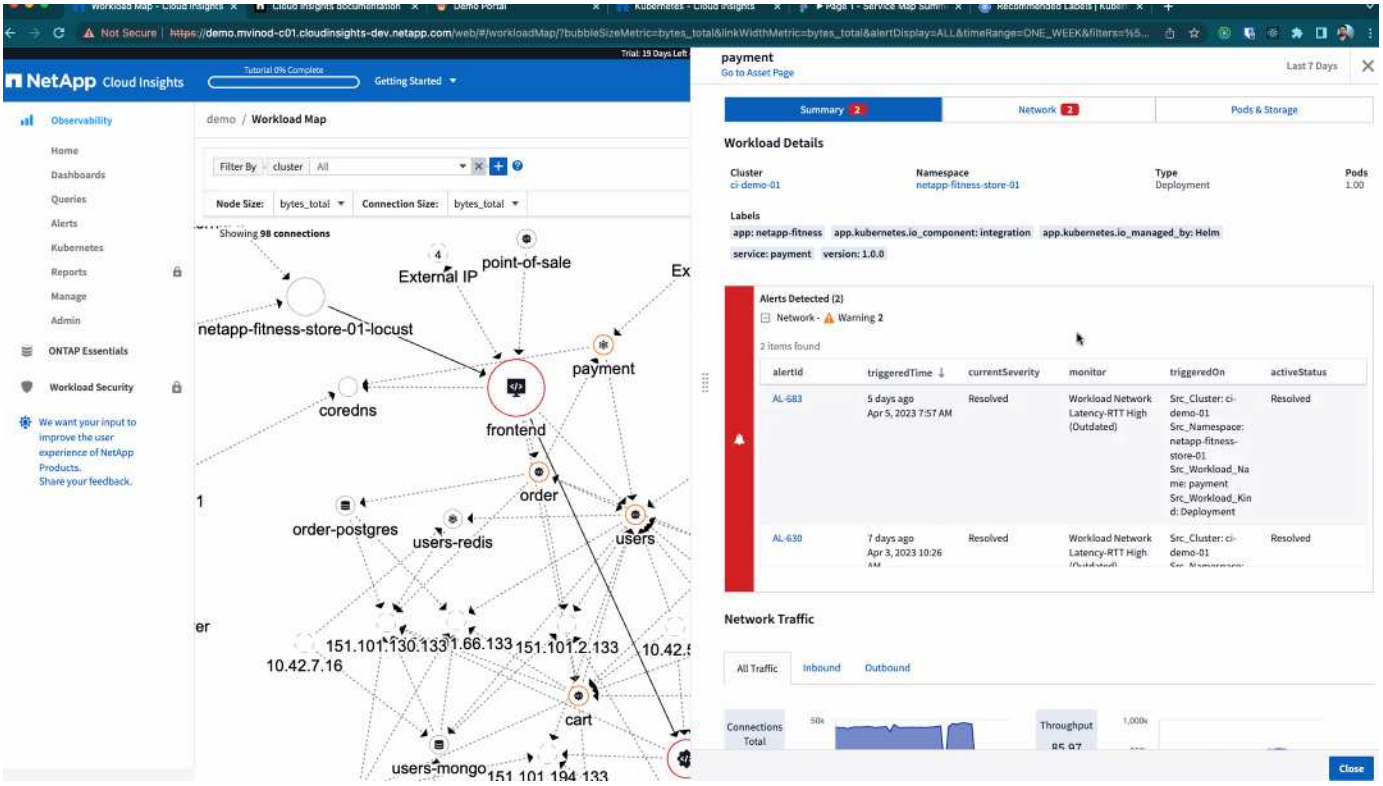
Les icônes situées dans les cercles représentent différents types de services. Notez que les icônes ne sont visibles que si les objets sous-jacents ont étiquettes.

La taille de chaque cercle indique la taille du nœud. Notez que ces tailles sont relatives, le niveau de zoom ou la taille de l'écran de votre navigateur peut affecter la taille réelle des cercles. De la même manière, le style de ligne de trafic vous donne une vue d'un coup d'œil de la taille de la connexion ; les lignes en gras sont à fort trafic, tandis que les lignes en pointillés légers sont à faible trafic.

Les nombres à l'intérieur des cercles correspondent au nombre de connexions externes en cours de traitement par le service.

