Kubernetes

Data Infrastructure Insights

NetApp
January 17, 2025

Sommario

Kubernetes

Panoramica del cluster Kubernetes

Data Infrastructure Insights Kubernetes Explorer è un potente tool per visualizzare la salute e l'utilizzo generali dei cluster Kubernetes e ti consente di analizzare facilmente le aree di indagine.

Facendo clic su **Dashboards > Kubernetes Explorer** si apre la pagina Kubernetes Cluster. Questa pagina di panoramica contiene una tabella dei cluster Kubernetes sul tenant.

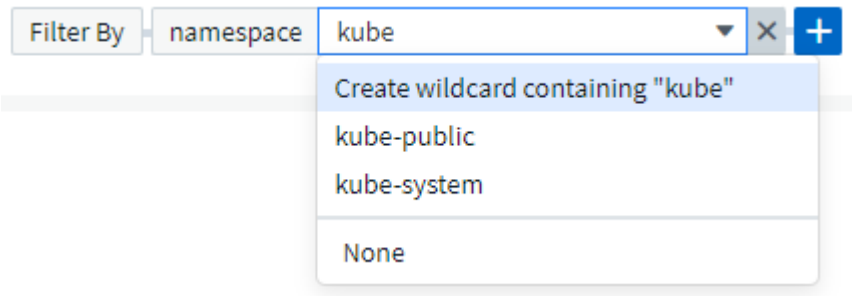
Elenco dei cluster

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

- Cluster **Nome**. Facendo clic sul nome di un cluster si apre il "**pagina dei dettagli**" relativo.
- Percentuali di **saturazione**. La saturazione complessiva è la più alta tra CPU, memoria o saturazione dello storage.
- Numero di **nodi** nel cluster. Facendo clic su questo numero si apre la pagina Node list (elenco nodi).
- Numero di **pod** nel cluster. Facendo clic su questo numero si apre la pagina dell'elenco dei pod.
- Numero di **namespace** nel cluster. Facendo clic su questo numero si apre la pagina dell'elenco dei namespace.
- Numero di **carichi di lavoro** nel cluster. Facendo clic su questo numero si apre la pagina elenco workload.

Rifinitura del filtro

Quando si esegue il filtraggio, quando si inizia a digitare viene visualizzata 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 con caratteri jolly. È inoltre possibile creare **espressioni** utilizzando NOR o E, oppure selezionare l'opzione "None" (Nessuno) per filtrare i valori nulli nel campo.



I filtri basati su caratteri jolly o espressioni (ad esempio, NOD, AND, "None", ecc.) vengono visualizzati in blu

scuro nel campo del filtro. Gli elementi selezionati direttamente dall'elenco vengono visualizzati in blu chiaro.

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

Si noti che i caratteri jolly e il filtraggio delle espressioni funzionano con testo o elenchi, ma non con valori numerici, date o booleani.

Prima di installare o aggiornare l'operatore di monitoraggio NetApp Kubernetes

Leggere queste informazioni prima di installare o aggiornare "Operatore di monitoring 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
OS Linux	Data Infrastructure Insights non supporta i nodi eseguiti con l'architettura Arm64. Monitoraggio della rete: Deve essere in esecuzione Linux kernel versione 4.18.0 o superiore. Il sistema operativo Photon non è supportato.
Etichette	Data Infrastructure Insights supporta il monitoring 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 ottenere i migliori risultati, aggiungere questi comandi al PERCORSO.

Componente	Requisito
Connettività	Kubectl cli è configurato per comunicare con il cluster K8s di destinazione e disporre di connettività Internet all'ambiente Data Infrastructure Insights. Se si è dietro un proxy durante l'installazione, seguire le istruzioni nella "Configurazione del supporto proxy"sezione Installazione dell'operatore. Per un controllo accurato e la refertazione dei dati, sincronizzare l'ora sul computer dell'agente utilizzando il protocollo NTP (Network Time Protocol) o SNTP (Simple Network Time Protocol).
Altro	Se si esegue OpenShift 4,6 o versione successiva, è necessario seguire "Istruzioni per OpenShift"oltre a garantire che i prerequisiti siano soddisfatti.
Token API	Se si sta ridistribuendo l'operatore (ad esempio se 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 si repository personalizzatoutilizza un proxy, un , o , OpenShiftleggere attentamente le sezioni seguenti.

Leggi anche su Permessi.

Configurazione del supporto proxy

Esistono due posizioni in cui è possibile utilizzare un proxy sul tenant per installare l'operatore di monitoraggio NetApp Kubernetes. Questi possono essere sistemi proxy identici o separati:

- Proxy necessario durante l'esecuzione del frammento di codice di installazione (utilizzando "curl") per connettere il sistema in cui il frammento viene eseguito all'ambiente Data Infrastructure Insights
- Proxy necessario dal 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 il monitor operativo NetApp Kubernetes è necessario innanzitutto assicurarsi che il proxy sia configurato in modo da consentire una buona comunicazione con l'ambiente informazioni sull'infrastruttura dati. Ad esempio, dai server/VM da cui si desidera installare l'operatore, è necessario essere in grado di accedere a Data Infrastructure Insights ed essere in grado di scaricare 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 la variabile *no_proxy environment*.

Per impostare le variabili, eseguire le seguenti operazioni sul sistema **prima** dell'installazione di NetApp Kubernetes Monitoring Operator:

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

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

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

Per il proxy utilizzato per il cluster Kubernetes e per comunicare con l'ambiente Data Infrastructure Insights, installare l'operatore di monitoraggio Kubernetes NetApp dopo aver letto tutte queste istruzioni.

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

Utilizzando un repository di docker personalizzato o privato

Per impostazione predefinita, l'operatore di monitoring NetApp Kubernetes estrarrà le immagini dei container dal repository di informazioni sull'infrastruttura dati. Se si utilizza un cluster Kubernetes come destinazione per il monitoraggio e tale cluster è configurato in modo da estrarre solo immagini container da un repository Docker personalizzato o privato o da un registro container, è necessario configurare l'accesso ai container richiesti dall'operatore di monitoraggio NetApp Kubernetes.

Eseguire il frammento Image Pull dalla sezione di installazione di NetApp Monitoring Operator. Questo comando consente di accedere al repository Data Infrastructure Insights, di estrarre tutte le dipendenze

dell'immagine per l'operatore e di disconnettersi dal repository Data Infrastructure Insights. Quando richiesto, inserire la password temporanea del repository fornita. Questo comando scarica tutte le immagini utilizzate dall'operatore, incluse le funzioni opzionali. Vedere di seguito per quali funzioni vengono utilizzate queste immagini.

Funzionalità principale dell'operatore e monitoraggio Kubernetes

- monitoraggio netapp
- kube-rbac-proxy
- kube-state-metrics
- telefono
- distroless-root-user

Registro eventi

- fluente
- kubernetes-event-exportent

Mappa e performance di rete

- ci-net-osservatore

Trasferire l'immagine del gestore nel repository del supporto privato/locale/aziendale in base alle policy aziendali. Verificare che i tag delle immagini e i percorsi delle directory per queste immagini nel repository siano coerenti con quelli nel repository Data Infrastructure Insights.

