Kubernetes

Data Infrastructure Insights

NetApp
February 03, 2026

Sommario

Kubernetes

Panoramica del cluster Kubernetes

Data Infrastructure Insights Kubernetes Explorer è un potente strumento per visualizzare lo stato generale e l'utilizzo dei cluster Kubernetes e consente di analizzare in modo più approfondito le aree di indagine.

Facendo clic su **Dashboard > Kubernetes Explorer** si apre la pagina con l'elenco dei cluster Kubernetes. Questa pagina di panoramica contiene la tabella dei cluster Kubernetes presenti nel tuo tenant.

[Pagina dell'elenco di Kubernetes]

Elenco dei cluster

L'elenco dei cluster visualizza le seguenti informazioni per ciascun cluster nel tenant:

- Cluster **Nome**. Facendo clic sul nome di un cluster si aprirà il**"pagina dei dettagli"** per quel cluster.
- Percentuali di **saturazione**. La saturazione complessiva è la più alta tra la saturazione della CPU, della memoria o dello storage.
- Numero di **Nodi** nel cluster. Facendo clic su questo numero si aprirà la pagina dell'elenco dei nodi.
- Numero di **Pod** nel cluster. Facendo clic su questo numero si aprirà la pagina dell'elenco dei Pod.
- Numero di **Namespace** nel cluster. Facendo clic su questo numero si aprirà la pagina dell'elenco degli spazi dei nomi.
- Numero di **carichi di lavoro** nel cluster. Facendo clic su questo numero si aprirà la pagina dell'elenco dei carichi di lavoro.

Perfezionamento del filtro

Quando si filtra, quando si inizia a digitare viene presentata l'opzione per creare un **filtro con caratteri jolly** in base al testo corrente. Selezionando questa opzione verranno restituiti tutti i risultati che corrispondono all'espressione jolly. È anche possibile creare **espressioni** utilizzando NOT o AND, oppure selezionare l'opzione "Nessuno" per filtrare i valori nulli nel campo.

[Filtraggio con caratteri jolly in K8S Explorer]

I filtri basati su caratteri jolly o espressioni (ad esempio NOT, AND, "Nessuno", ecc.) vengono visualizzati in blu scuro nel campo filtro. Gli elementi selezionati direttamente dall'elenco vengono visualizzati in azzurro.

[Filtro che mostra caratteri jolly ed elementi selezionati]

I filtri di Kubernetes sono contestuali, il che significa che se ci si trova, ad esempio, su una pagina di un nodo specifico, il filtro pod_name elenca solo i pod correlati a quel nodo. Inoltre, se si applica un filtro per uno specifico namespace, il filtro pod_name elencherà solo i pod su quel nodo *e* in quello namespace.

Si noti che il filtro con caratteri jolly ed espressioni funziona con testo o elenchi, ma non con valori numerici, date o valori booleani.

Prima di installare o aggiornare NetApp Kubernetes Monitoring Operator

Leggere queste informazioni prima di installare o aggiornare il"Operatore di monitoraggio Kubernetes" .

Componente	Requisito
Versione di Kubernetes	Kubernetes v1.20 e versioni successive.
Distribuzioni Kubernetes	AWS Elastic Kubernetes Service (EKS) Azure Kubernetes Service (AKS) Google Kubernetes Engine (GKE) Red Hat OpenShift Rancher Kubernetes Engine (RKE) VMware Tanzu
Sistema operativo Linux	Data Infrastructure Insights non supporta i nodi in esecuzione con architettura Arm64. Monitoraggio di rete: è necessario eseguire il kernel Linux versione 4.18.0 o successiva. Photon OS non è supportato.
Etichette	Data Infrastructure Insights supporta il monitoraggio dei nodi Kubernetes che eseguono Linux, specificando un selettore di nodi Kubernetes che cerca le seguenti etichette Kubernetes su queste piattaforme: Kubernetes v1.20 e versioni successive: Kubernetes.io/os = linux Rancher + cattle.io come piattaforma di orchestrazione/Kubernetes: cattle.io/os = linux
Comandi	I comandi curl e kubectl devono essere disponibili; per risultati ottimali, aggiungili al PATH.
Connettività	kubectl cli è configurato per comunicare con il cluster K8s di destinazione e per avere connettività Internet con l'ambiente Data Infrastructure Insights . Se durante l'installazione ci si trova dietro un proxy, seguire le istruzioni riportate nel"Configurazione del supporto proxy" sezione dell'installazione dell'operatore. Per un audit e una segnalazione dei dati accurati, sincronizzare l'ora sulla macchina dell'agente utilizzando il protocollo NTP (Network Time Protocol) o il protocollo SNTP (Simple Network Time Protocol).
Altro	Se utilizzi OpenShift 4.6 o versione successiva, devi seguire le istruzioni"Istruzioni OpenShift" oltre a garantire che questi prerequisiti siano soddisfatti.
Token API	Se si sta ridistribuendo l'operatore (ovvero lo si sta aggiornando o sostituendo), non è necessario creare un nuovo token API; è possibile riutilizzare il token precedente.

Cose importanti da notare prima di iniziare

Se stai correndo con unprocuratore , avere unrepository personalizzato , o stanno usandoOpenShift , leggere attentamente le seguenti sezioni.

Leggi anche suPermessi .

Configurazione del supporto proxy

Esistono due posizioni in cui è possibile utilizzare un proxy sul tenant per installare NetApp Kubernetes Monitoring Operator. Possono essere gli stessi sistemi proxy o sistemi proxy separati:

- Proxy necessario durante l'esecuzione dello snippet di codice di installazione (utilizzando "curl") per connettere il sistema in cui viene eseguito lo snippet al tuo ambiente Data Infrastructure Insights
- Proxy necessario al cluster Kubernetes di destinazione per comunicare con l'ambiente Data Infrastructure Insights

Se si utilizza un proxy per uno o entrambi questi elementi, per installare NetApp Kubernetes Operating Monitor è necessario innanzitutto assicurarsi che il proxy sia configurato per consentire una buona comunicazione con l'ambiente Data Infrastructure Insights . Ad esempio, dai server/VM da cui si desidera installare l'operatore, è necessario poter accedere a Data Infrastructure Insights e poter scaricare i file binari da Data Infrastructure Insights.

Per il proxy utilizzato per installare NetApp Kubernetes Operating Monitor, prima di installare l'operatore, impostare le variabili di ambiente *http_proxy/https_proxy*. Per alcuni ambienti proxy, potrebbe essere necessario impostare anche la variabile di ambiente *no_proxy*.

Per impostare le variabili, eseguire i seguenti passaggi sul sistema **prima** di installare NetApp Kubernetes Monitoring Operator:

1. Imposta le variabili di ambiente *https_proxy* e/o *http_proxy* per l'utente corrente:
   a. Se il proxy da configurare non dispone di autenticazione (nome utente/password), eseguire il seguente comando:

```
 export https_proxy=<proxy_server>:<proxy_port>
.. Se il proxy da configurare dispone di autenticazione (nome
utente/password), eseguire questo comando:
```

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

Per far sì che il proxy utilizzato per il cluster Kubernetes comunichi con l'ambiente Data Infrastructure Insights , installare NetApp Kubernetes Monitoring Operator dopo aver letto tutte queste istruzioni.

Configurare la sezione proxy di AgentConfiguration in operator-config.yaml prima di distribuire NetApp 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
  ...
...
```

Utilizzo di un repository Docker personalizzato o privato

Per impostazione predefinita, NetApp Kubernetes Monitoring Operator estrarrà le immagini dei container dal repository Data Infrastructure Insights . Se si utilizza un cluster Kubernetes come destinazione per il monitoraggio e tale cluster è configurato per estrarre immagini di container solo da un repository Docker personalizzato o privato o da un registro di container, è necessario configurare l'accesso ai container necessari all'operatore di monitoraggio NetApp Kubernetes.

Eseguire "Image Pull Snippet" dal riquadro di installazione di NetApp Monitoring Operator. Questo comando effettuerà l'accesso al repository Data Infrastructure Insights , estrarrà tutte le dipendenze delle immagini per l'operatore e uscirà dal repository Data Infrastructure Insights . Quando richiesto, immettere la password temporanea del repository fornita. Questo comando scarica tutte le immagini utilizzate dall'operatore, comprese quelle per le funzionalità opzionali. Di seguito sono riportate le funzioni per cui vengono utilizzate queste immagini.

Funzionalità dell'operatore principale e monitoraggio di Kubernetes

- monitoraggio netapp
- kube-rbac-proxy
- metriche dello stato di Kube
- telegrafo
- utente root senza distribuzione

Registro eventi

- fluente-bit

- esportatore di eventi kubernetes

Prestazioni e mappa della rete

- ci-net-observer

Invia l'immagine Docker dell'operatore al tuo repository Docker privato/locale/aziendale in base alle policy aziendali. Assicurati che i tag delle immagini e i percorsi delle directory di queste immagini nel tuo repository siano coerenti con quelli nel repository Data Infrastructure Insights .

Modifica la distribuzione monitoring-operator in operator-deployment.yaml e modifica tutti i riferimenti alle immagini per utilizzare il tuo repository Docker privato.

```
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>
```

Modifica AgentConfiguration in operator-config.yaml per riflettere la nuova posizione del repository Docker. Crea un nuovo imagePullSecret per il tuo repository privato, per maggiori dettagli consulta *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
```

Istruzioni OpenShift

Se utilizzi OpenShift 4.6 o versione successiva, devi modificare AgentConfiguration in *operator-config.yaml* per abilitare l'impostazione *runPrivileged*:

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

Openshift potrebbe implementare un livello di sicurezza aggiuntivo che potrebbe bloccare l'accesso ad alcuni componenti di Kubernetes.

Permessi

Se il cluster che stai monitorando contiene risorse personalizzate che non hanno un ClusterRole che"aggregati da visualizzare" , sarà necessario concedere manualmente all'operatore l'accesso a queste risorse per monitorarle con i registri eventi.

1. Modificare *operator-additional-permissions.yaml* prima dell'installazione oppure, dopo l'installazione, modificare la risorsa *ClusterRole/<namespace>-additional-permissions*
2. Crea una nuova regola per gli apiGroup e le risorse desiderati con i verbi ["get", "watch", "list"]. Vedi \ https://kubernetes.io/docs/reference/access-authn-authz/rbac/
3. Applica le modifiche al cluster

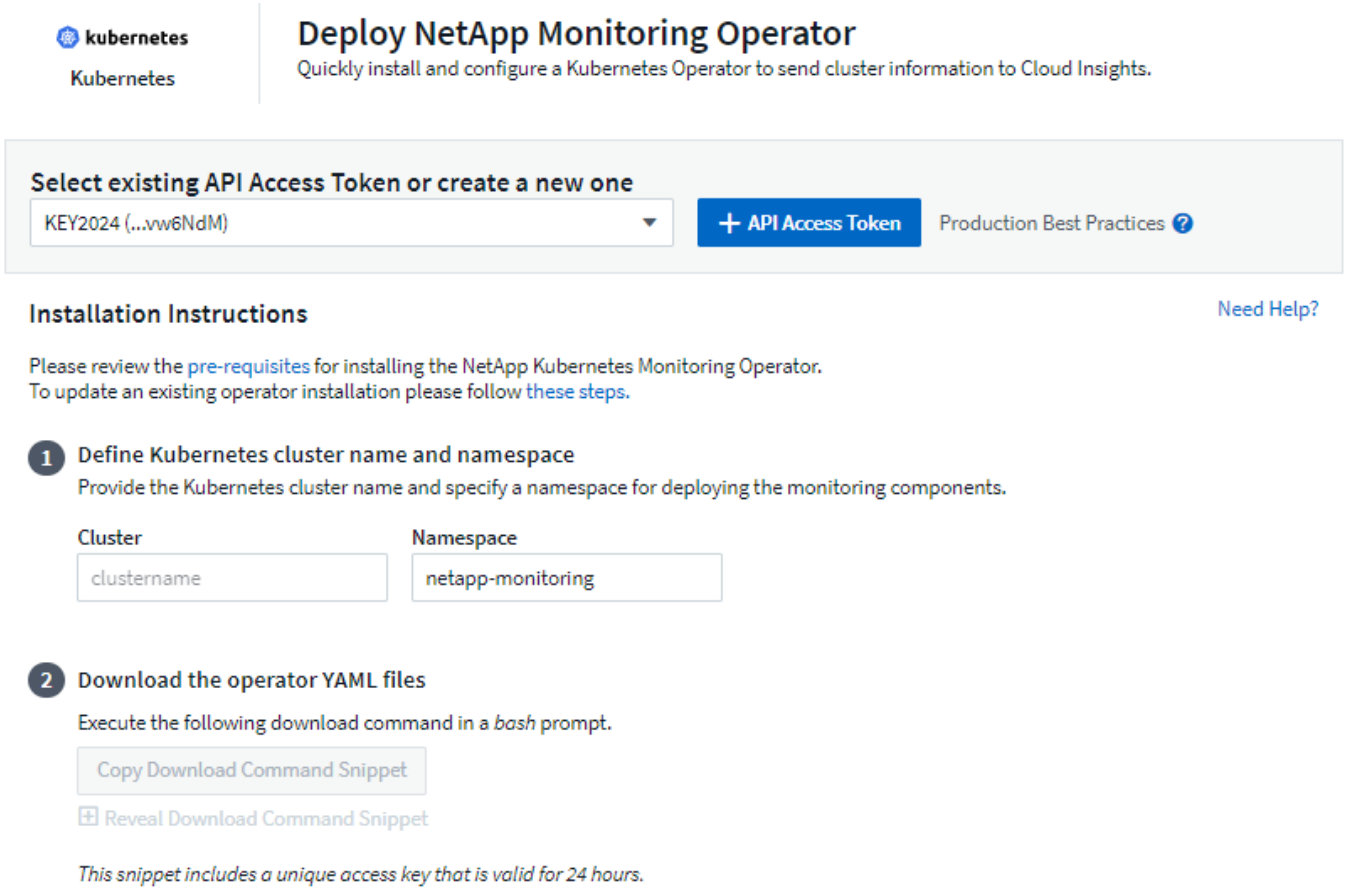
Installazione e configurazione dell'operatore di monitoraggio Kubernetes