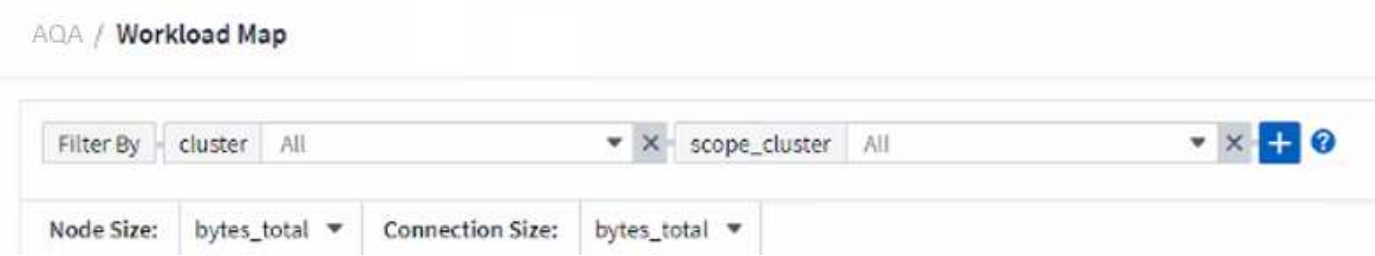
Détails de la charge de travail et alertes