Modificare l'implementazione dell'operatore di monitoraggio in operator-deployment.yaml e modificare tutti i riferimenti alle immagini per utilizzare il 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>
```

Modificare la configurazione dell'agente in operator-config.yaml in modo che rifletta la nuova posizione del responsabile del docker. Crea un nuovo imagePullSecret per il tuo repository privato, per maggiori dettagli vedi *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 per OpenShift

Se si utilizza OpenShift 4.6 o versione successiva, è necessario modificare la configurazione dell'agente in *operator-config.yaml* per attivare l'impostazione *runPrivileged*:

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

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

Permessi

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

1. Modificare *operator-additional-permissions.yaml* prima dell'installazione o dopo l'installazione modificare la risorsa *ClusterRole/<namespace>-additional-permissions*
2. Creare una nuova regola per gli apartGroup e le risorse desiderati con i verbi ["Get", "Watch", "list"]. Vedere https://kubernetes.io/docs/reference/access-authn-authz/rbac/
3. Applicare le modifiche al cluster

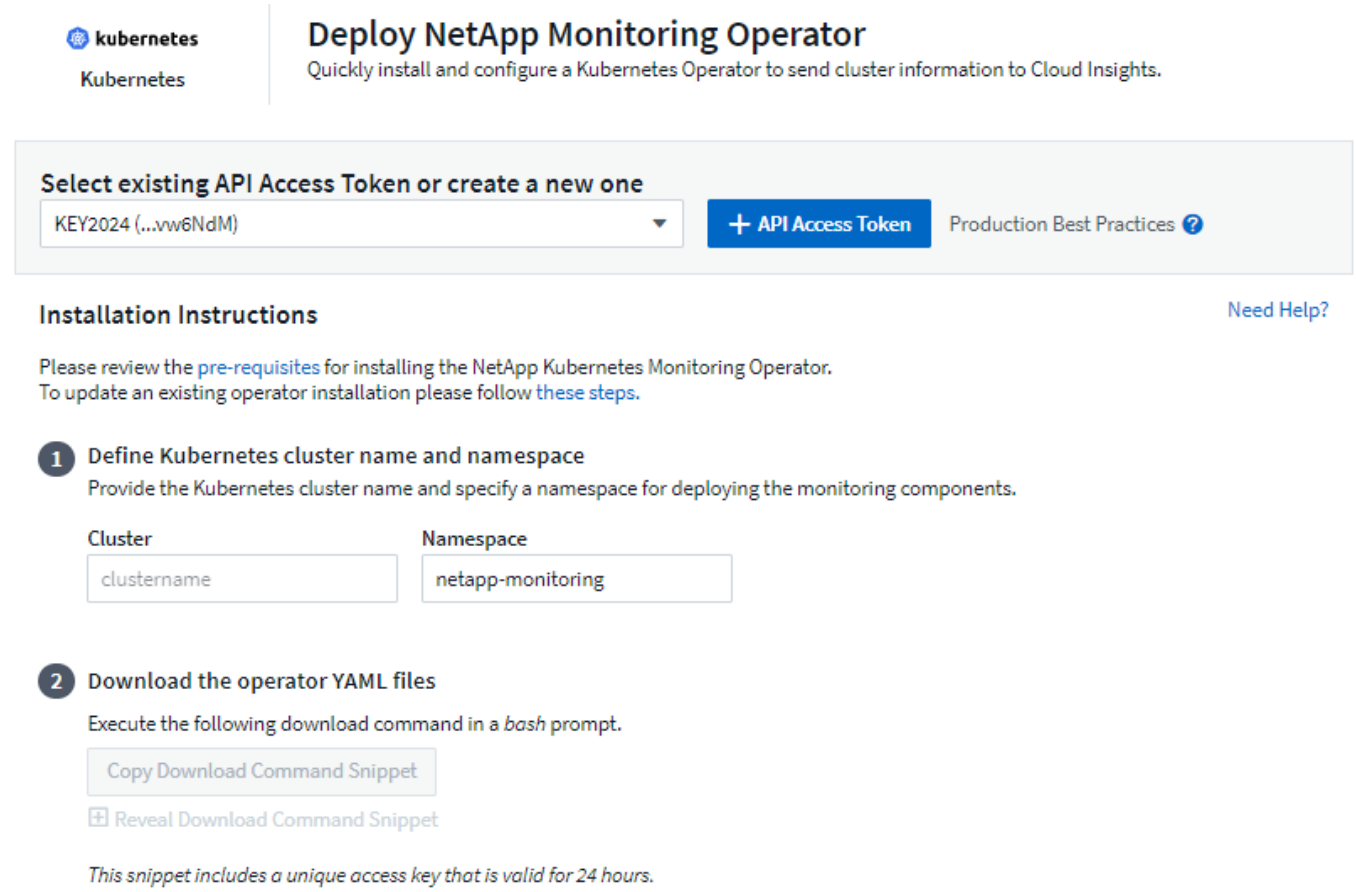
Installazione e configurazione dell'operatore di monitoraggio Kubernetes

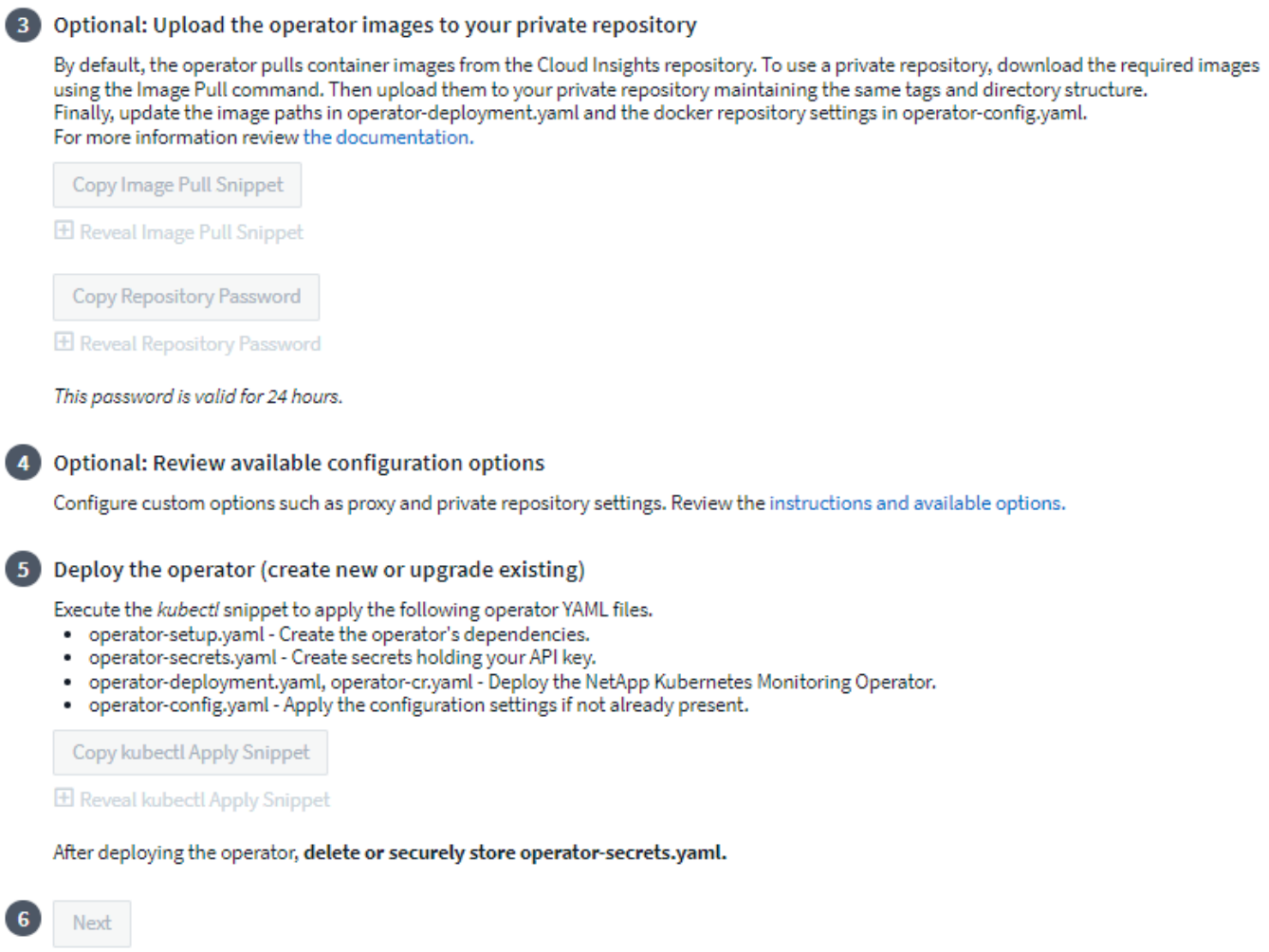
Data Infrastructure Insights offre la raccolta **Kubernetes Monitoring Operator** for Kubernetes. Navigare a **Kubernetes > Collector > +Kubernetes Collector** per implementare un nuovo operatore.

Prima di installare l'operatore di monitoraggio Kubernetes

Consultare la "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 si aggiornamento in corsoproviene da un operatore Kubernetes precedente, utilizzare lo stesso nome cluster e lo stesso namespace.
2. Una volta immessi, è possibile copiare il frammento Download Command negli Appunti.
3. Incollare il frammento in una finestra *bash* ed eseguirlo. I file di installazione dell'operatore verranno scaricati. Tenere presente che il frammento ha una chiave univoca ed è valido per 24 ore.