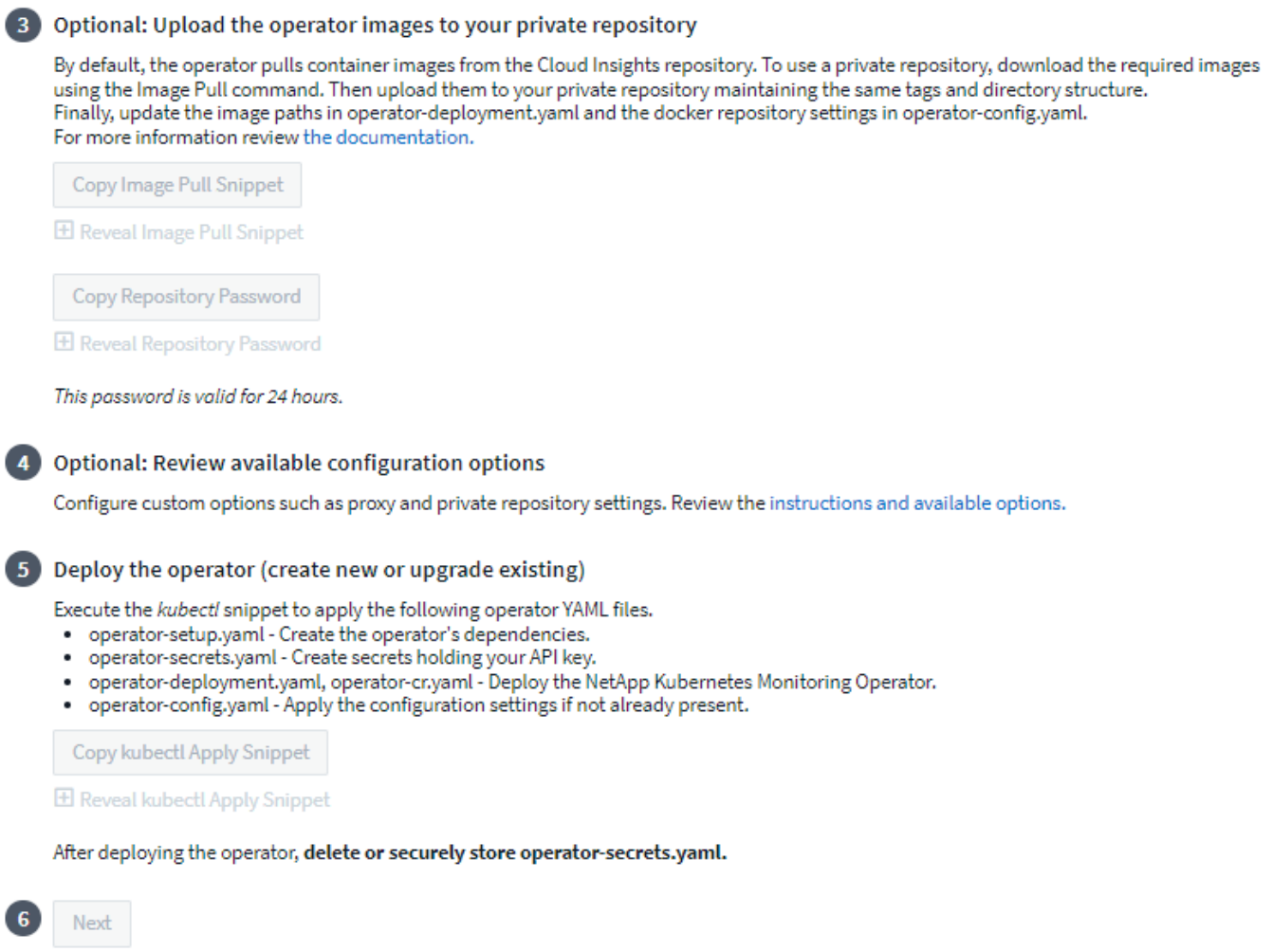
Data Infrastructure Insights offre l'**operatore di monitoraggio Kubernetes** per la raccolta Kubernetes. Passare a **Kubernetes > Collectors > +Kubernetes Collector** per distribuire un nuovo operatore.

Prima di installare Kubernetes Monitoring Operator

Vedi il"Prerequisiti" documentazione prima di installare o aggiornare Kubernetes Monitoring Operator.

Installazione dell'operatore di monitoraggio Kubernetes





Passaggi per installare l'agente Kubernetes Monitoring Operator su Kubernetes:

1. Immettere un nome cluster e uno spazio dei nomi univoci. Se seiaggiornamento da un precedente operatore Kubernetes, utilizzare lo stesso nome del cluster e lo stesso spazio dei nomi.
2. Una volta inseriti questi dati, è possibile copiare il frammento del comando Download negli appunti.
3. Incolla lo snippet in una finestra *bash* ed eseguilo. Verranno scaricati i file di installazione dell'operatore. Si noti che lo snippet ha una chiave univoca ed è valido per 24 ore.
4. Se hai un repository personalizzato o privato, copia il frammento di codice Image Pull facoltativo, incollalo in una shell *bash* ed eseguilo. Una volta estratte le immagini, copiale nel tuo repository privato. Assicuratevi di mantenere gli stessi tag e la stessa struttura delle cartelle. Aggiornare i percorsi in *operator-deployment.yaml* e le impostazioni del repository Docker in *operator-config.yaml*.
5. Se lo si desidera, rivedere le opzioni di configurazione disponibili, come le impostazioni del proxy o del repository privato. Puoi leggere di più su"opzioni di configurazione" .
6. Quando sei pronto, distribuisci l'operatore copiando lo snippet Apply di kubectl, scaricandolo ed eseguendolo.
7. L'installazione procede automaticamente. Una volta completato, fare clic sul pulsante *Avanti*.
8. Al termine dell'installazione, fare clic sul pulsante *Avanti*. Assicurati di eliminare o archiviare in modo sicuro anche il file *operator-secrets.yaml*.

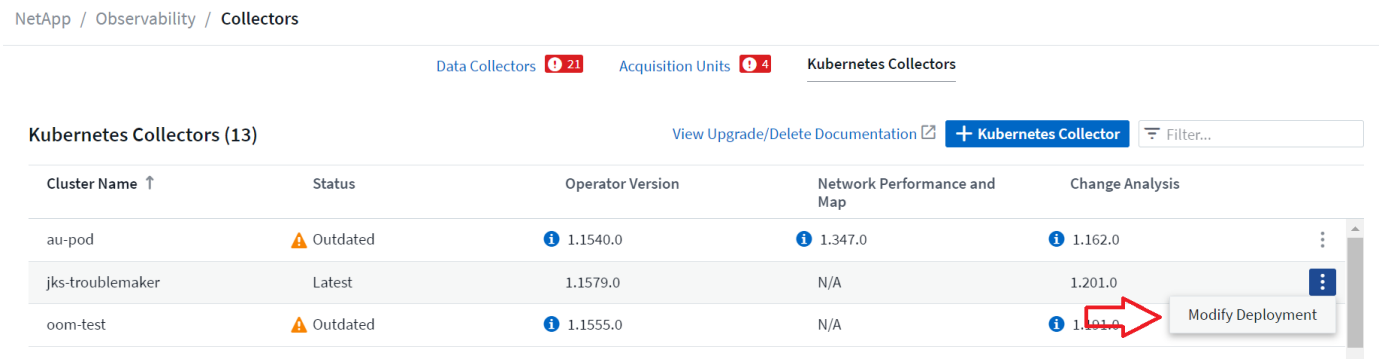
Se hai un repository personalizzato, leggi a riguardoutilizzando un repository Docker personalizzato/privato .

Componenti di monitoraggio di Kubernetes

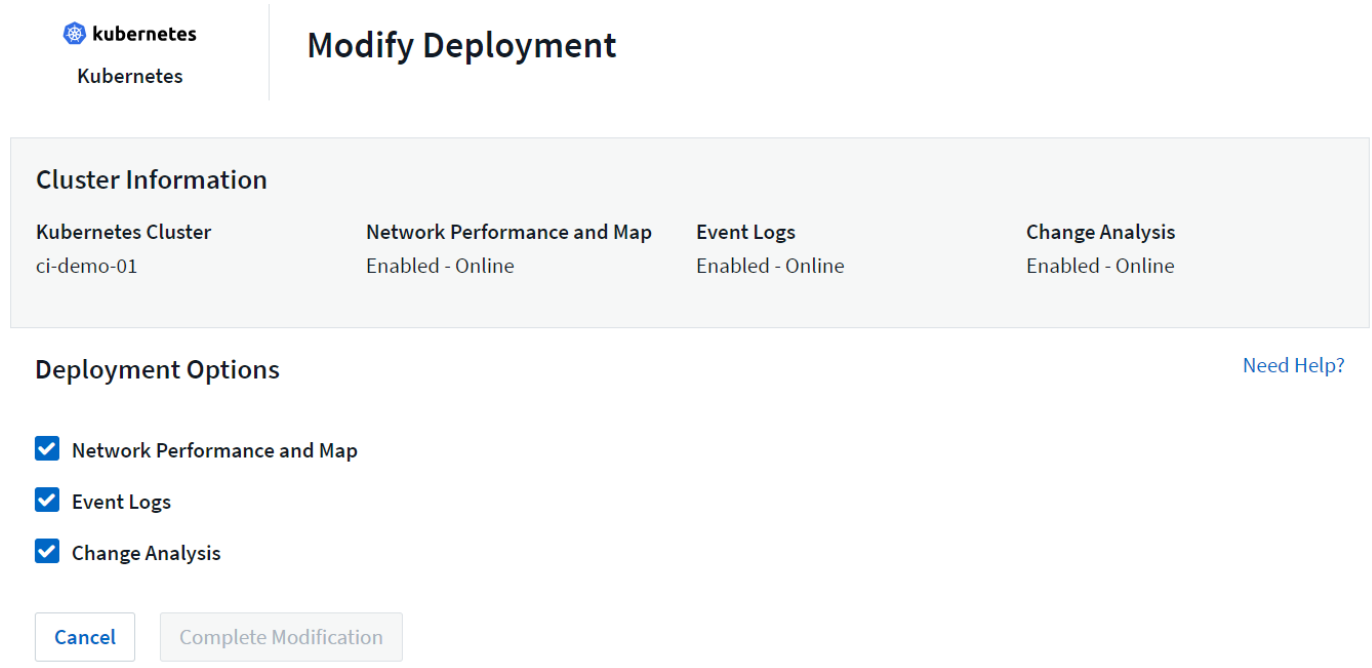
Data Infrastructure Insights Kubernetes Monitoring è composto da quattro componenti di monitoraggio:

- Metriche del cluster
- Prestazioni di rete e mappa (facoltativo)
- Registri eventi (facoltativo)
- Analisi del cambiamento (facoltativo)

I componenti facoltativi sopra indicati sono abilitati per impostazione predefinita per ciascun collector Kubernetes; se decidi che non hai bisogno di un componente per un collector specifico, puoi disabilitarlo andando su **Kubernetes > Collectors** e selezionando *Modifica distribuzione* dal menu "tre punti" del collector sulla destra dello schermo.



La schermata mostra lo stato attuale di ciascun componente e consente di disabilitare o abilitare i componenti per quel raccoglitore, a seconda delle necessità.



Aggiornamento all'ultima versione di Kubernetes Monitoring Operator

Aggiornamenti dei pulsanti DII

È possibile aggiornare Kubernetes Monitoring Operator tramite la pagina DII Kubernetes Collectors. Fare clic sul menu accanto al cluster che si desidera aggiornare e selezionare *Aggiorna*. L'operatore verificherà le firme delle immagini, eseguirà uno snapshot dell'installazione corrente ed eseguirà l'aggiornamento. Entro pochi minuti dovresti vedere l'avanzamento dello Stato dell'operatore da Aggiornamento in corso a Ultimo. Se si verifica un errore, è possibile selezionare lo stato Errore per maggiori dettagli e fare riferimento alla tabella di risoluzione dei problemi degli aggiornamenti tramite pulsante riportata di seguito.

Aggiornamenti rapidi con repository privati

Se il tuo operatore è configurato per utilizzare un repository privato, assicurati che tutte le immagini necessarie per eseguire l'operatore e le relative firme siano disponibili nel tuo repository. Se durante il processo di aggiornamento si verifica un errore per immagini mancanti, è sufficiente aggiungerle al repository e riprovare l'aggiornamento. Per caricare le firme delle immagini nel tuo repository, utilizza lo strumento di co-firma come segue, assicurandoti di caricare le firme per tutte le immagini specificate in 3 Facoltativo: carica le immagini dell'operatore nel tuo repository privato > Frammento di estrazione dell'immagine

```
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>
```

Ripristino di una versione precedentemente in esecuzione

Se hai effettuato l'aggiornamento utilizzando la funzionalità di aggiornamento tramite pulsante e riscontri difficoltà con la versione corrente dell'operatore entro sette giorni dall'aggiornamento, puoi effettuare il downgrade alla versione in esecuzione in precedenza utilizzando lo snapshot creato durante il processo di aggiornamento. Fare clic sul menu accanto al cluster di cui si desidera eseguire il rollback e selezionare *Roll back*.

Aggiornamenti manuali

Determinare se esiste un AgentConfiguration con l'operatore esistente (se lo spazio dei nomi non è quello predefinito *netapp-monitoring*, sostituirlo con lo spazio dei nomi appropriato):

```
 kubectl -n netapp-monitoring get agentconfiguration netapp-ci-monitoring-
configuration
Se esiste un AgentConfiguration:
```

- Installarel'ultimo Operatore rispetto all'Operatore esistente.
  - Assicurati di essereestrazione delle ultime immagini del contenitore se si utilizza un repository personalizzato.

Se AgentConfiguration non esiste:

- Prendi nota del nome del tuo cluster riconosciuto da Data Infrastructure Insights (se il tuo namespace non è quello predefinito netapp-monitoring, sostituiscilo con il namespace appropriato):

```
 kubectl -n netapp-monitoring get agent -o
jsonpath='{.items[0].spec.cluster-name}'
* Crea un backup dell'operatore esistente (se il tuo namespace non è il
netapp-monitoring predefinito, sostituiscilo con il namespace
appropriato):
```

```
 kubectl -n netapp-monitoring get agent -o yaml > agent_backup.yaml
* <<to-remove-the-kubernetes-monitoring-
operator,Disinstallare>>l'operatore esistente.
* <<installing-the-kubernetes-monitoring-operator,Installare>>l'ultimo
Operatore.
```

- Utilizzare lo stesso nome del cluster.
- Dopo aver scaricato gli ultimi file YAML dell'operatore, trasferire tutte le personalizzazioni presenti in agent_backup.yaml al file operator-config.yaml scaricato prima della distribuzione.
- Assicurati di essereestrazione delle ultime immagini del contenitore se si utilizza un repository personalizzato.

Arresto e avvio dell'operatore di monitoraggio Kubernetes

Per arrestare Kubernetes Monitoring Operator:

```
 kubectl -n netapp-monitoring scale deploy monitoring-operator
--replicas=0
Per avviare Kubernetes Monitoring Operator:
```

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

Disinstallazione

Per rimuovere l'operatore di monitoraggio Kubernetes

Si noti che lo spazio dei nomi predefinito per l'operatore di monitoraggio Kubernetes è "netapp-monitoring". Se hai impostato un tuo namespace, sostituiscilo in questi e in tutti i comandi e file successivi.