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

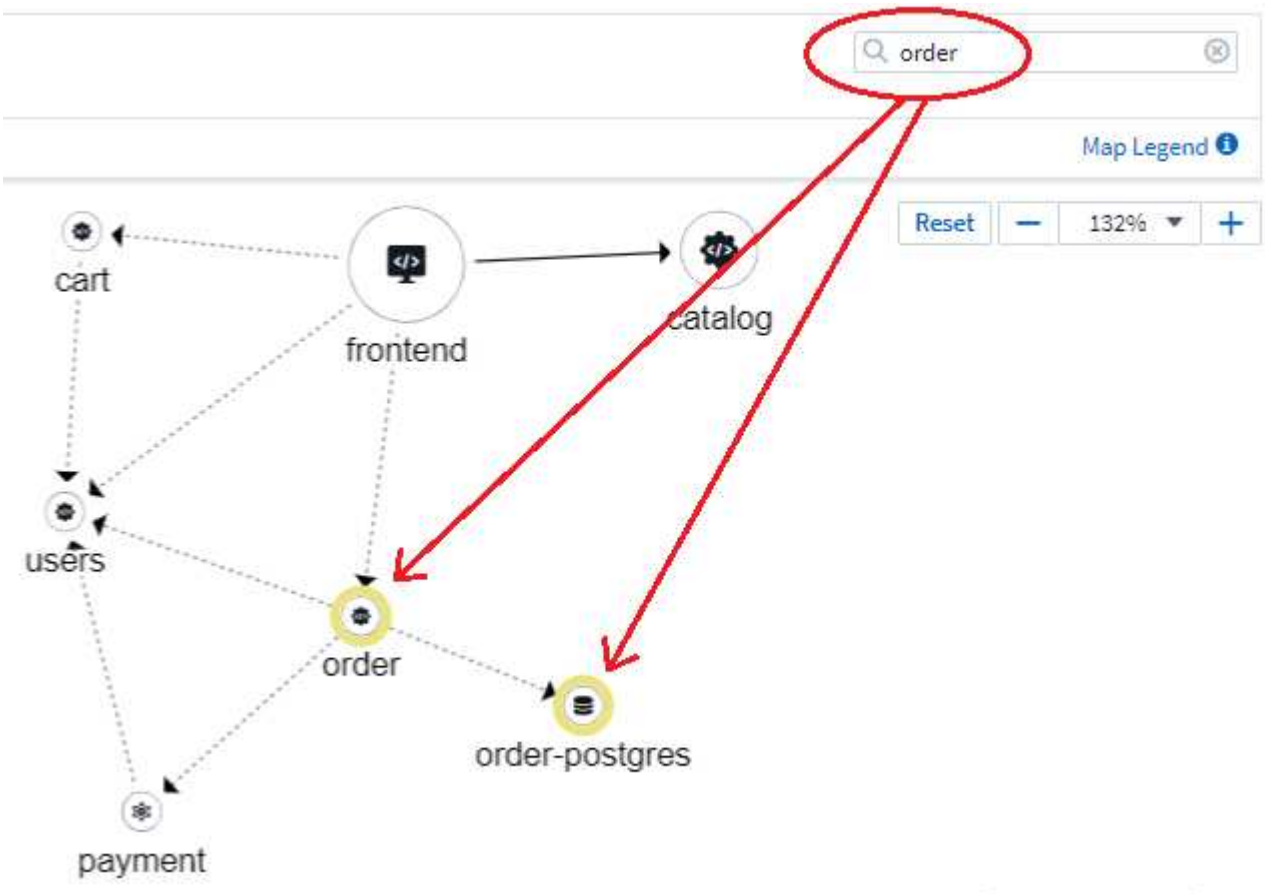


Recherche et filtrage

À l'instar des autres fonctionnalités Data Infrastructure Insights, vous pouvez facilement définir des filtres pour vous concentrer sur les objets ou les attributs de workload spécifiques que vous souhaitez.



De même, la saisie d'une chaîne dans le champ *find* met en surbrillance 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 cercle). Les étiquettes sont dérivées comme suit :

- Nom du service/de l'application s'exécutant en termes génériques