4. Se si dispone di un repository personalizzato o privato, copiare il frammento Image Pull opzionale, incollarlo in una shell *bash* ed eseguirlo. Una volta estratte le immagini, copiarle nel repository privato. Assicurarsi di mantenere gli stessi tag e la stessa struttura di cartelle. Aggiornare i percorsi in *operator-deployment.yaml* e le impostazioni del repository di docker in *operator-config.yaml*.
5. Se lo si desidera, esaminare le opzioni di configurazione disponibili, ad esempio le impostazioni del proxy o del repository privato. È possibile leggere ulteriori informazioni su "opzioni di configurazione".
6. Quando sei pronto, implementa l'operatore copiando il frammento kubectl apply, scaricandolo ed eseguendolo.
7. L'installazione procede automaticamente. Una volta completata l'operazione, fare clic sul pulsante *Avanti*.
8. Al termine dell'installazione, fare clic sul pulsante *Next*. Assicurarsi inoltre di eliminare o memorizzare in modo sicuro il file *operator-secrets.yaml*.

Se si utilizza un proxy, consultare informazioni su configurazione del proxy.

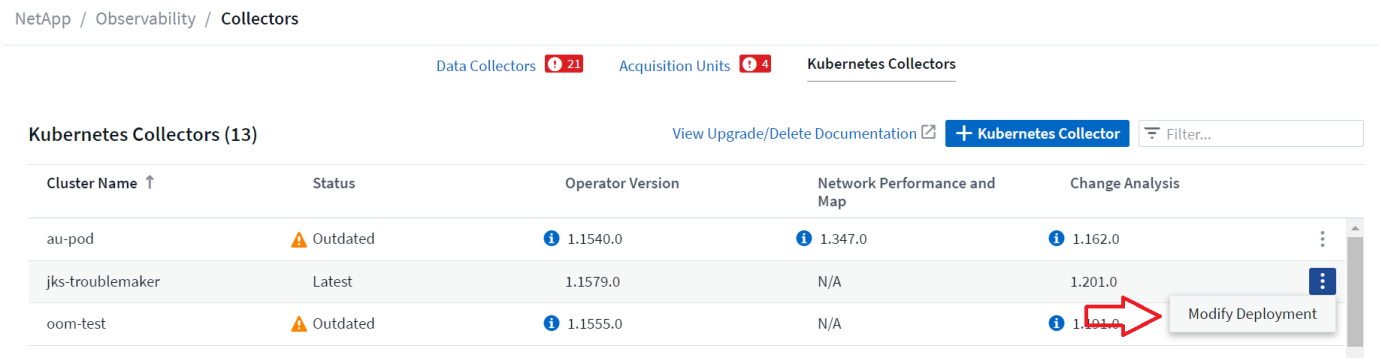
Se si dispone di un repository personalizzato, consultare informazioni su utilizzando un repository di docker personalizzato/privato.

Componenti di monitoring Kubernetes

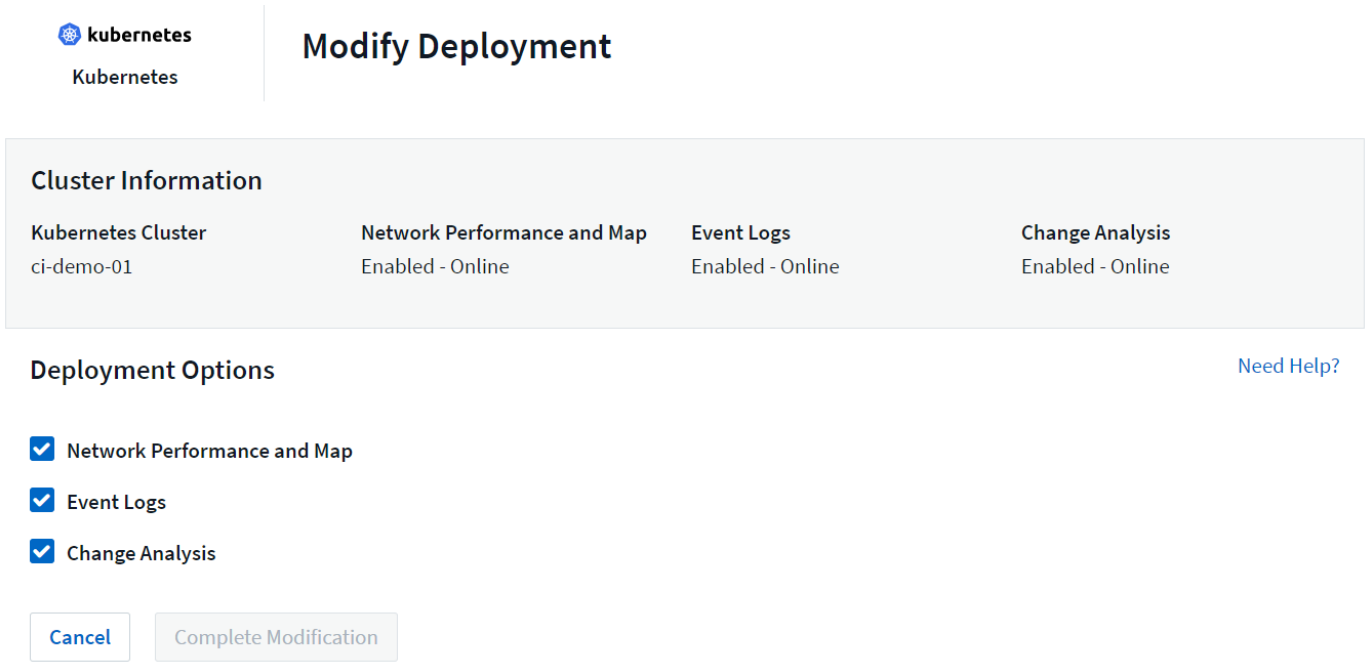
Data Infrastructure Insights Kubernetes Monitoring comprende quattro componenti di monitoring:

- Metriche cluster
- Mappa e prestazioni della rete (opzionale)
- Registri eventi (opzionali)
- Analisi delle modifiche (opzionale)

I componenti opzionali elencati in precedenza sono abilitati per impostazione predefinita per ogni collettore di Kubernetes; se si decide di non avere bisogno di un componente per un determinato collettore, è possibile disattivarlo accedendo a **Kubernetes > Collectors** e selezionando *Modify Deployment* dal menu "Three Dots" del collettore sulla destra dello schermo.



La schermata mostra lo stato corrente di ciascun componente e consente di disattivare o attivare i componenti per tale collettore, se necessario.



Aggiornamento alla versione più recente di Kubernetes Monitoring Operator

Determinare se esiste una configurazione Agentcon l'operatore esistente (se lo spazio dei nomi non è il *monitoraggio netapp* predefinito, sostituire lo spazio dei nomi appropriato):