Le versioni più recenti dell'operatore di monitoraggio possono essere disinstallate con i seguenti comandi:

```
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>
```

Se l'operatore di monitoraggio è stato distribuito nel proprio namespace dedicato, eliminare il namespace:

```
 kubectl delete ns <NAMESPACE>
Nota: se il primo comando restituisce "Nessuna risorsa trovata",
utilizzare le seguenti istruzioni per disinstallare le versioni precedenti
dell'operatore di monitoraggio.
```

Eseguire ciascuno dei seguenti comandi nell'ordine indicato. A seconda dell'installazione corrente, alcuni di questi comandi potrebbero restituire messaggi di tipo "oggetto non trovato". Questi messaggi possono essere tranquillamente ignorati.

```
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>
```

Se in precedenza è stato creato un vincolo di contesto di sicurezza:

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

Informazioni su Kube-state-metrics

NetApp Kubernetes Monitoring Operator installa le proprie metriche kube-state per evitare conflitti con altre istanze.

Per informazioni su Kube-State-Metrics, vedere"questa pagina" .

Configurazione/Personalizzazione dell'operatore

Queste sezioni contengono informazioni sulla personalizzazione della configurazione dell'operatore, sull'utilizzo del proxy, sull'utilizzo di un repository Docker personalizzato o privato o sull'utilizzo di OpenShift.

Opzioni di configurazione

Le impostazioni modificate più comunemente possono essere configurate nella risorsa personalizzata *AgentConfiguration*. È possibile modificare questa risorsa prima di distribuire l'operatore modificando il file *operator-config.yaml*. Questo file include esempi di impostazioni commentati. Vedi l'elenco di"impostazioni disponibili" per la versione più recente dell'operatore.

È anche possibile modificare questa risorsa dopo aver distribuito l'operatore utilizzando il seguente comando:

```
 kubectl -n netapp-monitoring edit AgentConfiguration
Per determinare se la versione distribuita dell'operatore supporta
AgentConfiguration, eseguire il seguente comando:
```

```
 kubectl get crd agentconfigurations.monitoring.netapp.com
Se viene visualizzato il messaggio "Errore dal server (NotFound)", è
necessario aggiornare l'operatore prima di poter utilizzare
AgentConfiguration.
```

Configurazione del supporto proxy

Esistono due posti in cui è possibile utilizzare un proxy sul tenant per installare Kubernetes Monitoring Operator. Possono essere gli stessi sistemi proxy o sistemi proxy separati:

- Proxy necessario durante l'esecuzione dello snippet di codice di installazione (utilizzando "curl") per connettere il sistema in cui viene eseguito lo snippet al tuo ambiente Data Infrastructure Insights
- Proxy necessario al cluster Kubernetes di destinazione per comunicare con l'ambiente Data Infrastructure Insights

Se si utilizza un proxy per uno o entrambi questi elementi, per installare Kubernetes Operating Monitor è necessario innanzitutto assicurarsi che il proxy sia configurato per consentire una buona comunicazione con l'ambiente Data Infrastructure Insights . Se disponi di un proxy e puoi accedere a Data Infrastructure Insights dal server/VM da cui desideri installare l'operatore, è probabile che il tuo proxy sia configurato correttamente.

Per il proxy utilizzato per installare Kubernetes Operating Monitor, prima di installare l'operatore, impostare le variabili di ambiente *http_proxy/https_proxy*. Per alcuni ambienti proxy, potrebbe essere necessario impostare anche la variabile di ambiente *no_proxy*.

Per impostare le variabili, esegui i seguenti passaggi sul tuo sistema **prima** di installare Kubernetes Monitoring Operator:

1. Imposta le variabili di ambiente *https_proxy* e/o *http_proxy* per l'utente corrente:
   a. Se il proxy da configurare non dispone di autenticazione (nome utente/password), eseguire il seguente comando:

```
 export https_proxy=<proxy_server>:<proxy_port>
.. Se il proxy da configurare dispone di autenticazione (nome
utente/password), eseguire questo comando:
```

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

Per far sì che il proxy utilizzato per il cluster Kubernetes comunichi con l'ambiente Data Infrastructure Insights , installare Kubernetes Monitoring Operator dopo aver letto tutte queste istruzioni.

Configurare la sezione proxy di AgentConfiguration in operator-config.yaml prima di distribuire 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
  ...
...
```

Utilizzo di un repository Docker personalizzato o privato

Per impostazione predefinita, Kubernetes Monitoring Operator estrarrà le immagini dei container dal repository Data Infrastructure Insights . Se si utilizza un cluster Kubernetes come destinazione per il monitoraggio e tale cluster è configurato per estrarre immagini di container solo da un repository Docker personalizzato o privato o da un registro di container, è necessario configurare l'accesso ai container necessari all'operatore di monitoraggio Kubernetes.

Eseguire "Image Pull Snippet" dal riquadro di installazione di NetApp Monitoring Operator. Questo comando effettuerà l'accesso al repository Data Infrastructure Insights , estrarrà tutte le dipendenze delle immagini per l'operatore e uscirà dal repository Data Infrastructure Insights . Quando richiesto, immettere la password temporanea del repository fornita. Questo comando scarica tutte le immagini utilizzate dall'operatore, comprese quelle per le funzionalità opzionali. Di seguito sono riportate le funzioni per cui vengono utilizzate queste immagini.

Funzionalità dell'operatore principale e monitoraggio di Kubernetes

- monitoraggio netapp
- ci-kube-rbac-proxy
- ci-ksm
- ci-telegraf

- utente root senza distribuzione

Registro eventi

- ci-fluent-bit
- ci-kubernetes-event-exporter

Prestazioni e mappa della rete

- ci-net-observer

Invia l'immagine Docker dell'operatore al tuo repository Docker privato/locale/aziendale in base alle policy aziendali. Assicurati che i tag delle immagini e i percorsi delle directory di queste immagini nel tuo repository siano coerenti con quelli nel repository Data Infrastructure Insights .

Modifica la distribuzione monitoring-operator in operator-deployment.yaml e modifica tutti i riferimenti alle immagini per utilizzare il tuo repository Docker privato.

```
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>
```

Modifica AgentConfiguration in operator-config.yaml per riflettere la nuova posizione del repository Docker. Crea un nuovo imagePullSecret per il tuo repository privato, per maggiori dettagli consulta *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
```

Istruzioni OpenShift

Se utilizzi OpenShift 4.6 o versione successiva, devi modificare AgentConfiguration in *operator-config.yaml* per abilitare l'impostazione *runPrivileged*:

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

Openshift potrebbe implementare un livello di sicurezza aggiuntivo che potrebbe bloccare l'accesso ad alcuni componenti di Kubernetes.

Tolleranze e difetti

I DaemonSet *netapp-ci-telegraf-ds*, *netapp-ci-fluent-bit-ds* e *netapp-ci-net-observer-l4-ds* devono pianificare un pod su ogni nodo del cluster per raccogliere correttamente i dati su tutti i nodi. L'operatore è stato configurato per tollerare alcune **imperfezioni** ben note. Se hai configurato delle taint personalizzate sui tuoi nodi, impedendo così ai pod di essere eseguiti su ogni nodo, puoi creare una **tolleranza** per quelle taint"nella *AgentConfiguration*" . Se hai applicato taint personalizzati a tutti i nodi del tuo cluster, devi anche aggiungere le tolleranze necessarie alla distribuzione dell'operatore per consentire la pianificazione e l'esecuzione del pod dell'operatore.

Scopri di più su Kubernetes"Contaminazioni e tolleranze" .

Ritorno al"* Pagina di installazione dell'operatore di monitoraggio NetApp Kubernetes*"

Una nota sui segreti

Per rimuovere l'autorizzazione per l'operatore di monitoraggio Kubernetes a visualizzare i segreti a livello di cluster, eliminare le seguenti risorse dal file *operator-setup.yaml* prima dell'installazione:

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

Se si tratta di un aggiornamento, elimina anche le risorse dal tuo cluster:

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

Se l'analisi delle modifiche è abilitata, modificare *AgentConfiguration* o *operator-config.yaml* per rimuovere il commento dalla sezione change-management e includere *kindsToIgnoreFromWatch: '"secrets"'* nella sezione change-management. Notare la presenza e la posizione delle virgolette singole e doppie in questa riga.

```
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"'
  ...
```

Verifica delle firme delle immagini degli operatori di monitoraggio di Kubernetes

L'immagine per l'operatore e tutte le immagini correlate che distribuisce sono firmate da NetApp. È possibile verificare manualmente le immagini prima dell'installazione utilizzando lo strumento di co-firma oppure configurare un controller di ammissione Kubernetes. Per maggiori dettagli si prega di consultare il"Documentazione di Kubernetes" .

La chiave pubblica utilizzata per verificare le firme delle immagini è disponibile nel riquadro di installazione dell'operatore di monitoraggio in *Facoltativo: carica le immagini dell'operatore nel tuo repository privato > Chiave pubblica della firma dell'immagine*

Per verificare manualmente una firma immagine, procedere come segue:

1. Copia ed esegui l'Image Pull Snippet
2. Copia e inserisci la password del repository quando richiesto
3. Memorizza la chiave pubblica della firma dell'immagine (dii-image-signing.pub nell'esempio)
4. Verificare le immagini tramite co-firma. Fare riferimento al seguente esempio di utilizzo del cosign

```
$ 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}]
```

Risoluzione dei problemi

Ecco alcune cose da provare se riscontri problemi durante la configurazione dell'operatore di monitoraggio Kubernetes:

Problema:	Prova questo:
Non vedo alcun collegamento ipertestuale/connessione tra il mio volume persistente Kubernetes e il dispositivo di archiviazione back-end corrispondente. Il mio volume persistente Kubernetes è configurato utilizzando il nome host del server di archiviazione.	Seguire i passaggi per disinstallare l'agente Telegraf esistente, quindi reinstallare l'agente Telegraf più recente. È necessario utilizzare Telegraf versione 2.0 o successiva e l'archiviazione del cluster Kubernetes deve essere monitorata attivamente da Data Infrastructure Insights.

Problema:	Prova questo:
Nei log vedo messaggi simili ai seguenti: E0901 15:21:39.962145 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: Impossibile elencare *v1.MutatingWebhookConfiguration: il server non è riuscito a trovare la risorsa richiesta E0901 15:21:43.168161 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: Impossibile elencare *v1.Lease: il server non è riuscito a trovare la risorsa richiesta (ottenere leases.coordination.k8s.io) ecc.	Questi messaggi possono essere visualizzati se si esegue kube-state-metrics versione 2.0.0 o successiva con versioni di Kubernetes precedenti alla 1.20. Per ottenere la versione di Kubernetes: *kubectl version* Per ottenere la versione di kube-state-metrics: *kubectl get deploy/kube-state-metrics -o jsonpath='{..image}'* Per evitare che questi messaggi si verifichino, gli utenti possono modificare la distribuzione di kube-state-metrics per disabilitare i seguenti lease: *mutatingwebhookconfigurations validatingwebhookconfigurations volumeattachments resources* Più specificamente, possono utilizzare il seguente argomento CLI: resources=certificatesigningrequests,configmaps,cronjobs,daemonsets, deployments,endpoints,horizontalpodautoscalers,ingresses,jobs,limitranges, namespaces,networkpolicies,nodes,persistentvolumeclaims,persistentvolumes, poddisruptionbudgets,pods,replicasets,replicationcontrollers,resourcequotas, secrets,services,statefulsets,storageclasses L’elenco di risorse predefinito è: "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"

Problema:	Prova questo:
Vedo messaggi di errore da Telegraf simili ai seguenti, ma Telegraf si avvia ed è in esecuzione: 11 ott 14:23:41 ip-172-31-39-47 systemd[1]: Avviato L'agente server basato su plugin per la segnalazione delle metriche in InfluxDB. 11 ott 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="impossibile creare la directory della cache. /etc/telegraf/.cache/snowflake, err: mkdir /etc/telegraf/.ca che: permesso negato. ignorato\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" 11 ott 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="apertura non riuscita. Ignorato. Apri /etc/telegraf/.cache/snowflake/ocsp_response_cache.json: nessun file o directory\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" 11 ott 14:23:41 ip-172-31-39-47 telegraf[1827]: 2021-10-11T14:23:41Z I! Avvio di Telegraf 1.19.3	Questo è un problema noto. Fare riferimento a"Questo articolo di GitHub" per maggiori dettagli. Finché Telegraf è attivo e funzionante, gli utenti possono ignorare questi messaggi di errore.
Su Kubernetes, i miei pod Telegraf segnalano il seguente errore: "Errore nell'elaborazione delle informazioni mountstats: impossibile aprire il file mountstats: /hostfs/proc/1/mountstats, errore: apertura /hostfs/proc/1/mountstats: autorizzazione negata"	Se SELinux è abilitato e applicato, è probabile che impedisca ai pod Telegraf di accedere al file /proc/1/mountstats sul nodo Kubernetes. Per superare questa restrizione, modificare agentconfiguration e abilitare l'impostazione runPrivileged. Per maggiori dettagli, fare riferimento alle istruzioni di OpenShift.
Su Kubernetes, il mio pod Telegraf ReplicaSet segnala il seguente errore: [inputs.prometheus] Errore nel plugin: impossibile caricare la coppia di chiavi /etc/kubernetes/pki/etcd/server.crt:/etc/kubernetes/pki/etcd/server.key: aprire /etc/kubernetes/pki/etcd/server.crt: nessun file o directory del genere	Il pod Telegraf ReplicaSet è progettato per essere eseguito su un nodo designato come master o per etcd. Se il pod ReplicaSet non è in esecuzione su uno di questi nodi, verranno visualizzati questi errori. Controlla se i tuoi nodi master/etcd presentano delle anomalie. In tal caso, aggiungere le tolleranze necessarie al Telegraf ReplicaSet, telegraf-rs. Ad esempio, modifica ReplicaSet… kubectl edit rs telegraf-rs …e aggiungi le tolleranze appropriate alla specifica. Quindi, riavviare il pod ReplicaSet.