- Si la source est un pod :
  - L'étiquette est dérivée de l'étiquette de charge de travail du pod
  - Libellé attendu sur le workload : app.kubernetes.io/composant
  - Référence du nom de l'étiquette : https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels/
  - Étiquettes recommandées :
    - front-end

- back-end
- base de données
- cache
- file d'attente
- kafka

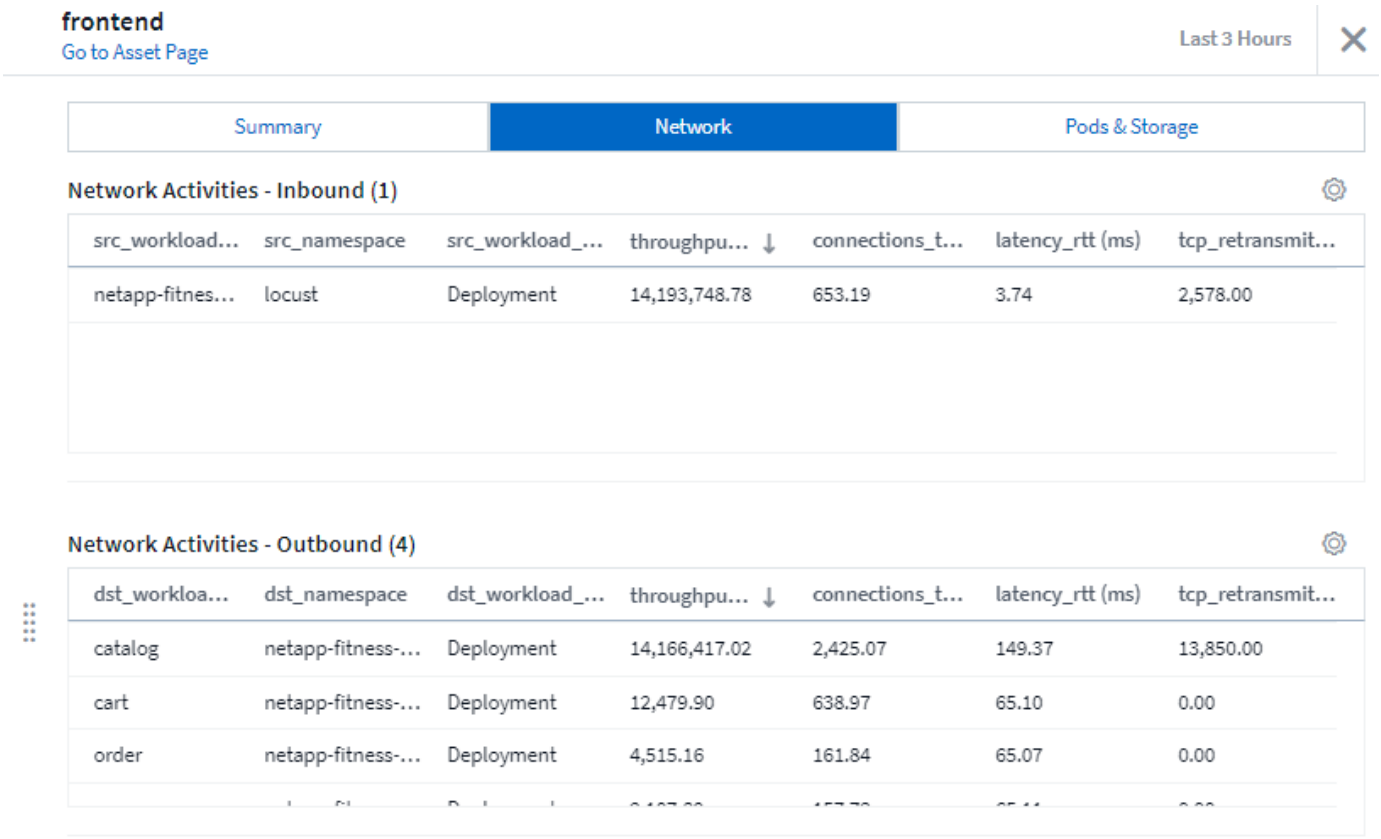
- Si la source est externe au cluster kubernetes :
  - Data Infrastructure Insights tente d'analyser le nom DNS résolu pour extraire le type de service.