```
 kubectl -n netapp-monitoring get agentconfiguration netapp-monitoring-
configuration
Se esiste una configurazione AgentConfiguration:
```

- Installare L'operatore più recente rispetto all'operatore esistente.
  - Assicurarsi di estrarre le immagini container più recentiutilizzare un repository personalizzato.

Se AgentConfiguration non esiste:

- Prendere nota del nome del cluster come riconosciuto da Data Infrastructure Insights (se il namespace non è quello predefinito di NetApp-monitoring, sostituire il namespace appropriato):

```
 kubectl -n netapp-monitoring get agent -o
jsonpath='{.items[0].spec.cluster-name}'
* Creare un backup dell'operatore esistente (se lo spazio dei nomi non è
il monitoraggio netapp predefinito, sostituire lo spazio dei nomi
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'operatore più recente.
```

  - Utilizzare lo stesso nome del cluster.
  - Dopo aver scaricato i file YAML dell'operatore più recenti, portare le personalizzazioni trovate in Agent_backup.yaml nell'operator-config.yaml scaricato prima di eseguire la distribuzione.
  - Assicurarsi di estrarre le immagini container più recentiutilizzare un repository personalizzato.

Arresto e avvio dell'operatore di monitoraggio Kubernetes

Per arrestare l'operatore di monitoraggio Kubernetes:

```
 kubectl -n netapp-monitoring scale deploy monitoring-operator
--replicas=0
Per avviare l'operatore di monitoraggio Kubernetes:
```

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

Disinstallazione in corso

Per rimuovere l'operatore di monitoraggio Kubernetes

Si noti che il namespace predefinito per Kubernetes Monitoring Operator è "netapp-monitoring". Se è stato impostato uno spazio dei nomi personalizzato, sostituire tale spazio dei nomi 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 spazio dei nomi dedicato, eliminare lo spazio dei nomi:

```
 kubectl delete ns <NAMESPACE>
Se il primo comando restituisce "Nessuna risorsa trovata", attenersi alle
istruzioni riportate di seguito 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 i messaggi 'oggetto non trovato'. Questi messaggi possono essere ignorati in modo sicuro.

```
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 del contesto di protezione:

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

A proposito di Kube-state-metrics

NetApp Kubernetes Monitoring Operator installa le proprie metriche di stato kube 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 di proxy, sull'utilizzo di un repository di docker personalizzato o privato o sull'utilizzo di OpenShift.

Opzioni di configurazione

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

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

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

```
 kubectl get crd agentconfigurations.monitoring.netapp.com
Se viene visualizzato il messaggio "Error from server (notfound)" (errore
dal server (non trovato)), l'operatore deve essere aggiornato prima di
poter utilizzare AgentConfiguration.
```

Configurazione del supporto proxy

Esistono due posizioni in cui è possibile utilizzare un proxy sul tenant per installare l'operatore di monitoraggio Kubernetes. Questi possono essere sistemi proxy identici o separati:

- Proxy necessario durante l'esecuzione del frammento di codice di installazione (utilizzando "curl") per connettere il sistema in cui il frammento viene eseguito all'ambiente Data Infrastructure Insights
- Proxy necessario dal cluster Kubernetes di destinazione per comunicare con l'ambiente Data Infrastructure Insights

Se si utilizza un proxy per uno o per entrambi, per installare il monitor operativo Kubernetes è necessario innanzitutto assicurarsi che il proxy sia configurato in modo da consentire una buona comunicazione con l'ambiente Data Infrastructure Insights. Se si dispone di un proxy e si può accedere a Data Infrastructure Insights dal server/VM da cui si desidera installare l'operatore, è probabile che il proxy sia configurato

correttamente.

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

Per impostare le variabili, eseguire i seguenti passaggi sul sistema **prima** di installare l'operatore di monitoraggio Kubernetes:

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

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

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

Per il proxy utilizzato per il cluster Kubernetes e per comunicare 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 l'operatore di monitoraggio Kubernetes.

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

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

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

Utilizzando un repository di docker personalizzato o privato

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

Eseguire il frammento Image Pull dalla sezione di installazione di NetApp Monitoring Operator. Questo comando consente di accedere al repository Data Infrastructure Insights, di estrarre tutte le dipendenze dell'immagine per l'operatore e di disconnettersi dal repository Data Infrastructure Insights. Quando richiesto, inserire la password temporanea del repository fornita. Questo comando scarica tutte le immagini utilizzate dall'operatore, incluse le funzioni opzionali. Vedere di seguito per quali funzioni vengono utilizzate queste immagini.

Funzionalità principale dell'operatore e monitoraggio Kubernetes

- monitoraggio netapp
- ci-kube-rbac-proxy
- ci-ksm
- ci-telegraf
- distroless-root-user

Registro eventi

- ci-fluent-bit

- ci-kukasub-esportatore-di-eventi

Mappa e performance di rete

- ci-net-osservatore

Trasferire l'immagine del gestore nel repository del supporto privato/locale/aziendale in base alle policy aziendali. Verificare che i tag delle immagini e i percorsi delle directory per queste immagini nel repository siano coerenti con quelli nel repository Data Infrastructure Insights.

Modificare l'implementazione dell'operatore di monitoraggio in operator-deployment.yaml e modificare tutti i riferimenti alle immagini per utilizzare il 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>
```

Modificare la configurazione dell'agente in operator-config.yaml in modo che rifletta la nuova posizione del responsabile del docker. Crea un nuovo imagePullSecret per il tuo repository privato, per maggiori dettagli vedi *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 per OpenShift

Se si utilizza OpenShift 4.6 o versione successiva, è necessario modificare la configurazione dell'agente in *operator-config.yaml* per attivare l'impostazione *runPrivileged*:

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

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

Tolerazioni e contamini

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 in modo da tollerare alcuni **segni** noti. Se sono stati configurati dei tag personalizzati sui nodi, impedendo così l'esecuzione dei pod su ogni nodo, è possibile creare una **tolleranza** per tali tag "In *AgentConfiguration*". Se sono stati applicati dei tipi di manutenzione personalizzati a tutti i nodi del cluster, è necessario aggiungere anche le tolleranze necessarie all'implementazione dell'operatore per consentire la pianificazione e l'esecuzione del pod operatore.