Problema:	Prova questo:
Ho un ambiente PSP/PSA. Ciò ha ripercussioni sul mio operatore di monitoraggio?	Se il cluster Kubernetes è in esecuzione con Pod Security Policy (PSP) o Pod Security Admission (PSA), è necessario eseguire l'aggiornamento alla versione più recente di Kubernetes Monitoring Operator. Per effettuare l'aggiornamento all'operatore corrente con supporto per PSP/PSA, seguire questi passaggi: 1. Disinstallare l'operatore di monitoraggio precedente: 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. Installare l'ultima versione dell'operatore di monitoraggio.
Ho riscontrato problemi nel tentativo di distribuire l'Operatore e sto utilizzando PSP/PSA.	1. Modificare l'agente utilizzando il seguente comando: kubectl -n <name-space> edit agent 2. Contrassegna 'security-policy-enabled' come 'false'. In questo modo verranno disattivati i criteri di sicurezza del Pod e l'ammissione di sicurezza del Pod e sarà consentito all'operatore di effettuare la distribuzione. Confermare utilizzando i seguenti comandi: kubectl get psp (dovrebbe mostrare che la politica di sicurezza del pod è stata rimossa) kubectl get all -n <namespace>
grep -i psp (dovrebbe mostrare che non è stato trovato nulla)	Errori "ImagePullBackoff" rilevati
Questi errori potrebbero verificarsi se si dispone di un repository Docker personalizzato o privato e non è ancora stato configurato Kubernetes Monitoring Operator per riconoscerlo correttamente. Per saperne di più sulla configurazione per repository personalizzati/privati.	Ho un problema con la distribuzione del mio operatore di monitoraggio e la documentazione attuale non mi aiuta a risolverlo.

Problema:	Prova questo:

Acquisire o annotare in altro modo l'output dei seguenti comandi e contattare il team di supporto tecnico.
 	
`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`
	I pod net-observer (Workload Map) nello spazio dei nomi Operator sono in CrashLoopBackOff

Questi pod corrispondono al raccoglitore di dati Workload Map per Network Observability. Prova questi: • Controlla i log di uno dei pod per confermare la versione minima del kernel. Ad esempio: ---- {"ci-tenant-id":"your-tenant-id","collector-cluster":"your-k8s-cluster-name","environment":"prod","level":"error","msg":"fallimento nella convalida. Motivo: la versione del kernel 3.10.0 è inferiore alla versione minima del kernel 4.18.0","time":"2022-11-09T08:23:08Z"} ---- • I pod Net-observer richiedono che la versione del kernel Linux sia almeno 4.18.0. Controllare la versione del kernel utilizzando il comando "uname -r" e assicurarsi che sia >= 4.18.0	I pod sono in esecuzione nello spazio dei nomi Operatore (predefinito: netapp-monitoring), ma nell'interfaccia utente non vengono visualizzati dati per la mappa del carico di lavoro o metriche Kubernetes nelle query
Controllare l'impostazione dell'ora sui nodi del cluster K8S. Per un audit e una segnalazione dei dati accurati, si consiglia vivamente di sincronizzare l'ora sulla macchina dell'agente utilizzando il protocollo NTP (Network Time Protocol) o il protocollo SNTP (Simple Network Time Protocol).	Alcuni dei pod net-observer nello spazio dei nomi Operator sono nello stato In sospeso
Net-observer è un DaemonSet ed esegue un pod in ogni nodo del cluster k8s. • Prendi nota del pod che si trova nello stato In sospeso e controlla se sta riscontrando un problema di risorse per la CPU o la memoria. Assicurarsi che nel nodo siano disponibili la memoria e la CPU richieste.	Subito dopo aver installato Kubernetes Monitoring Operator, vedo quanto segue nei miei log: [inputs.prometheus] Errore nel plugin: errore durante la richiesta HTTP a http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: Ottieni http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: dial tcp: cerca kube-state-metrics.<namespace>.svc.cluster.local: nessun host del genere

Problema:	Prova questo:
In genere questo messaggio viene visualizzato solo quando viene installato un nuovo operatore e il pod *telegraf-rs* è attivo prima del pod *ksm*. Questi messaggi dovrebbero cessare una volta che tutti i pod saranno in esecuzione.	Non vedo alcuna metrica raccolta per i CronJob di Kubernetes presenti nel mio cluster.
Verifica la tua versione di Kubernetes (ad esempio `kubectl version`). Se la versione è v1.20.x o precedente, si tratta di una limitazione prevista. La versione kube-state-metrics distribuita con Kubernetes Monitoring Operator supporta solo v1.CronJob. Con Kubernetes 1.20.x e versioni precedenti, la risorsa CronJob si trova in v1beta.CronJob. Di conseguenza, kube-state-metrics non riesce a trovare la risorsa CronJob.	Dopo aver installato l'operatore, i pod telegraf-ds entrano in CrashLoopBackOff e i log dei pod indicano "su: Authentication failure".
Modificare la sezione telegraf in *AgentConfiguration* e impostare *dockerMetricCollectionEnabled* su false. Per maggiori dettagli fare riferimento al manuale dell'operatore"opzioni di configurazione" . … spec: … telegraf: … - nome: docker modalità di esecuzione: - sostituzioni DaemonSet: - chiave: DOCKER_UNIX_SOCK_PLACEHOLDER valore: unix:///run/docker.sock … …	Nei miei registri di Telegraf vedo messaggi di errore ricorrenti simili ai seguenti: E! [agente] Errore durante la scrittura su output.http: Post "https://<tenant_url>/rest/v1/lake/ingest/influxdb": scadenza del contesto superata (Client.Timeout superato durante l'attesa delle intestazioni)
Modificare la sezione telegraf in *AgentConfiguration* e aumentare *outputTimeout* a 10 s. Per maggiori dettagli fare riferimento al manuale dell'operatore"opzioni di configurazione" .	Mancano i dati *involvedobject* per alcuni registri eventi.
Assicurati di aver seguito i passaggi indicati in"Permessi" sezione sopra.	Perché vedo due pod di operatori di monitoraggio in esecuzione, uno denominato netapp-ci-monitoring-operator-<pod> e l'altro denominato monitoring-operator-<pod>?
A partire dal 12 ottobre 2023, Data Infrastructure Insights ha riorganizzato l'operatore per servire meglio i nostri utenti; affinché tali modifiche vengano adottate completamente, è necessariorimuovere il vecchio operatore Einstallare quello nuovo .	I miei eventi Kubernetes hanno smesso inaspettatamente di segnalare a Data Infrastructure Insights.
Recupera il nome del pod event-exporter: `` `kubectl -n netapp-monitoring get pods ``	grep event-exporter

Problema:	Prova questo:
awk '{print $1}'	sed 's/event-exporter./event-exporter/` Dovrebbe essere "netapp-ci-event-exporter" o "event-exporter". Successivamente, modifica l'agente di monitoraggio `kubectl -n netapp-monitoring edit agent` e impostare il valore per LOG_FILE in modo che rifletta il nome appropriato del pod di esportazione eventi trovato nel passaggio precedente. Più specificatamente, LOG_FILE dovrebbe essere impostato su "/var/log/containers/netapp-ci-event-exporter.log" o "/var/log/containers/event-exporter*.log" …. fluent-bit: … - name: event-exporter-ci substitutions: - key: LOG_FILE values: - /var/log/containers/netapp-ci-event-exporter*.log … …. In alternativa, si può anchedisinstallare Ereinstallare l'agente.
Vedo che i pod distribuiti dal Kubernetes Monitoring Operator si bloccano a causa di risorse insufficienti.	Fare riferimento all'operatore di monitoraggio Kubernetes"opzioni di configurazione" per aumentare i limiti della CPU e/o della memoria secondo necessità.
Un'immagine mancante o una configurazione non valida hanno impedito l'avvio o la disponibilità dei pod netapp-ci-kube-state-metrics. Ora StatefulSet è bloccato e le modifiche alla configurazione non vengono applicate ai pod netapp-ci-kube-state-metrics.	Lo StatefulSet è in un"rotto" stato. Dopo aver risolto eventuali problemi di configurazione, riavviare i pod netapp-ci-kube-state-metrics.
I pod netapp-ci-kube-state-metrics non riescono ad avviarsi dopo aver eseguito un aggiornamento dell'operatore Kubernetes, generando l'errore ErrImagePull (impossibilità di estrarre l'immagine).	Prova a reimpostare manualmente i pod.
Durante l'analisi dei log, vengono visualizzati i messaggi "Evento scartato perché più vecchio di maxEventAgeSeconds" per il mio cluster Kubernetes.	Modificare l'operatore *agentconfiguration* e aumentare *event-exporter-maxEventAgeSeconds* (ad esempio a 60 s), *event-exporter-kubeQPS* (ad esempio a 100) e *event-exporter-kubeBurst* (ad esempio a 500). Per maggiori dettagli su queste opzioni di configurazione, vedere"opzioni di configurazione" pagina.

Problema:	Prova questo:
Telegraf avvisa o si blocca a causa di una memoria bloccabile insufficiente.	Prova ad aumentare il limite di memoria bloccabile per Telegraf nel sistema operativo/nodo sottostante. Se aumentare il limite non è un'opzione, modificare la configurazione dell'agente NKMO e impostare *unprotected* su *true*. Ciò indicherà a Telegraf di non tentare di riservare pagine di memoria bloccate. Sebbene ciò possa rappresentare un rischio per la sicurezza, in quanto i segreti decrittati potrebbero essere trasferiti su disco, consente l'esecuzione in ambienti in cui non è possibile riservare memoria bloccata. Per maggiori dettagli sulle opzioni di configurazione *non protette*, fare riferimento a"opzioni di configurazione" pagina.
Vedo messaggi di avviso da Telegraf simili ai seguenti: *W! [inputs.diskio] Impossibile raccogliere il nome del disco per "vdc": errore durante la lettura di /dev/vdc: nessun file o directory del genere*	Per l'operatore di monitoraggio di Kubernetes, questi messaggi di avviso sono innocui e possono essere tranquillamente ignorati. In alternativa, modificare la sezione telegraf in AgentConfiguration e impostare *runDsPrivileged* su true. Per maggiori dettagli fare riferimento al"opzioni di configurazione dell'operatore" .

Problema:	Prova questo:

Il mio pod fluent-bit non funziona con i seguenti errori: [2024/10/16 14:16:23] [errore] [/src/fluent-bit/plugins/in_tail/tail_fs_inotify.c:360 errno=24] Troppi file aperti [2024/10/16 14:16:23] [errore] inizializzazione input tail.0 non riuscita [2024/10/16 14:16:23] [errore] [motore] inizializzazione input non riuscita	Prova a modificare le impostazioni *fsnotify* nel tuo 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>`
	Riavvia Fluent-bit.
	Nota: per rendere queste impostazioni persistenti tra i riavvii del nodo, è necessario inserire le seguenti righe in */etc/sysctl.conf*
	`fs.inotify.max_user_instances=<something larger than current setting>`
	`fs.inotify.max_user_watches=<something larger than current setting>`

Problema:	Prova questo:
I pod DS di Telegraf segnalano errori relativi al plugin di input Kubernetes che non riesce a effettuare richieste HTTP a causa dell’impossibilità di convalidare il certificato TLS. Ad esempio: E! [inputs.kubernetes] Errore nel plugin: errore durante la richiesta HTTP a"<a href="https://<kubelet_IP>:10250/stats/summary": " class="bare">https://<kubelet_IP>:10250/stats/summary": Ottenere"<a href="https://<kubelet_IP>:10250/stats/summary": " class="bare">https://<kubelet_IP>:10250/stats/summary": tls: impossibile verificare il certificato: x509: impossibile convalidare il certificato per <kubelet_IP> perché non contiene alcun IP SAN	Ciò si verifica se il kubelet utilizza certificati autofirmati e/o il certificato specificato non include <kubelet_IP> nell'elenco *Subject Alternative Name* dei certificati. Per risolvere questo problema, l'utente può modificare il"configurazione dell'agente" e impostare *telegraf:insecureK8sSkipVerify* su *true*. In questo modo il plugin di input Telegraf verrà configurato per saltare la verifica. In alternativa, l'utente può configurare il kubelet per"serverTLSBootstrap" , che attiverà una richiesta di certificato dall'API 'certificates.k8s.io'.

Ulteriori informazioni possono essere trovate presso"Supporto" pagina o nella"Matrice di supporto del raccoglitore dati" .

Opzioni di configurazione dell'operatore di monitoraggio Kubernetes

IL "Operatore di monitoraggio Kubernetes" Offre ampie opzioni di personalizzazione tramite il file AgentConfiguration. È possibile configurare limiti di risorse, intervalli di raccolta, impostazioni proxy, tolleranze e impostazioni specifiche per ogni componente per ottimizzare il monitoraggio del proprio ambiente Kubernetes. Utilizzare queste opzioni per personalizzare telegraf, kube-state-metrics, la raccolta dei log, la mappatura dei carichi di lavoro, la gestione delle modifiche e altri componenti di monitoraggio.

File di configurazione dell'agente di esempio

Di seguito è riportato un esempio di file *AgentConfiguration*, con descrizioni per ciascuna opzione.

```
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
```

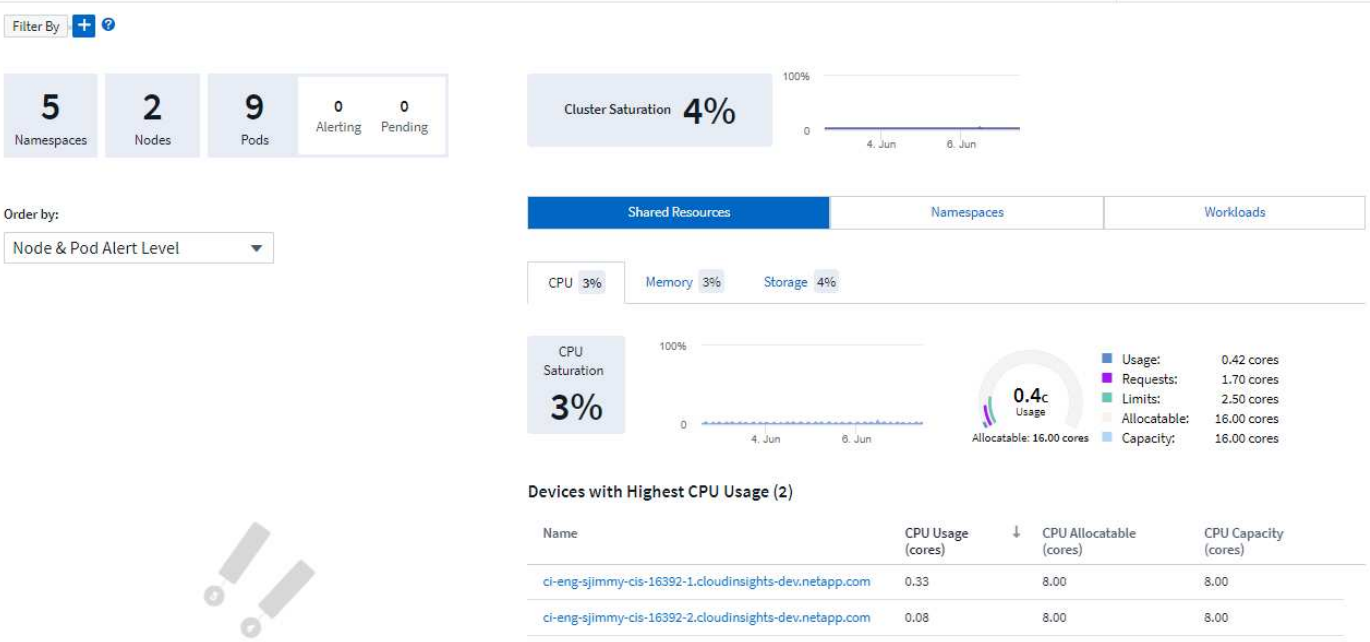
```
    ## Set bpfPollInterval between 3 and 15
    ##
    # bpfPollInterval: '8'

    ##
    ## 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: ''
```

Pagina dei dettagli del cluster Kubernetes

La pagina dei dettagli del cluster Kubernetes mostra una panoramica dettagliata del tuo cluster Kubernetes.