    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.

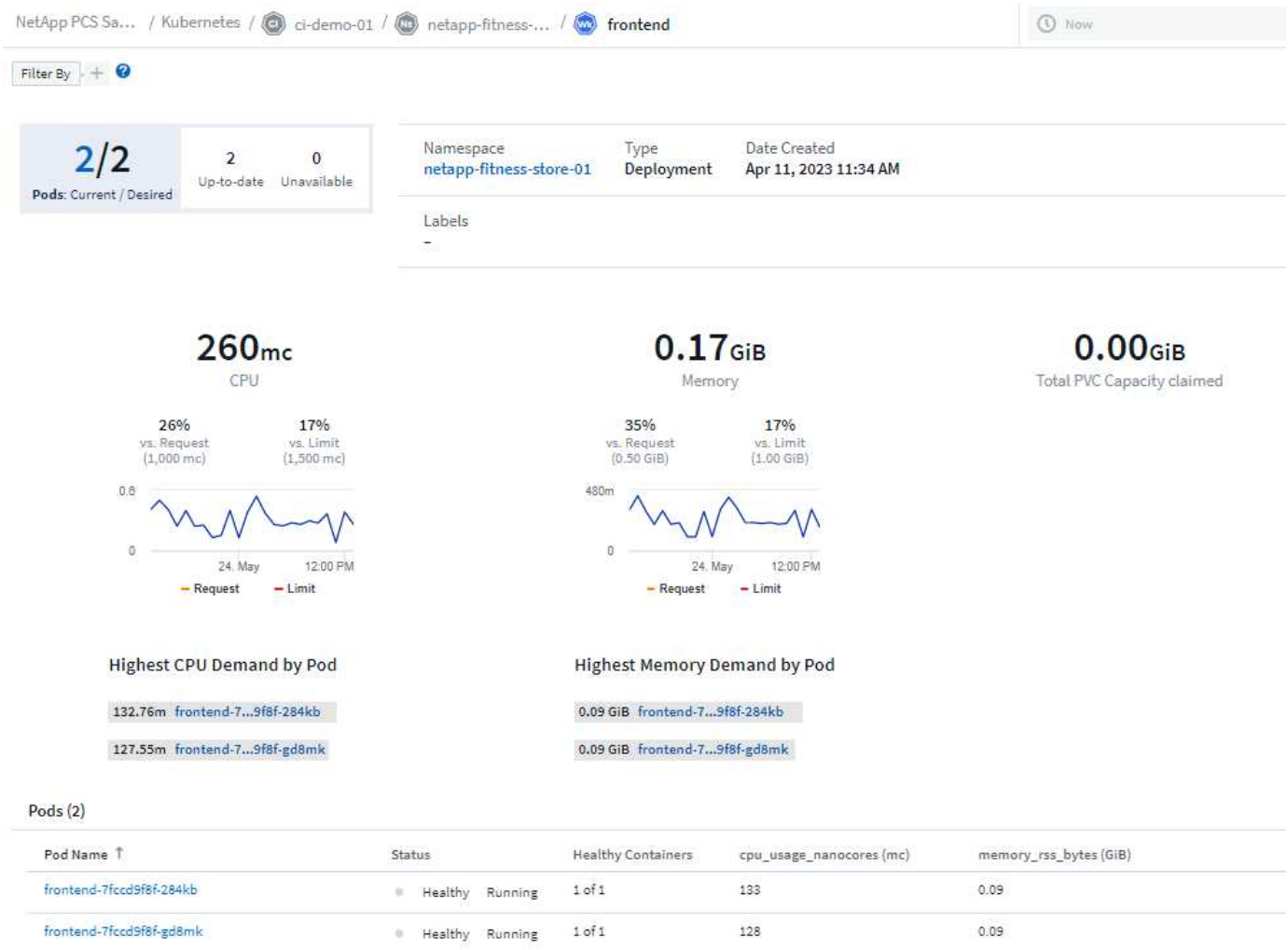
Plongez au cœur de l'aventure

Cliquez avec le bouton droit de la souris sur une charge de travail pour afficher des options supplémentaires afin d'en savoir plus. Par exemple, vous pouvez effectuer un zoom avant pour afficher les connexions de cette charge de travail.

Vous pouvez également ouvrir le panneau détaillé pour afficher directement l'onglet *Summary*, *Network* ou *Pod & Storage*.



Enfin, en sélectionnant *aller à la page de ressources*, vous ouvrez la page d'accueil détaillée de la ressource pour la charge de travail.