Scopri di più su Kubernetes "Contamini e pedaggi".

Tornare al **"Pagina Installazione dell'operatore di monitoraggio NetApp Kubernetes"**

Una nota sui segreti

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

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

Se si tratta di un aggiornamento, eliminare anche le risorse dal 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 attivata, modificare *AgentConfiguration* o *operator-config.yaml* per annullare il commento alla sezione di gestione delle modifiche e includere *kindsToIgnoreFromWatch: '"secrets"'* nella sezione di gestione delle modifiche. Notare la presenza e la posizione di 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 dell'immagine dell'operatore di monitoraggio Kubernetes

L'immagine per l'operatore e tutte le immagini correlate che implementa sono firmate da NetApp. Puoi verificare manualmente le immagini prima dell'installazione usando lo strumento csign, o configurare un controller di ammissione Kubernetes. Per ulteriori informazioni, vedere "Documentazione Kubernetes".

La chiave pubblica utilizzata per verificare le firme delle immagini è disponibile nel riquadro di installazione dell'operatore di monitoraggio in *Optional: Upload the operator images to your private repository > Image Signature Public Key*

Per verificare manualmente la firma di un'immagine, attenersi alla seguente procedura:

1. Copiare ed eseguire il frammento di estrazione dell'immagine
2. Quando richiesto, copiare e immettere la password dell'archivio
3. Memorizzare la chiave pubblica di firma dell'immagine (dii-image-signing.pub nell'esempio)
4. Verificare le immagini utilizzando il copiglia. Fare riferimento al seguente esempio di utilizzo dei cognomi

```
$ 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

Alcuni elementi da provare in caso di problemi durante la configurazione dell'operatore di monitoring Kubernetes:

Problema:	Prova:
Non viene visualizzato un collegamento ipertestuale/connessione tra il volume persistente Kubernetes e il dispositivo di storage back-end corrispondente. Il volume persistente Kubernetes viene configurato utilizzando il nome host del server di storage.	Seguire la procedura per disinstallare l'agente Telegraf esistente, quindi reinstallare l'agente Telegraf più recente. Devi utilizzare Telegraf versione 2,0 o successiva e lo storage del cluster Kubernetes deve essere monitorato attivamente da Data Infrastructure Insights.

Problema:	Prova:
Sto vedendo messaggi nei log che assomigliano a quanto segue: E0901 15:21 178 v1:39,962145 1 k8s k8s Reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: Impossibile elencare *352.MutatingWebhookConfigurazione: Il server non ha trovato la risorsa richiesta E0901 15:21:43,168161 1:v1 Reflector.go.me.get coordinazione del server.go.oblies	Questi messaggi possono verificarsi se si utilizza kube-state-metrics versione 2.0.0 o superiore con versioni di Kubernetes inferiori 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 loro implementazione di kube-state-metrics per disabilitare le seguenti Leases: _Mutatingwebcooki_argomenti_conservil possono usare le configurazioni_convalide_construzione_web: Resources=certificatesigningrequests,configmaps,cro ntowjobs,demonset,implementazioni,endpoint,horizont alpodautoscaler,ingassets,proxims,proxims,proxims,pr oxims,proxims,proxims,proxims,proxims,proxims,proxi ms,proxims,proxims,proxims,proxims,proxims,proxims ,proxims,proxims,proxims,proxims,proxims,proxims,pr oxims,proxims,proxims,proxims,proxims,proxims,proxi ms,proxims,proxims,proxims,proxims,proxims,proxims ,proxims,proxims,proxims,proxims,proxims, validatingwebhookconfigurations,volumeattachments"
Vedo messaggi di errore da Telegraf che assomigliano ai seguenti, ma Telegraf si avvia ed esegue: Oct 11 14:23:41:00 ip-172-31-39-47 systemd[1]: Avviato l'agente server plugin-driven per la generazione di rapporti in InfluxDB. Ottobre 11 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:172 1827 23" ott 2021 41Z:39-47 10 ip-31-23:41 telegraf[120]:-11 14"="errore di apertura. Ignorato. Open /etc/telegraf/.cache/snowflake/ocsp_Response_cache .json: No such file o directory\n" func="gosnowflake.(*defaultLogger).errorf" file="log.go:23" Oct 2021 41Z:10 ip-1827-31:39-47 traf[172]: 11 14-23:41:120! Avvio di Telegraf 1.19.3	Si tratta di un problema noto. Per "Questo articolo di GitHub"ulteriori dettagli, fare riferimento a. Finché Telegraf è in funzione, gli utenti possono ignorare questi messaggi di errore.
Su Kubernetes, i miei pod Telegraf riportano il seguente errore: "Errore nell'elaborazione delle informazioni sui mountstats: Impossibile aprire il file mountstats: /Hostfs/proc/1/mountstats, errore: Open /hostfs/proc/1/mountstats: Permesso negato"	Se SELinux è abilitato e abilitato, probabilmente impedisce ai pod Telegraf di accedere al file /proc/1/mountstats sul nodo Kubernetes. Per superare questa restrizione, modificare la configurazione dell'agente e attivare l'impostazione runPrivileged. Per ulteriori informazioni, fare riferimento alla "Istruzioni per OpenShift".

Problema:	Prova:
Su Kubernetes, il mio pod ReplicaSet Telegraf riporta 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.no	Il pod ReplicaSet di Telegraf è destinato all'esecuzione su un nodo designato come master o etcd. Se il pod ReplicaSet non è in esecuzione su uno di questi nodi, si otterranno questi errori. Verificare se i nodi master/etcd presentano delle contaminazioni. In tal caso, aggiungere le tolleranze necessarie a Telegraf ReplicaSet, telegraf-rs. Ad esempio, modificare il Replica Set… kubectl edita rs telegraf-rs …e aggiunga le tolleranze appropriate alla specifica. Quindi, riavviare il pod ReplicaSet.
Ho un ambiente PSP/PSA. Questo influisce 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 all'ultima versione di Kubernetes Monitoring Operator. Per eseguire l'aggiornamento all'operatore corrente con il supporto per PSP/PSA, procedere come segue: 1. Disinstallare l'operatore di monitoraggio precedente: kubectl delete agent-monitoring-NetApp -n NetApp-monitoring kubectl delete ns NetApp-monitoring kubectl delete crd agents.monitoring.NetApp.com kubectl delete clusterrole agent-manager-ruolo-proxy agent-metrics-reader kubectl delete clusterrolebinding agent-manager-rolebinding agent-rolebinding-proxy-ading-cluster-2. Installare la versione più recente dell'operatore di monitoraggio.
Ho riscontrato dei problemi durante la distribuzione dell'operatore e ho utilizzato PSP/PSA.	1. Modificare l'agente usando il seguente comando: Kubectl -n <name-space> edit Agent 2. Contrassegna "Security-policy-enabled" come "false". In questo modo si disattivano i criteri di protezione del pod e l'ammissione alla protezione del pod e si consente all'operatore di eseguire la distribuzione. Confermare con i seguenti comandi: Kubectl Get psp (dovrebbe mostrare la politica di sicurezza Pod rimossa) kubectl Get all -n <namespace>
grep -i psp (dovrebbe mostrare che non viene trovato nulla)	Errori "ImagePullBackoff" rilevati
Questi errori possono essere rilevati se si dispone di un repository di docker personalizzato o privato e non è ancora stato configurato l'operatore di monitoraggio Kubernetes in modo da riconoscerlo correttamente. Scopri di più informazioni sulla configurazione per repo personalizzato/privato.	Si verifica un problema con l'implementazione dell'operatore di monitoraggio e la documentazione corrente non mi aiuta a risolverlo.

Problema:	Prova:

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 (mappa del carico di lavoro) nello spazio dei nomi Operator si trovano in CrashLoopBackOff