Conteggio di spazi dei nomi, nodi e pod

I conteggi nella parte superiore della pagina mostrano il numero totale di namespace, nodi e pod nel cluster, nonché il numero di popd attualmente in stato di avviso e in sospeso.

Risorse condivise e saturazione

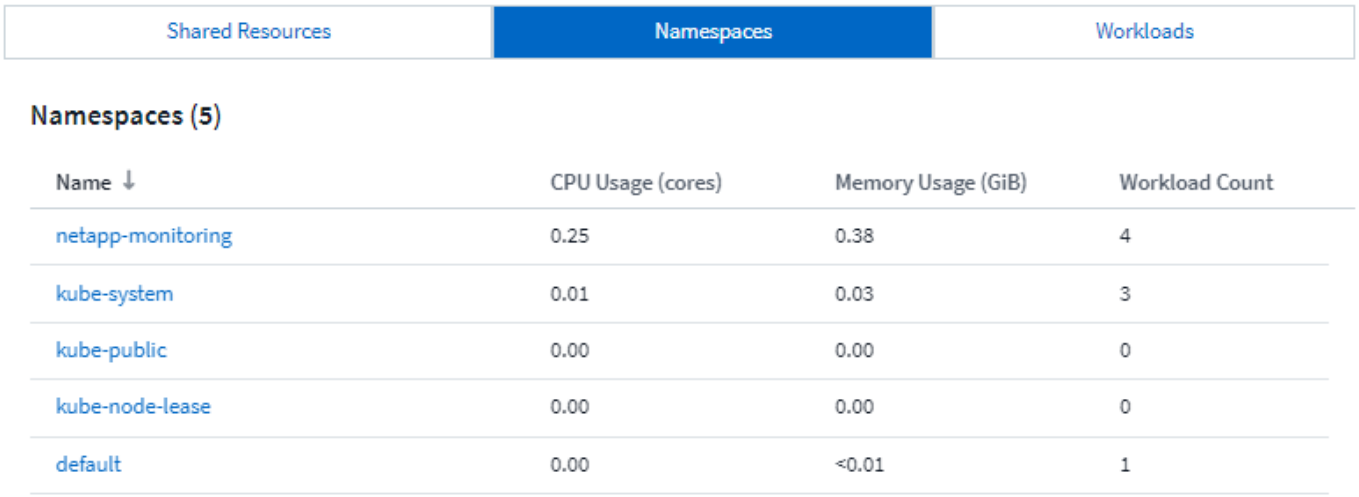
In alto a destra della pagina dei dettagli è visualizzata la saturazione del cluster in percentuale, nonché un grafico che mostra l'andamento recente nel tempo. La saturazione del cluster è il livello più alto di saturazione della CPU, della memoria o dello storage in ogni momento.

Al di sotto, la pagina mostra per impostazione predefinita l'utilizzo delle **Risorse condivise**, con schede per CPU, Memoria e Archiviazione. Ogni scheda mostra la percentuale di saturazione e l'andamento nel tempo, con ulteriori dettagli sull'utilizzo. Per l'archiviazione, il valore mostrato è il maggiore tra la saturazione del backend e quella del file system, calcolate in modo indipendente.

I dispositivi con il maggiore utilizzo sono mostrati in una tabella in basso. Clicca su un link qualsiasi per esplorare questi dispositivi.

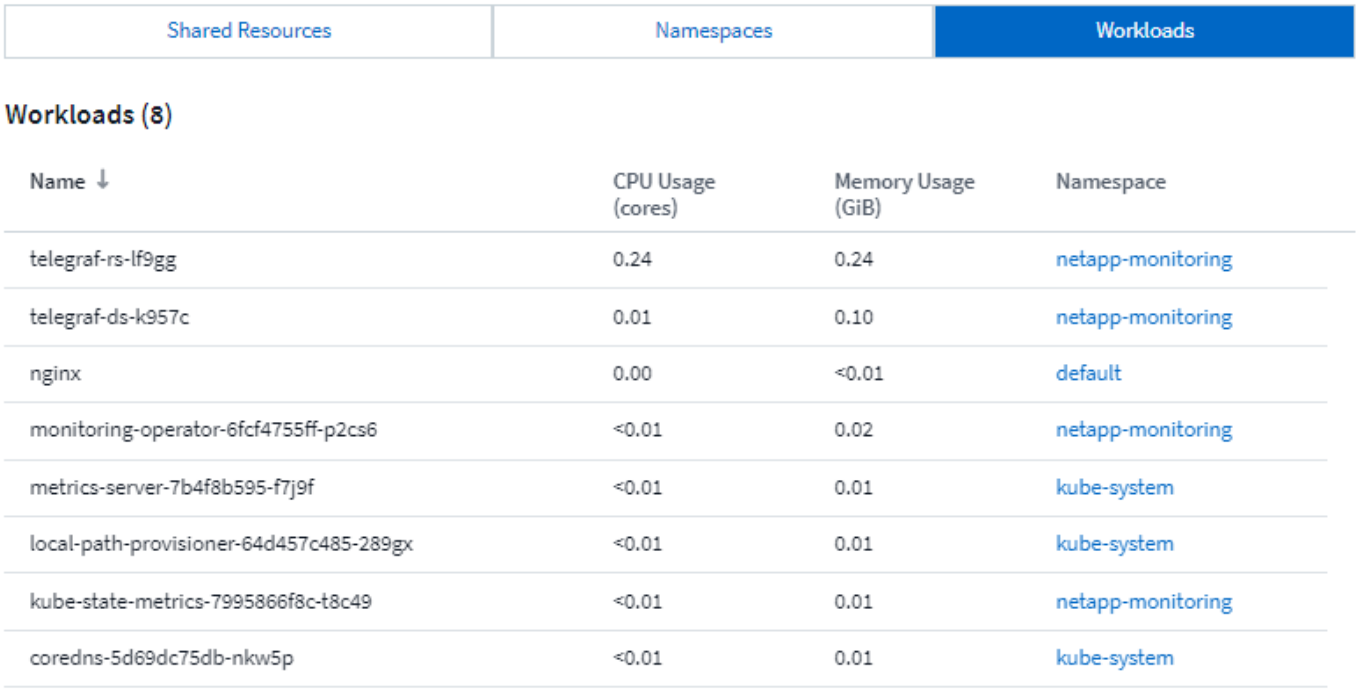
Spazi dei nomi

La scheda Namespace visualizza un elenco di tutti i namespace presenti nell'ambiente Kubernetes, mostrando l'utilizzo di CPU e memoria, nonché il conteggio dei carichi di lavoro in ciascun namespace. Fare clic sui collegamenti Nome per esplorare ogni spazio dei nomi.



Carichi di lavoro

Allo stesso modo, la scheda Carichi di lavoro visualizza un elenco dei carichi di lavoro in ogni namespace, mostrando ancora una volta l'utilizzo di CPU e memoria. Facendo clic sui collegamenti Namespace si accede a ciascuno di essi.



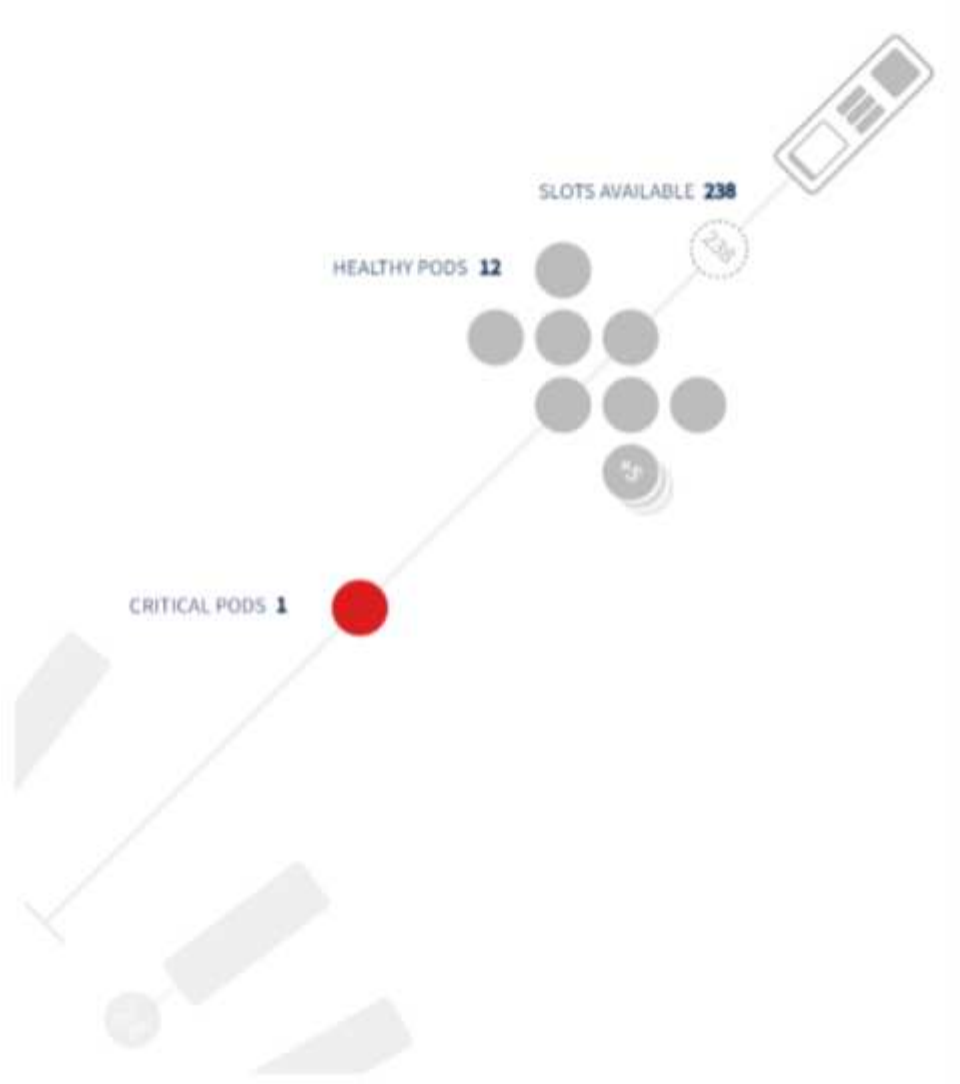
La "Ruota" del Cluster

La sezione "Ruota" del cluster fornisce una panoramica sullo stato di salute dei nodi e dei pod, su cui è possibile approfondire per ottenere maggiori informazioni. Se il cluster contiene più nodi di quanti ne possano essere visualizzati in quest'area della pagina, sarà possibile girare la rotellina utilizzando i pulsanti disponibili.

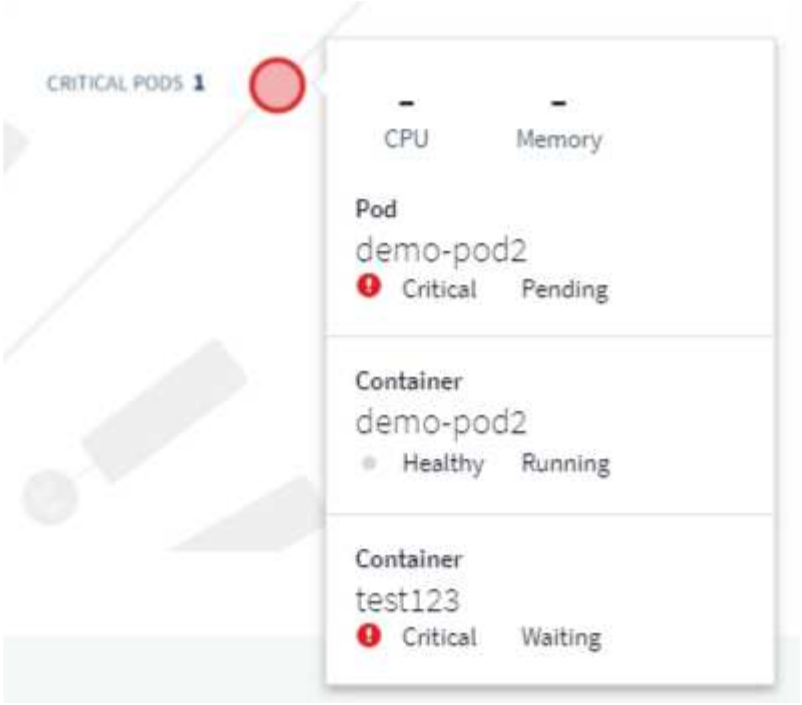
I pod o i nodi di avviso vengono visualizzati in rosso. Le aree di "Avvertenza" sono visualizzate in arancione. I pod non programmati (ovvero non collegati) verranno visualizzati nell'angolo inferiore della "Ruota" del cluster.

Passando il mouse sopra un pod (cerchio) o un nodo (barra) si estenderà la vista del nodo.

Facendo clic sul pod o sul nodo in quella vista, si ingrandirà la vista del nodo espansa.



Da qui puoi passare il mouse su un elemento per visualizzarne i dettagli. Ad esempio, passando il mouse sul pod critico in questo esempio vengono visualizzati i dettagli su quel pod.



È possibile visualizzare le informazioni su file system, memoria e CPU passando il mouse sugli elementi Node.

Una nota sugli indicatori

Gli indicatori di memoria e CPU mostrano tre colori, poiché mostrano la capacità *utilizzata* in relazione sia alla capacità *allocabile* che alla capacità *totale*.