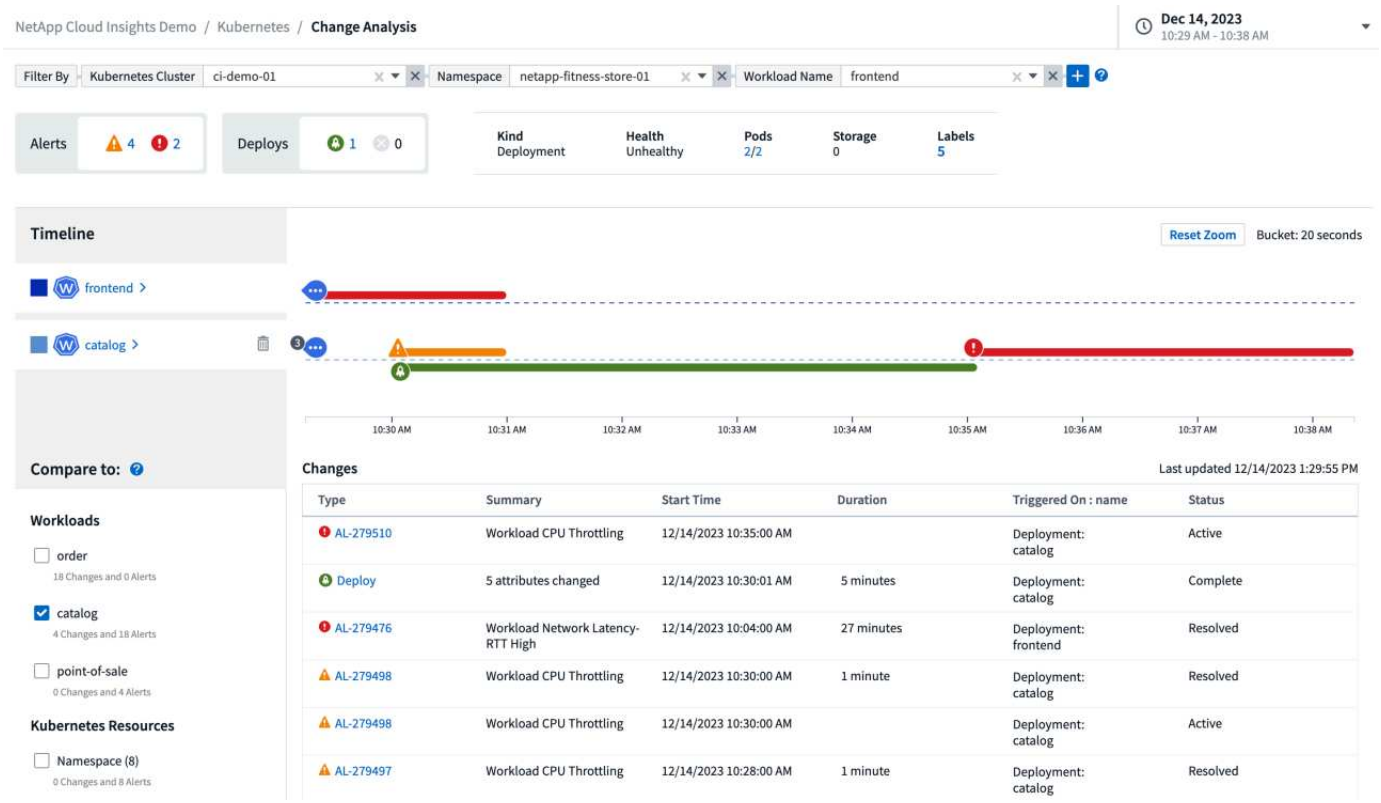
Analyse des changements Kubernetes

L'analytique des changements Kubernetes vous offre une vue complète des modifications récentes de 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 mettre en corrélation avec l'état et les performances des services, de l'infrastructure et des clusters K8s.

Comment l'analyse des changements peut-elle aider ?

- Dans les environnements Kubernetes mutualisés, les pannes peuvent se produire en raison de modifications mal configurées. L'analyse des changements permet d'y apporter une solution en fournissant une seule interface pour afficher et mettre en corrélation l'état des workloads et les modifications de configuration. Cela peut vous aider dans le dépannage des environnements Kubernetes dynamiques.

Pour afficher l'analyse des changements Kubernetes, accédez à **Kubernetes > analyse des changements**.



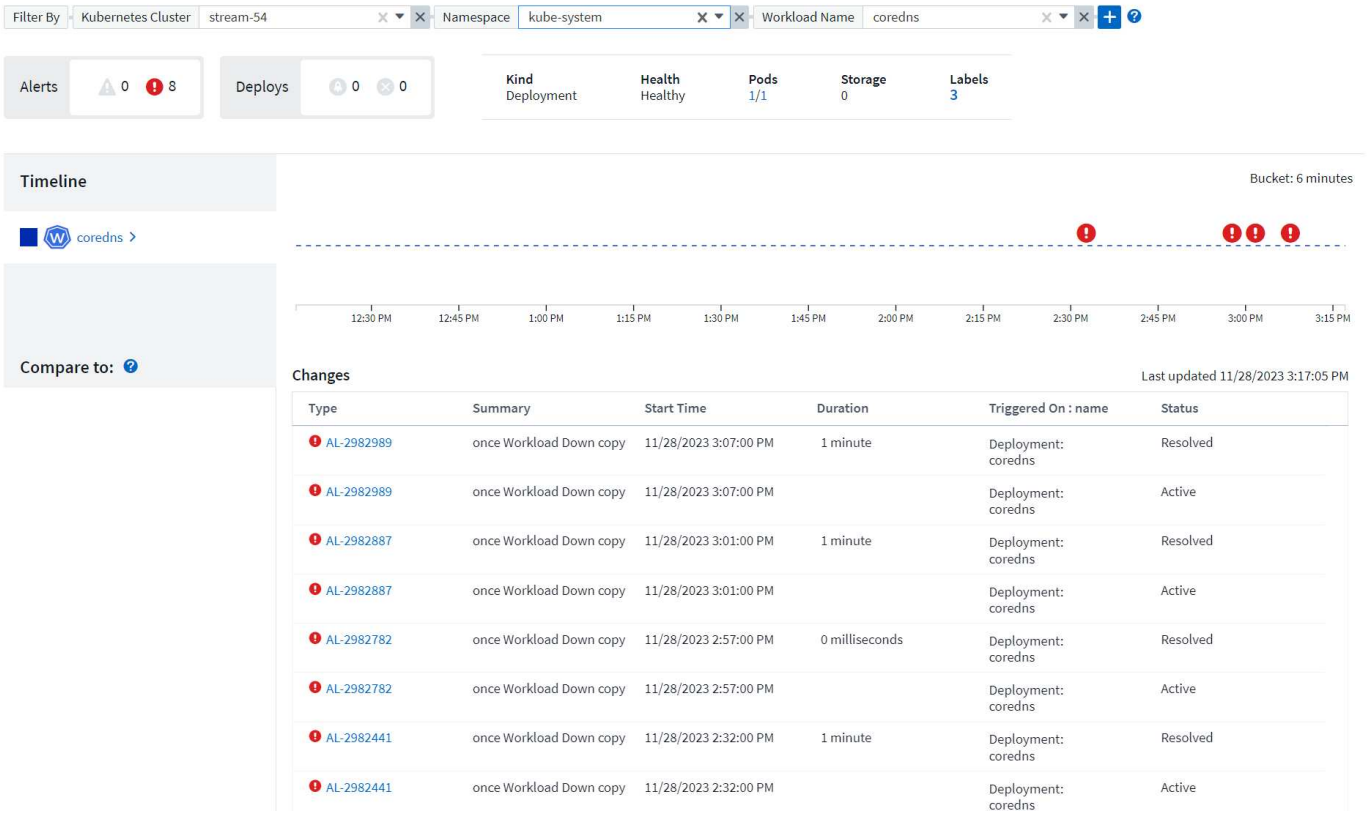
.