Questi pod corrispondono al data collector Workload Map per l'osservabilità della rete. Prova: • Verifica 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","ambiente":"prod","level":"error","msg":"failed in validation. Motivo: La versione del kernel 3.10.0 è inferiore alla versione minima del kernel di 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 usando il comando "uname -r" e assicurarsi che siano >= 4.18.0	I pod vengono eseguiti in Operator namespace (predefinito: Monitoring netapp), ma non vengono visualizzati dati nell'interfaccia utente per la mappa dei carichi di lavoro o le metriche Kubernetes nelle query
Controllare l'impostazione dell'ora sui nodi del cluster K8S. Per un controllo accurato e la creazione di report dei dati, si consiglia di sincronizzare l'ora sul computer dell'agente utilizzando il protocollo NTP (Network Time Protocol) o SNTP (Simple Network Time Protocol).	Alcuni dei pod net-observer nello spazio dei nomi Operator sono in stato Pending
NET-osservatore è un DemonSet che esegue un pod in ogni nodo del cluster k8s. • Prendere nota del pod in stato Pending (in sospeso) e verificare se si verifica un problema di risorse per la CPU o la memoria. Assicurarsi che la memoria e la CPU richieste siano disponibili nel nodo.	Nei miei registri, subito dopo l'installazione dell'operatore di monitoraggio di Kubernetes, viene visualizzato quanto segue: [inputs.prometheus] errore nel plugin: Errore durante la richiesta HTTP a http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: Get http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: Dial tcp: Lookup kube-state-metrics.<namespace>.svc.cluster.local: No such host

Problema:	Prova:
Questo messaggio viene visualizzato in genere solo quando viene installato un nuovo operatore e il pod *telegraf-rs* è attivo prima che il pod *ksm* sia attivo. Questi messaggi dovrebbero interrompersi una volta che tutti i pod sono in esecuzione.	Non vedo alcuna metrica raccolta per Kubernetes Cronjobs che esiste nel mio cluster.
Verificare la versione di Kubernetes (ad es. `kubectl version`). Se è v1.20.x o inferiore, si tratta di un limite previsto. La release kube-state-metrics implementata con Kubernetes Monitoring Operator supporta solo v1.cronjob. Con Kubernetes 1.20.x e versioni precedenti, la risorsa cronjob è v1beta.cronjob. Di conseguenza, le metriche dello stato del kube non riescono a trovare la risorsa di crono-job.	Dopo aver installato l'operatore, i pod telegraf-ds entrano in CrashLoopBackOff e i registri del pod indicano "su: Authentication failure" (su: Errore di autenticazione).
Modificare la sezione telegraf in *AgentConfiguration* e impostare *dockerMetricCollectionEnabled* su false. Per ulteriori dettagli, fare riferimento alla "opzioni di configurazione". telegraf: … - Name: docker run-mode: - DaemonSet sostituzioni: - Chiave: DOCKER_UNIX_SOCK_PLACEHOLDER valore: unix://run/docker.sock … …	Vedo messaggi di errore ricorrenti simili ai seguenti nei miei registri Telegraf: E! [Agent] errore di scrittura in outputs.http: Post "https://<tenant_url>/REST/v1/Lake/ingerment/influenzxdb": Scadenza contesto superata (client. Timeout durante l'attesa delle intestazioni)
Modificare la sezione telegraf in *AgentConfiguration* e aumentare *outputTimeout* a 10s. Per ulteriori dettagli, fare riferimento alla "opzioni di configurazione".	Mancano i dati *involvedobject* per alcuni registri eventi.
Assicurarsi di aver seguito i passaggi descritti nella "Permessi"sezione precedente.	Perché vedo due pod operatore di monitoring 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 ridefinito l'operatore per servire meglio i nostri utenti; affinché tali modifiche vengano completamente adottate, è necessario rimuovere il vecchio operatore e installare il nuovo.	I miei eventi kuowdi hanno interrotto inaspettatamente la segnalazione a Data Infrastructure Insights.
Recuperare il nome del pod dell'esportatore di eventi: `` `kubectl -n netapp-monitoring get pods ``	grep event-exporter

Problema:	Prova:
awk '{print $1}'	sed 's/event-exporter./event-exporter/'` Deve essere "netapp-ci-event-exportant" o "event-exportant". Quindi, modificare l'agente di monitoraggio `kubectl -n netapp-monitoring edit agent` e impostare il valore per LOG_FILE in modo che rifletta il nome del pod dell'esportatore di eventi appropriato trovato nel passaggio precedente. In particolare, LOG_FILE deve essere impostato su "/var/log/containers/netapp-ci-event-exportant.log" o "/var/log/containers/event-exportant*.log" …. fluent-bit: … - name: event-exporter-ci substitutions: - key: LOG_FILE values: - /var/log/containers/netapp-ci-event-exporter*.log … …. In alternativa, si può anche disinstallazione e reinstallare l'agente.
Sto vedendo i pod implementati dal crash dell'operatore di monitoring Kubernetes a causa di risorse insufficienti.	Fare riferimento a Kubernetes Monitoring Operator "opzioni di configurazione" per aumentare i limiti di CPU e/o memoria secondo necessità.
Un'immagine mancante o una configurazione non valida ha causato il mancato avvio o la mancata preparazione dei pod di metriche a stato di netapp-ci-kube. Ora StatefulSet è bloccato e le modifiche della configurazione non vengono applicate ai pod di metriche stato netapp-ci-kube.	StatefulSet è in uno "rotto" stato. Dopo aver risolto eventuali problemi di configurazione, bounce i pod di metrica stato netapp-ci-kube.
I pod con metriche a stato di netapp-ci-kube non si avviano dopo l'aggiornamento di un operatore Kubernetes, lanciando ErrImagePull (non riuscendo a estrarre l'immagine).	Provare a reimpostare i pod manualmente.
I messaggi "evento scartato come vecchio allora maxEventAgeSeconds" vengono osservati per il mio cluster Kubernetes in Log Analysis.	Modificare l'operatore *agentconfiguration* e aumentare il *event-exportant-maxEventAgeSeconds* (cioè a 60s), il *event-exportant-kubeQPS* (cioè a 100) e il *event-exportant-kubeBurst* (cioè a 500). Per ulteriori informazioni su queste opzioni di configurazione, consultare la "opzioni di configurazione" pagina.

Problema:	Prova:
Telegraf avverte di, o si blocca a causa di, memoria bloccabile insufficiente.	Provare ad aumentare il limite di memoria bloccabile per Telegraf nel sistema operativo/nodo sottostante. Se l'aumento del limite non è un'opzione, modificare la configurazione dell'agente NKMO e impostare *non protetto* su *true*. In questo modo, Telegraf non tenterà di riservare pagine di memoria bloccate. Sebbene ciò possa rappresentare un rischio per la sicurezza poiché i segreti decrittografati potrebbero essere scambiati sul disco, consente l'esecuzione in ambienti in cui non è possibile riservare la memoria bloccata. Per ulteriori informazioni sulle opzioni di configurazione *non protetto*, fare riferimento alla "opzioni di configurazione"pagina.
Vedo messaggi di avviso da Telegraf simili a quanto segue: *W! [Inputs.diskio] Impossibile raccogliere il nome del disco per "vdc": Errore di lettura /dev/vdc: Nessun file o directory*	Per l'operatore di monitoring Kubernetes, questi messaggi di avviso sono benigni e possono essere ignorati in modo sicuro. In alternativa, modificare la sezione telegraf in AgentConfiguration e impostare *runDsPrivileged* su true. Per ulteriori informazioni, fare riferimento alla "opzioni di configurazione dell'operatore".

Problema:	Prova:

Il mio Fluent-bit pod non funziona con i seguenti errori: [2024/10/16 14 16:16 23:23] [errore] [/src/fluent-bit/plugins/in_tail/tail_fs_inotify.c:tail,0 errno=2024] troppi file aperti [10/16 14/10/16 14:16:23] [errore] Inizializzazione input non riuscita [24/2024:360] [errore] [motore] Inizializzazione non riuscita	Prova a modificare le impostazioni di *fsnotify* nel 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>`
	
	Riavviare Fluent-bit.
	
	Nota: Per rendere queste impostazioni persistenti durante i riavvii dei nodi, 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>`