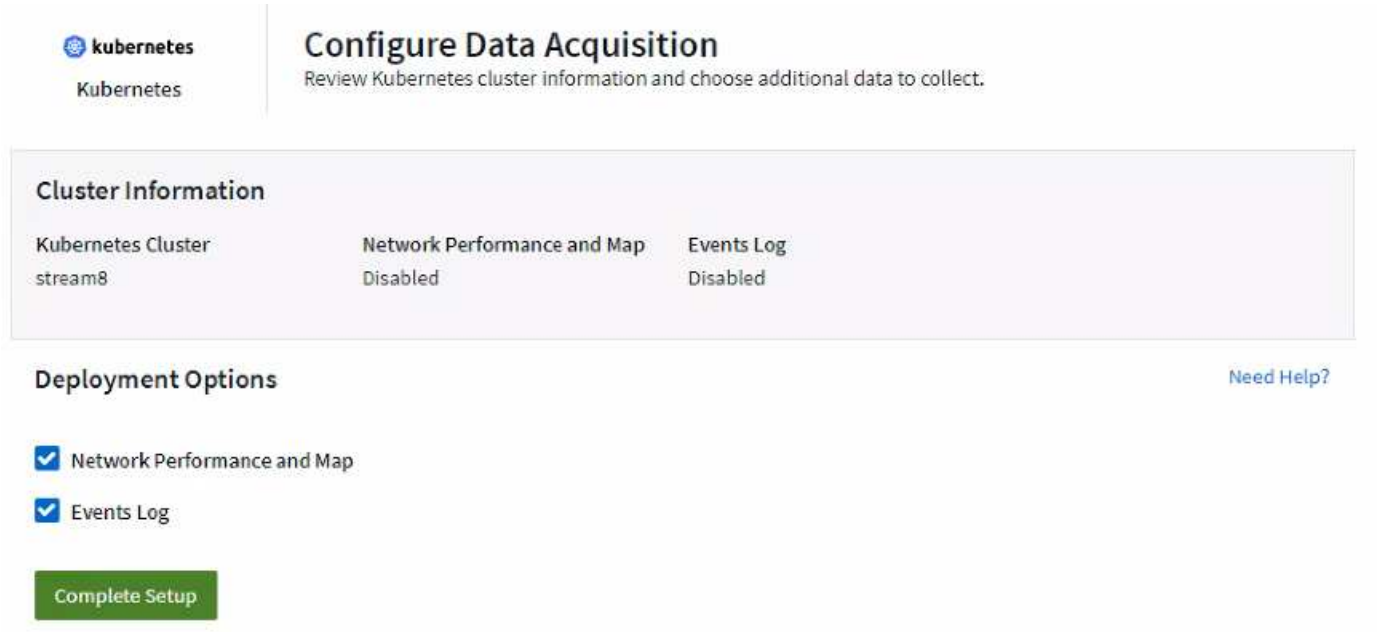
Monitoraggio e mappa delle prestazioni della rete Kubernetes

La funzionalità Kubernetes Network Performance Monitoring and Map semplifica la risoluzione dei problemi mappando le dipendenze tra i servizi (chiamati anche carichi di lavoro) e fornisce visibilità in tempo reale sulle latenze e sulle anomalie delle prestazioni di rete per identificare i problemi di prestazioni prima che influiscano sugli utenti. Questa funzionalità aiuta le organizzazioni a ridurre i costi complessivi analizzando e verificando i flussi di traffico di Kubernetes.

Caratteristiche principali: • La mappa del carico di lavoro presenta le dipendenze e i flussi del carico di lavoro di Kubernetes e mette in evidenza i problemi di rete e prestazioni. • Monitora il traffico di rete tra pod, carichi di lavoro e nodi Kubernetes; identifica l'origine del traffico e i problemi di latenza. • Ridurre i costi complessivi analizzando il traffico di rete in ingresso, in uscita, tra regioni e tra zone.

Prerequisiti

Prima di poter utilizzare Kubernetes Network Performance Monitoring and Map, è necessario aver configurato"Operatore di monitoraggio NetApp Kubernetes" per abilitare questa opzione. Durante l'implementazione dell'operatore, selezionare la casella di controllo "Prestazioni e mappa di rete" per abilitarla. Puoi anche abilitare questa opzione andando alla landing page di Kubernetes e selezionando "Modifica distribuzione".



Monitor

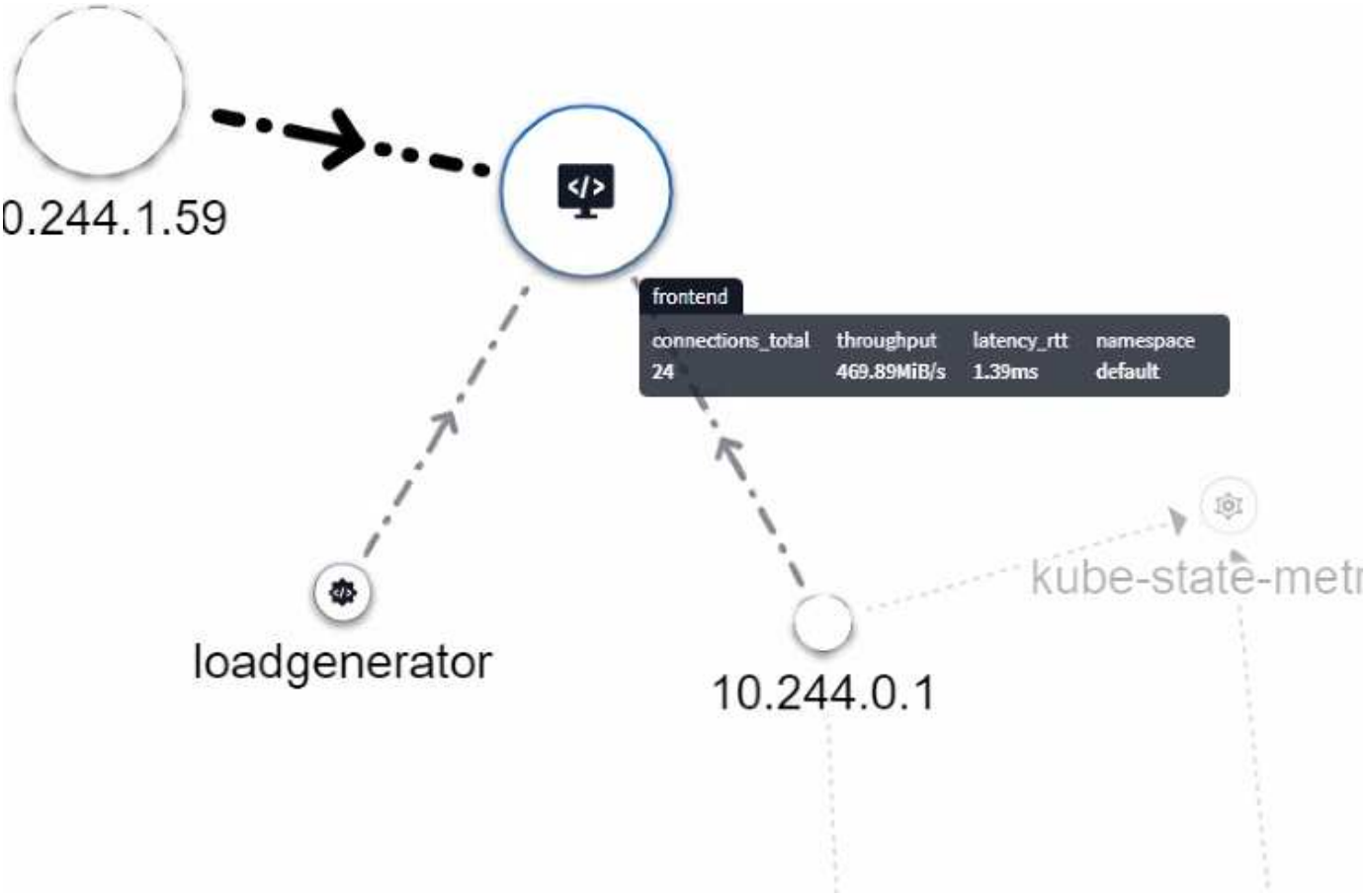
La mappa del carico di lavoro utilizza"monitor" per ricavare informazioni. Data Infrastructure Insights fornisce una serie di monitor Kubernetes predefiniti (si noti che questi potrebbero essere *in pausa* per impostazione predefinita). È possibile *Riprendere* (ovvero abilitare) i monitor desiderati oppure creare monitor personalizzati per gli oggetti Kubernetes, che verranno utilizzati anche da Workload Map.

È possibile creare avvisi sulle metriche di Data Infrastructure Insights su uno qualsiasi dei tipi di oggetti indicati di seguito. Assicurarsi che i dati siano raggruppati in base al tipo di oggetto predefinito.

- carico di lavoro di kubernetes
- kubernetes.daemonset
- distribuzione di kubernetes
- kubernetes.cronjob
- kubernetes.job
- kubernetes.replicaset
- kubernetes.statefulset
- kubernetes.pod
- kubernetes.network_traffic_l4

La mappa

La mappa mostra i servizi/carichi di lavoro e le loro relazioni reciproche. Le frecce indicano le direzioni del traffico. Passando il mouse sopra un carico di lavoro vengono visualizzate informazioni riepilogative per quel carico di lavoro, come puoi vedere in questo esempio:



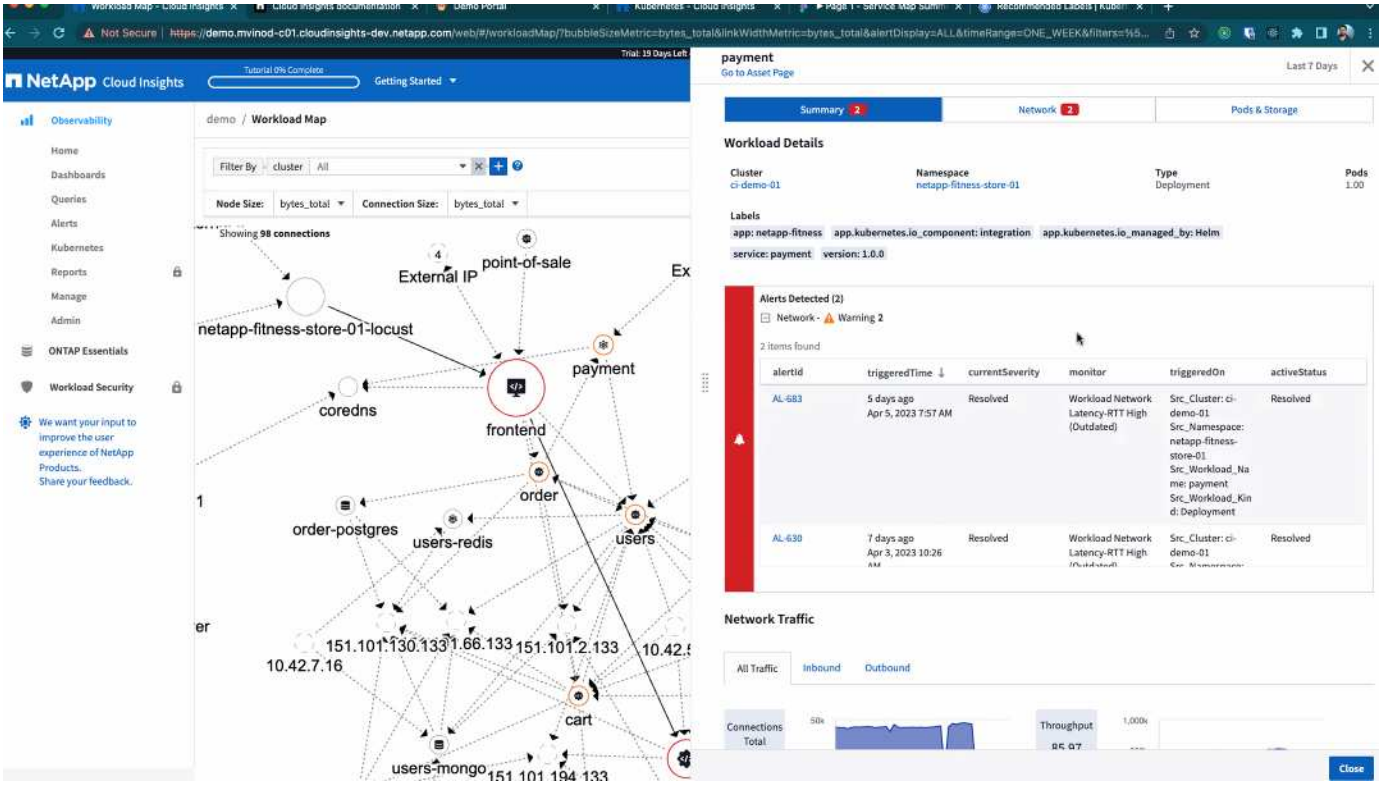
Le icone all'interno dei cerchi rappresentano diversi tipi di servizio. Si noti che le icone sono visibili solo se gli oggetti sottostanti hannoetichette .

La dimensione di ogni cerchio indica la dimensione del nodo. Tieni presente che queste dimensioni sono relative: il livello di zoom del tuo browser o le dimensioni dello schermo potrebbero influire sulle dimensioni effettive dei cerchi. Allo stesso modo, lo stile della linea di traffico fornisce una visione immediata delle dimensioni della connessione: le linee continue in grassetto indicano un traffico elevato, mentre le linee tratteggiate chiare indicano un traffico inferiore.

I numeri all'interno dei cerchi indicano il numero di connessioni esterne attualmente elaborate dal servizio.

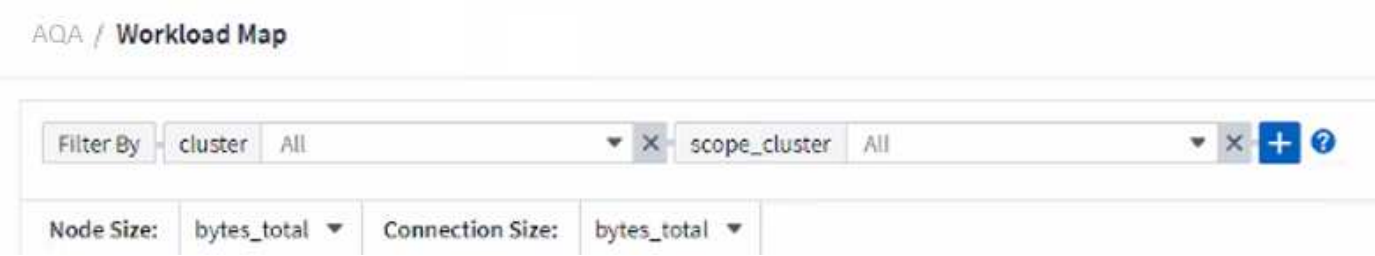
Dettagli e avvisi sul carico di lavoro

I cerchi colorati indicano un avviso di livello critico o di avvertimento per il carico di lavoro. Passa il mouse sul cerchio per un riepilogo del problema oppure clicca sul cerchio per aprire un pannello scorrevole con maggiori dettagli.