La page est automatiquement actualisée en fonction de la plage horaire Data Infrastructure Insights actuellement sélectionnée. Des plages de temps plus petites permettent un rafraîchissement plus fréquent de l'écran.

Filtrage

Comme toutes les fonctionnalités de Data Infrastructure Insights, il est intuitif de filtrer la liste de modifications : en haut de la page, entrez ou sélectionnez des valeurs pour votre cluster, espace de noms ou workload Kubernetes, ou ajoutez vos propres filtres en cliquant sur le bouton {+].

Lorsque vous filtrez vers un cluster, un espace de noms et une charge de travail spécifique (ainsi que tous les autres filtres que vous avez définis), la chronologie des déploiements et des alertes s'affiche pour cette charge de travail dans cet espace de noms sur ce cluster. Pour effectuer un zoom avant, cliquez sur le graphique et faites-le glisser pour vous concentrer sur une plage horaire plus spécifique.

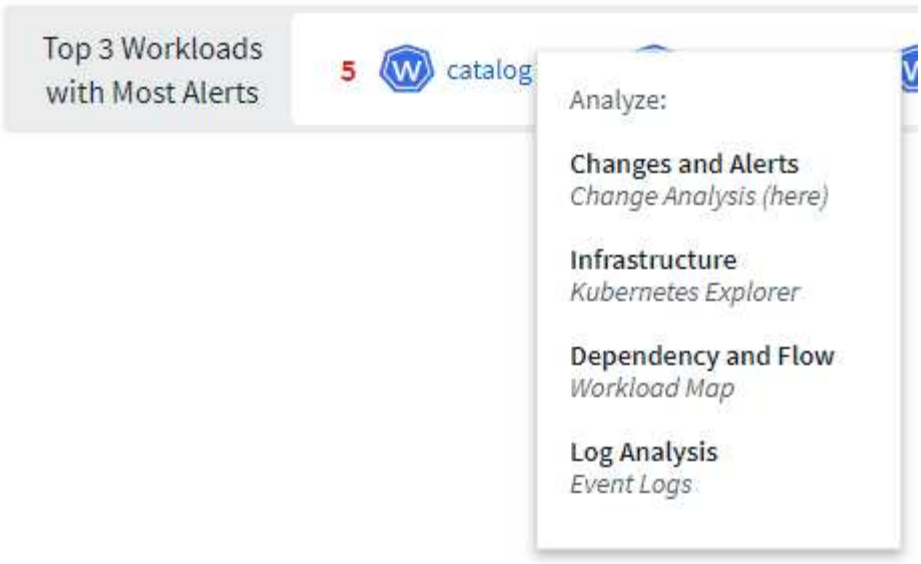


Statut rapide

Sous la zone de filtrage se trouvent un certain nombre d'indicateurs de haut niveau. Le nombre d'alertes (Avertissement et critique) se trouve à gauche. Ce nombre inclut les alertes *Active* ainsi que les alertes *Resolved*. Pour afficher uniquement les alertes *Active*, définissez un filtre pour « Status » et choisissez « Active ».