I pod DS di telegraf riportano errori relativi al mancato invio di richieste HTTP da parte del plugin di input kuPdi a causa dell’impossibilità di convalidare il certificato TLS. Ad esempio: E! [Inputs.kuPQ] errore nel plugin: Errore durante la richiesta HTTP di "<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 alcuna SAN IP	Questo si verifica se il kubelet utilizza certificati autofirmati e/o il certificato specificato non include il <kubelet_IP> nell'elenco dei certificati *Subject alternative Name*. Per risolvere questo problema, l'utente può modificare il "configurazione dell'agente"e impostare *telegraf:insecureK8sSkipVerify* su *true*. Questo configurerà il plugin di input telegraf 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 sono disponibili nella "Supporto"pagina o nella "Matrice di supporto Data Collector".

Opzioni di configurazione dell'operatore di monitoraggio Kubernetes

La "Operatore di monitoring Kubernetes" configurazione può essere personalizzata.

La tabella seguente elenca le opzioni possibili per il file *AgentConfiguration*:

Componente	Opzione	Descrizione
agente		Opzioni di configurazione comuni a tutti i componenti che l'operatore può installare. Queste opzioni possono essere considerate "globali".
	DockerRepo	Un dockerRepo override per estrarre le immagini dai repos privati dei docker dei clienti rispetto a Data Infrastructure Insights docker repo. Di default è incluso il repo del docker di Data Infrastructure Insights
	DockerImagePullSecret	Facoltativo: Un segreto per i clienti privati
	Nome cluster	Campo di testo libero che identifica in modo univoco un cluster in tutti i cluster dei clienti. Questa impostazione deve essere univoca in un tenant Data Infrastructure Insights. Il valore predefinito è quello che il cliente inserisce nell'interfaccia utente per il campo "Cluster Name" (Nome cluster)
	Proxy Format: Proxy: Server: Porta: Nome utente: Password: NoProxy: IsTelegrafProxyEnabled: IsAuProxyEnabled: IsFluentbitProxyEnabled: IsCollectorProxyEnabled:	Opzionale per impostare proxy. Si tratta in genere del proxy aziendale del cliente.
telefono		Opzioni di configurazione che consentono di personalizzare l'installazione di telegraf dell'operatore
	CollectionInterval	Intervallo di raccolta delle metriche, in secondi (max=60s)
	DsCpuLimit	Limite CPU per telegraf ds
	DsMemLimit	Limite di memoria per telegraf ds
	DsCpuRequest	Richiesta CPU per telegraf ds
	DsMemRequest	Richiesta di memoria per telegraf ds
	RsCpuLimit	Limite CPU per telegraf rs
	RsMemLimit	Limite di memoria per telegraf rs
	RsCpuRequest	Richiesta CPU per telegraf rs

Componente	Opzione	Descrizione
	RsMemRequest	Richiesta di memoria per telegraf rs
	RunPriveged	Eseguire il contenitore *telegraf-mountstats-polliner* di telegraf DaemonSet in modalità privilegiata. Impostare questo valore su true se SELinux è abilitato sui nodi Kubernetes.
	RunDsPrivileged	Impostare runDsPrivileged su true per eseguire il contenitore telegraf DaemonSet in modalità privilegiata.
	Batch Size (dimensione batch)	Vedere "Documentazione sulla configurazione di Telegraf"
	BufferLimit	Vedere "Documentazione sulla configurazione di Telegraf"
	RoundInterval	Vedere "Documentazione sulla configurazione di Telegraf"
	CollectionJitter	Vedere "Documentazione sulla configurazione di Telegraf"
	precisione	Vedere "Documentazione sulla configurazione di Telegraf"
	FlushInterval	Vedere "Documentazione sulla configurazione di Telegraf"
	FlushJitter	Vedere "Documentazione sulla configurazione di Telegraf"
	OutputTimeout	Vedere "Documentazione sulla configurazione di Telegraf"
	DsTolerazioni	teletegraf-ds tollerazioni aggiuntive.
	RsTolerazioni	tollerazioni aggiuntive di telegraf-rs.
	SkipProcessorsAfterAggregators	Vedere "Documentazione sulla configurazione di Telegraf"
	non protetto	ee questo "Problema noto di Telegraf". L'impostazione *non protetta* indicherà all'operatore di monitoraggio Kubernetes di eseguire Telegraf con il `--unprotected` flag.
	insecureK8sSkipVerify	Se telegraf non è in grado di verificare il certificato a causa della mancanza di SAN IP, provare ad attivare il salto di verifica
kube-state-metrics		Opzioni di configurazione che possono personalizzare l'installazione delle metriche di stato kube dell'operatore
	CpuLimit	Limite di CPU per l'implementazione delle metriche di stato kube
	MemLimit	Limite MEM per l'implementazione delle metriche dello stato del kube

Componente	Opzione	Descrizione
	CpuRequest	Richiesta di CPU per l'implementazione delle metriche di stato del kube
	MemRequest	Richiesta MEM per l'implementazione delle metriche di stato del kube
	risorse	un elenco separato da virgole di risorse da acquisire. esempio: cronjobs,daemonsets,deployments,ingresses,jobs,namespaces,node,persistentvolumeclaims, persistentvolumes,pods,replicasets,resourcequotas,servizi,statefulsets
	tollerazioni	tolleranze aggiuntive delle metriche dello stato del kube.
	etichette	un elenco separato da virgole di risorse che le metriche di stato kube devono acquisire + esempio: cronjobs=**[],demonsets=[]**,deployments=**[],ingresses=[]**,jobs=**[],namespaces=[]**,nodes=**[], persistentvolumeclaims=[]**[][][][][+]
registri		Opzioni di configurazione che consentono di personalizzare la raccolta e l'installazione dei log dell'operatore
	ReadFromHead	vero/falso, dovrebbe leggere fluentemente il log dalla testa
	timeout	timeout, in sec.
	DnsMode	TCP/UDP, modalità per DNS
	tolleranza ai bit fluente	tolleranza aggiuntiva ai bit fluenti.
	tolleranza-evento-esportatore	tolleranza aggiuntiva per gli esportatori di eventi.
	Event-exportant-maxEventAgeSeconds	età massima dell'evento dell'esportatore. Vedere https://github.com/jkroepke/resmoio-kubernetes-event-exporter
mappa del carico di lavoro		Opzioni di configurazione che possono personalizzare la raccolta della mappa del carico di lavoro e l'installazione dell'operatore.
	CpuLimit	Limite CPU per i server di osservazione della rete
	MemLimit	limite mem per gli osservatori netti
	CpuRequest	Richiesta CPU per net osservatore ds
	MemRequest	richiesta mem per net osservatore ds
	MetricAggregationInterval	intervallo di aggregazione metrico, in secondi
	BpfPollInterval	Intervallo di polling BPF, in secondi
	EnableDNSLookup	Vero/falso, attiva ricerca DNS

Componente	Opzione	Descrizione
	l4-tollerazioni	tolleranza aggiuntiva net-observer-l4-ds.
	RunPriveged	Vero/falso - impostare runPrivileged su true se SELinux è abilitato sui tuoi nodi Kubernetes.
change-management		Opzioni di configurazione per l'analisi e la gestione delle modifiche di Kubernetes
	CpuLimit	Limite CPU per change-observer-watch-rs
	MemLimit	Limite MEM per change-observer-watch-rs
	CpuRequest	Richiesta CPU per change-observer-watch-rs
	MemRequest	richiesta mem per change-observer-watch-rs
	FailureDeclarationIntervalMins	Intervallo in minuti dopo il quale un'implementazione non riuscita di un carico di lavoro viene contrassegnata come non riuscita
	DeployAggrIntervalSeconds	Frequenza con cui vengono inviati gli eventi di distribuzione del carico di lavoro in corso
	NonWorkloadAggrIntervalSeconds	Frequenza di combinazione e invio delle implementazioni non a carico di lavoro
	TermsToRedact	Un insieme di espressioni regolari utilizzate nei nomi env e nelle mappe di dati il cui valore sarà redacted termini di esempio:"pwd", "password", "token", "apikey", "api-key", "jwt"
	AdditionalKindsToWatch	Un elenco separato da virgole di tipi aggiuntivi da guardare dal set di tipi predefinito guardato dal raccoglitore
	KindsToIgnoreFromWatch	Un elenco di tipi separati da virgole da ignorare dall'insieme predefinito di tipi controllati dal raccoglitore
	LogRecordAggrIntervalSeconds	Frequenza con cui i record di registro vengono inviati al ci dal raccoglitore
	tolleranza di controllo	modifica-osservatore-guarda-ds tolleranze aggiuntive. Solo formato abbreviato a riga singola. Esempio: '{key: taint1, operator: Exists, Effect: NoSchedule},{key: taint2, operator: Exists, Effect: NoExecute}'

Esempio di file AgentConfiguration

Di seguito è riportato un file *AgentConfiguration* di esempio.