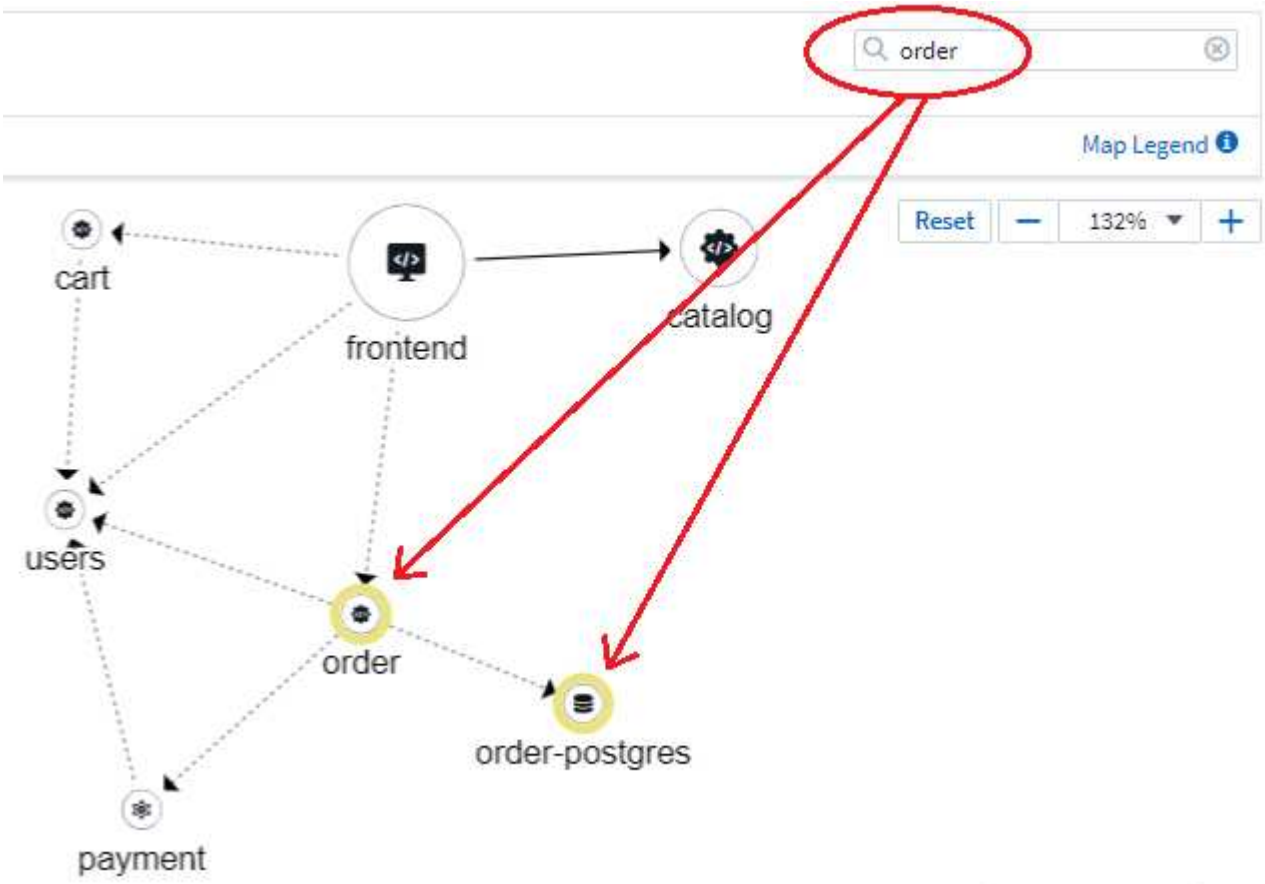


Ricerca e filtraggio

Come con altre funzionalità Data Infrastructure Insights , puoi facilmente impostare filtri per concentrarti sugli oggetti specifici o sugli attributi del carico di lavoro desiderati.



Allo stesso modo, digitando una stringa nel campo *Trova* verranno evidenziati i carichi di lavoro corrispondenti.



Etichette del carico di lavoro

Le etichette dei carichi di lavoro sono necessarie se si desidera che la mappa identifichi i tipi di carichi di lavoro visualizzati (ad esempio le icone circolari). Le etichette vengono derivate come segue:

- Nome del servizio/applicazione in esecuzione in termini generici
- Se la sorgente è un pod:
  - L'etichetta deriva dall'etichetta del carico di lavoro del pod
  - Etichetta prevista sul carico di lavoro: app.kubernetes.io/component
  - Riferimento nome etichetta: https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels/
  - Etichette consigliate:
    - fine frontale

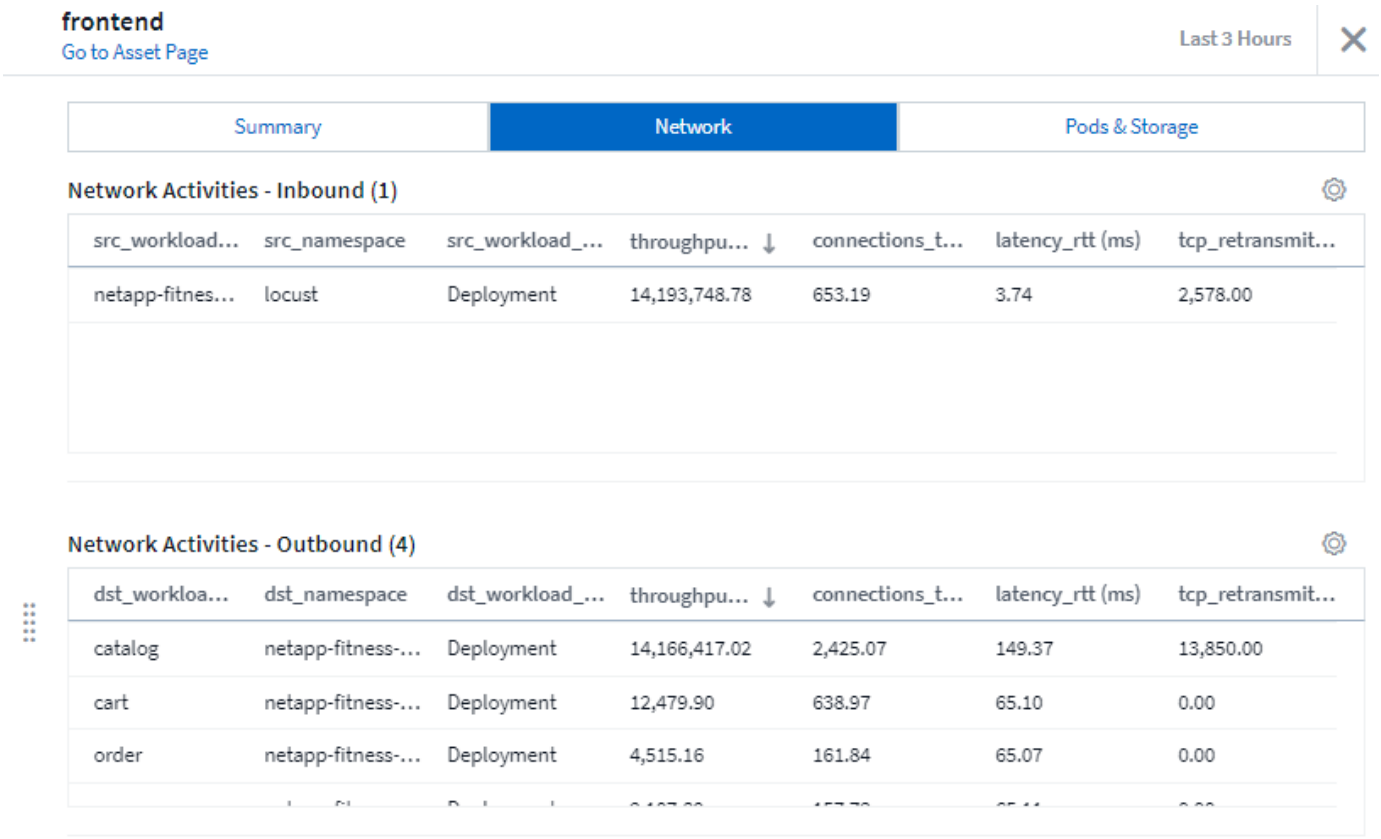
        - backend
        - banca dati
        - nascondiglio
        - coda
        - Kafka
- Se la sorgente è esterna al cluster Kubernetes:
    - Data Infrastructure Insights tenterà di analizzare il nome DNS risolto per estrarre il tipo di servizio.

      Ad esempio, con un nome DNS risolto di *s3.eu-north-1.amazonaws.com*, il nome risolto viene analizzato per ottenere *s3* come tipo di servizio.

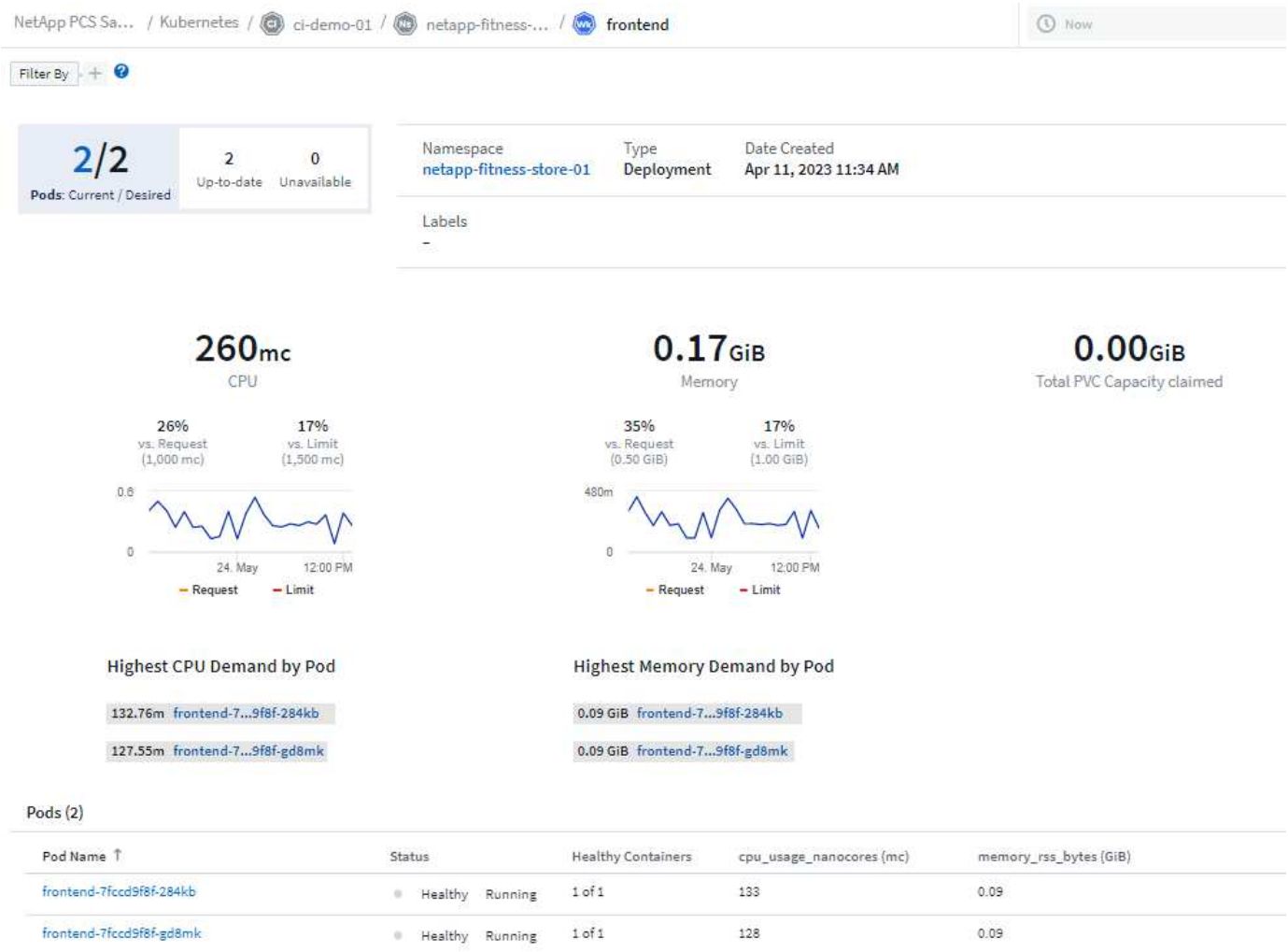
Immergiti in profondità

Facendo clic con il pulsante destro del mouse su un carico di lavoro vengono visualizzate opzioni aggiuntive da esplorare ulteriormente. Ad esempio, da qui è possibile ingrandire per visualizzare le connessioni per quel carico di lavoro.

In alternativa, è possibile aprire il pannello scorrevole dei dettagli per visualizzare direttamente la scheda *Riepilogo*, *Rete* o *Pod e archiviazione*.



Infine, selezionando *Vai alla pagina delle risorse* si aprirà la pagina di destinazione dettagliata delle risorse per il carico di lavoro.