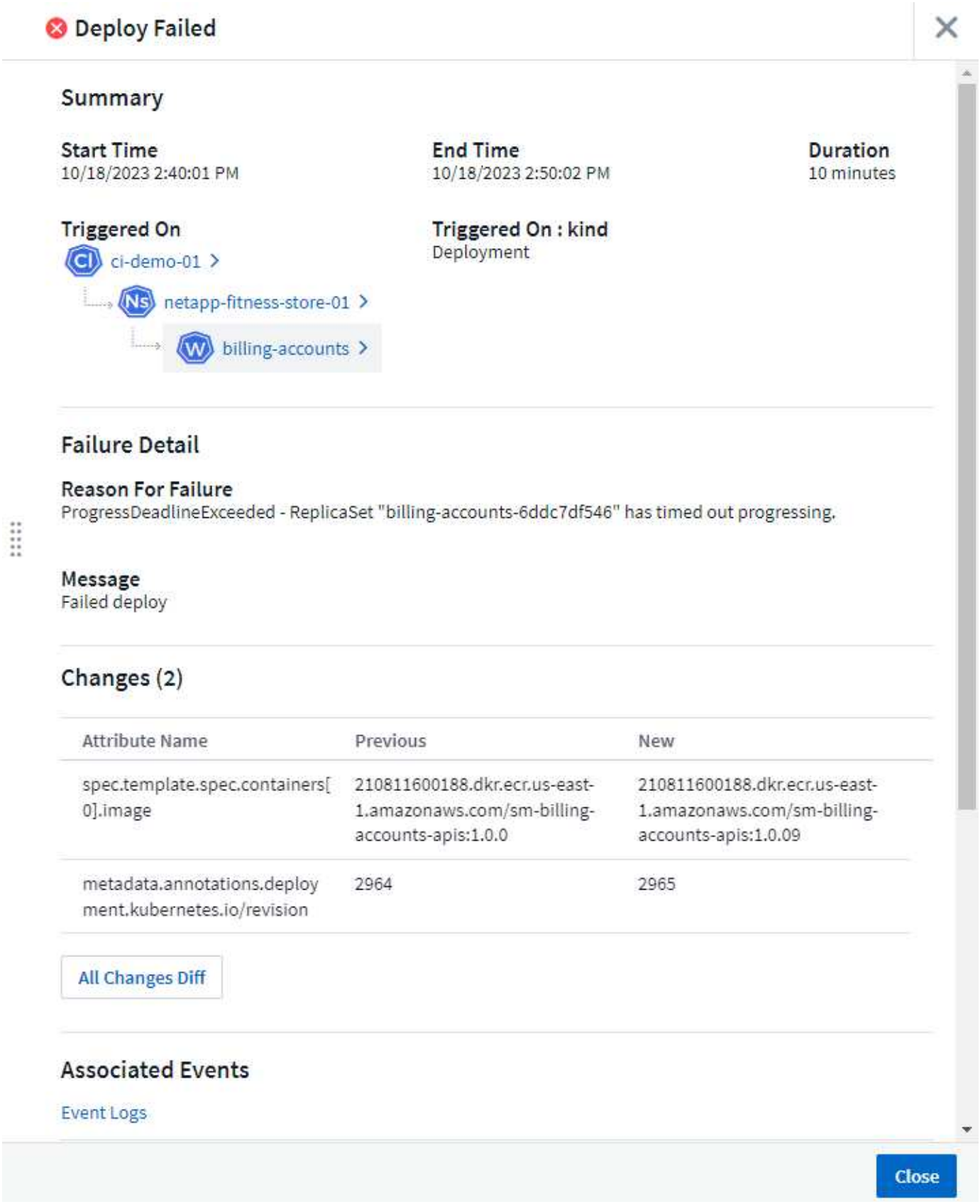
L'état du déploiement est également indiqué ici. Encore une fois, la valeur par défaut est d'afficher le nombre de déploiements *démarrés*, *complets* et *échoués*. Pour afficher uniquement les déploiements *FAILED*, définissez un filtre pour « Status » et sélectionnez « FAILED ».

Les 3 principaux workloads avec le plus grand nombre d'alertes viennent ensuite. Le nombre en rouge en regard de chaque charge de travail indique le nombre d'alertes associées à cette charge de travail. Cliquez sur le lien correspondant aux workloads pour explorer votre infrastructure (Explorateur Kubernetes), les dépendances (carte des workloads) ou l'analyse des journaux (journaux d'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 en échec affiche un résumé du déploiement, avec les heures de début et de fin, la durée et l'endroit où le déploiement a été déclenché, avec des liens permettant d'explorer ces ressources. Il affiche également la raison de l'échec, les modifications associées et les événements associés.



La sélection d'une alerte de même type fournit des détails sur l'alerte, y compris le moniteur qui a déclenché l'alerte, ainsi qu'un graphique montrant un chronogramme visuel pour l'alerte.