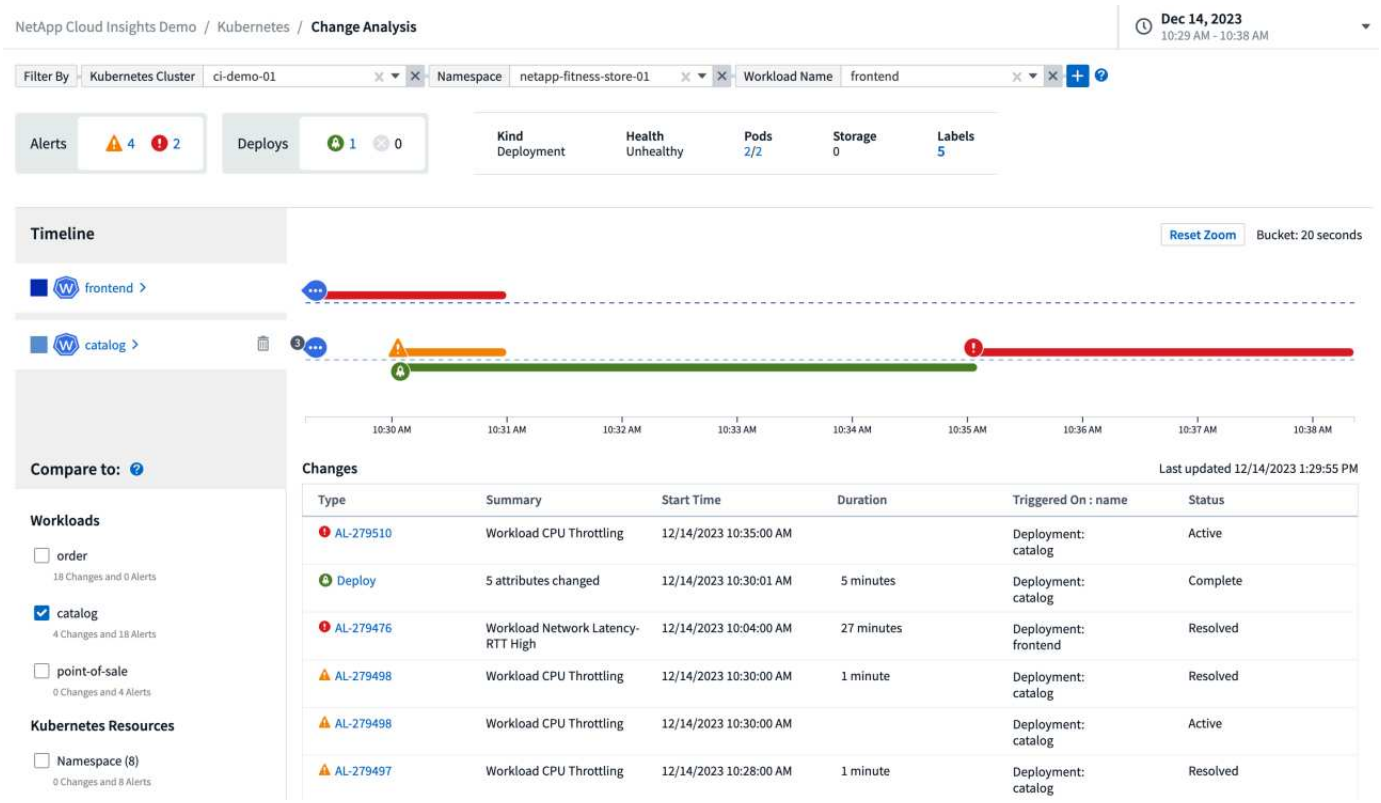
Analisi delle modifiche di Kubernetes

Kubernetes Change Analytics ti fornisce una visione completa delle modifiche recenti apportate al tuo ambiente K8s. Avvisi e stato di distribuzione sono a portata di mano. Con Change Analytics puoi monitorare ogni modifica di distribuzione e configurazione e correlarla allo stato di salute e alle prestazioni dei servizi, dell'infrastruttura e dei cluster di K8.

In che modo l'analisi del cambiamento può essere utile?

- Negli ambienti Kubernetes multi-tenant, le interruzioni possono verificarsi a causa di modifiche configurate in modo errato. Change Analytics aiuta in questo senso fornendo un unico riquadro per visualizzare e correlare lo stato dei carichi di lavoro e le modifiche alla configurazione. Ciò può aiutare nella risoluzione dei problemi degli ambienti Kubernetes dinamici.

Per visualizzare Kubernetes Change Analytics, vai su **Kubernetes > Change Analysis**.



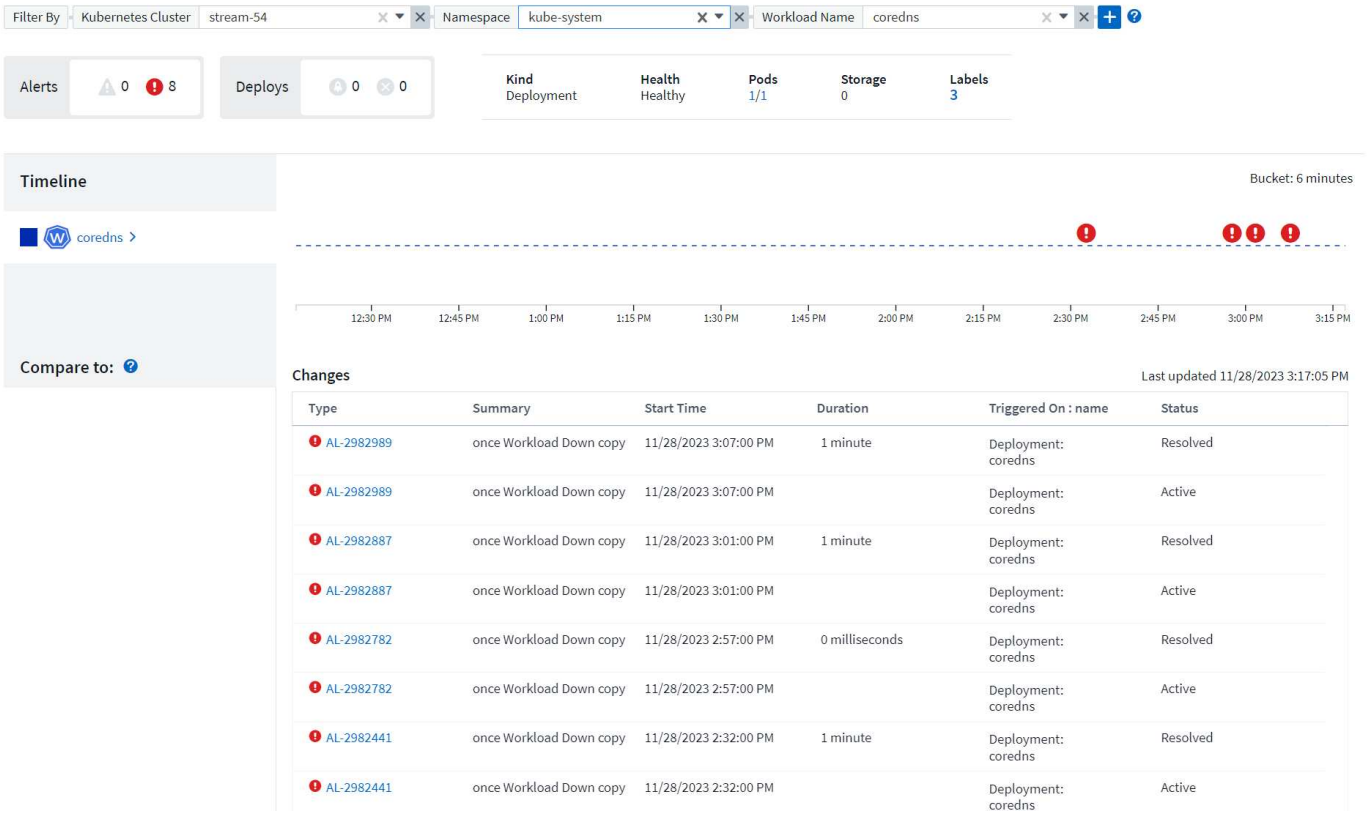
.

La pagina si aggiorna automaticamente in base all'intervallo di tempo Data Infrastructure Insights attualmente selezionato. Intervalli di tempo più brevi implicano aggiornamenti dello schermo più frequenti.

Filtraggio

Come per tutte le funzionalità di Data Infrastructure Insights, filtrare l'elenco delle modifiche è intuitivo: nella parte superiore della pagina, inserisci o seleziona i valori per il tuo cluster Kubernetes, namespace o workload, oppure aggiungi i tuoi filtri selezionando il pulsante {+].

Quando si filtra in base a un cluster, uno spazio dei nomi e un carico di lavoro specifici (insieme a tutti gli altri filtri impostati), viene visualizzata una cronologia delle distribuzioni e degli avvisi per quel carico di lavoro in quello spazio dei nomi su quel cluster. Per ingrandire ulteriormente la visualizzazione, fare clic e trascinare il grafico per concentrarsi su un intervallo di tempo più specifico.

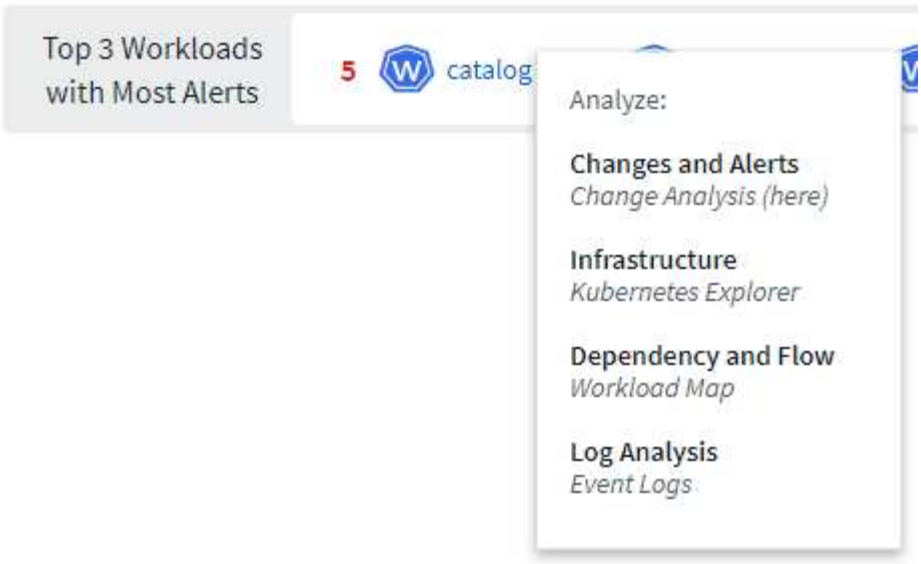


Stato rapido

Al di sotto dell'area di filtraggio si trovano una serie di indicatori di alto livello. Sulla sinistra è riportato il numero di avvisi (Avviso e Critico). Questo numero include sia gli avvisi *Attivi* che quelli *Risolti*. Per visualizzare solo gli avvisi *Attivi*, imposta un filtro per "Stato" e scegli "Attivo".

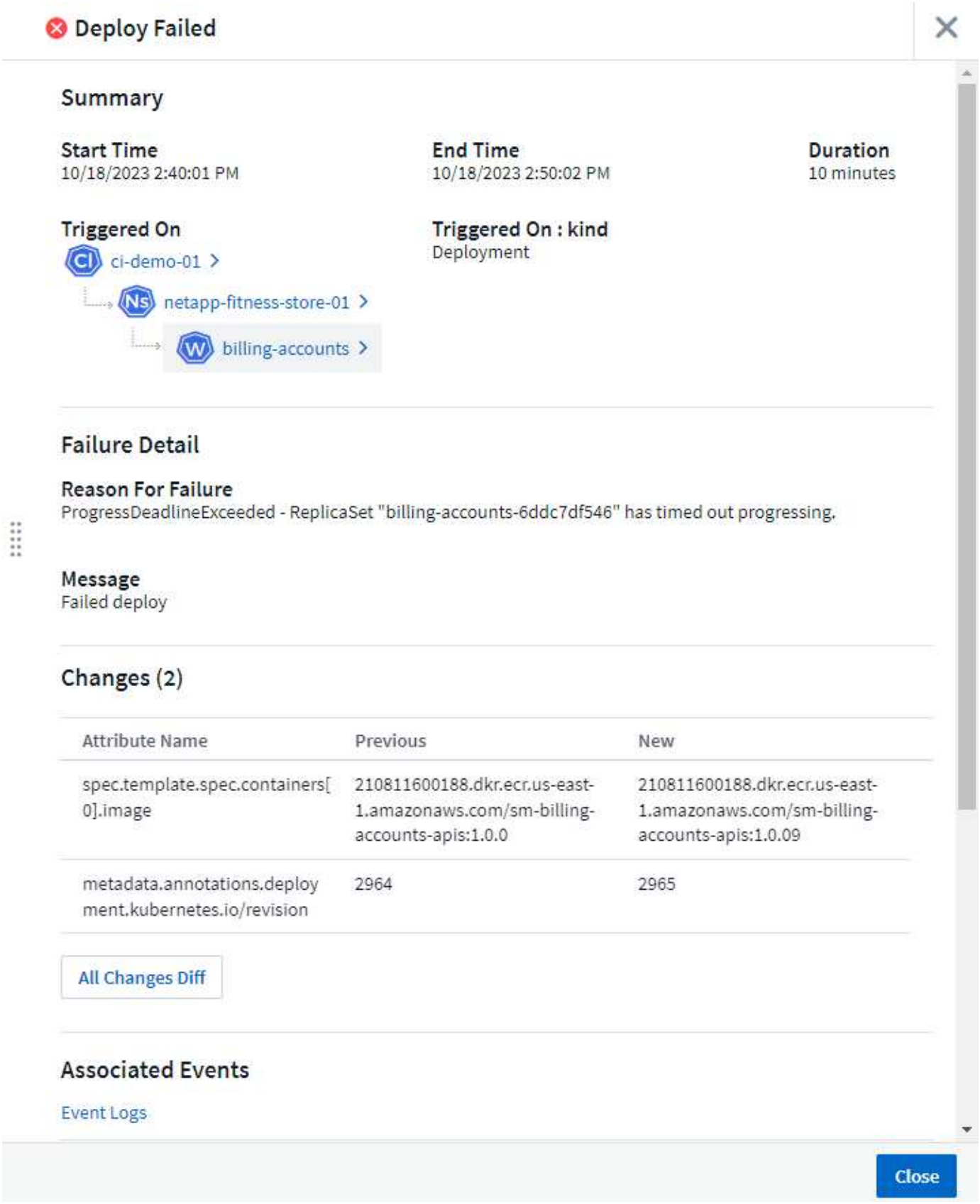
Qui viene mostrato anche lo stato di distribuzione. Anche in questo caso, l'impostazione predefinita è quella di mostrare il conteggio delle distribuzioni *Avviate*, *Completate* e *Non riuscite*. Per visualizzare solo le distribuzioni *Non riuscite*, imposta un filtro per "Stato" e seleziona "Non riuscito".

I successivi sono i 3 carichi di lavoro con il maggior numero di avvisi. Il numero in rosso accanto a ciascun carico di lavoro indica il numero di avvisi correlati a quel carico di lavoro. Fare clic sul collegamento del carico di lavoro per esplorare l'infrastruttura (Kubernetes Explorer), le dipendenze (mappa del carico di lavoro) o l'analisi dei registri (registri eventi).



Pannello di dettaglio

Selezionando una modifica nell'elenco si apre un pannello che descrive la modifica in modo più dettagliato. Ad esempio, selezionando una distribuzione non riuscita viene visualizzato un riepilogo della distribuzione, con orari di inizio e fine, durata e punto in cui è stata attivata la distribuzione, con collegamenti per esplorare tali risorse. Visualizza inoltre il motivo dell'errore, eventuali modifiche correlate ed eventuali eventi associati.



Anche selezionando un avviso si ottengono dettagli sull'avviso, tra cui il monitor che ha attivato l'avviso e un grafico che mostra una cronologia visiva dell'avviso.

Informazioni sul copyright

Informazioni sul marchio commerciale