```
apiVersion: monitoring.netapp.com/v1alpha1
kind: AgentConfiguration
metadata:
  name: netapp-ci-monitoring-configuration
```

```
  namespace: "netapp-monitoring"
  labels:
    installed-by: nkmo-netapp-monitoring

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

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

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

    # # Allow the operator to automatically rotate its ApiKey before
expiration.
```

```
    # tokenRotationEnabled: 'true'
    # # Number of days before expiration that the ApiKey should be
rotated. This must be less than the total ApiKey duration.
    # tokenRotationThresholdDays: '30'

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

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

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

```
    # dsCpuRequest: '100m'
    # dsMemRequest: '500Mi'

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

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

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


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

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

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

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

    # # Collect NFS IO metrics.
```

```
    # dsNfsIOEnabled: 'true'

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

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

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

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

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

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

    # # Comma-separated list of metrics to enable.
    # # See metric-allowlist in https://github.com/kubernetes/kube-state-
metrics/blob/main/docs/cli-arguments.md
    # metrics:
'kube_cronjob_created,kube_cronjob_status_active,kube_cronjob_labels,kube_
daemonset_created,kube_daemonset_status_current_number_scheduled,kube_daem
onset_status_desired_number_scheduled,kube_daemonset_status_number_availab
le,kube_daemonset_status_number_misscheduled,kube_daemonset_status_number_
ready,kube_daemonset_status_number_unavailable,kube_daemonset_status_obser
ved_generation,kube_daemonset_status_updated_number_scheduled,kube_daemons
```

```
et_metadata_generation,kube_daemonset_labels,kube_deployment_status_replic
as,kube_deployment_status_replicas_available,kube_deployment_status_replic
as_unavailable,kube_deployment_status_replicas_updated,kube_deployment_sta
tus_observed_generation,kube_deployment_spec_replicas,kube_deployment_spec
_paused,kube_deployment_spec_strategy_rollingupdate_max_unavailable,kube_d
eployment_spec_strategy_rollingupdate_max_surge,kube_deployment_metadata_g
eneration,kube_deployment_labels,kube_deployment_created,kube_job_created,
kube_job_owner,kube_job_status_active,kube_job_status_succeeded,kube_job_s
tatus_failed,kube_job_labels,kube_job_status_start_time,kube_job_status_co
mpletion_time,kube_namespace_created,kube_namespace_labels,kube_namespace_
status_phase,kube_node_info,kube_node_labels,kube_node_role,kube_node_spec
_unschedulable,kube_node_created,kube_persistentvolume_capacity_bytes,kube
_persistentvolume_status_phase,kube_persistentvolume_labels,kube_persisten
tvolume_info,kube_persistentvolume_claim_ref,kube_persistentvolumeclaim_ac
cess_mode,kube_persistentvolumeclaim_info,kube_persistentvolumeclaim_label
s,kube_persistentvolumeclaim_resource_requests_storage_bytes,kube_persiste
ntvolumeclaim_status_phase,kube_pod_info,kube_pod_start_time,kube_pod_comp
letion_time,kube_pod_owner,kube_pod_labels,kube_pod_status_phase,kube_pod_
status_ready,kube_pod_status_scheduled,kube_pod_container_info,kube_pod_co
ntainer_status_waiting,kube_pod_container_status_waiting_reason,kube_pod_c
ontainer_status_running,kube_pod_container_state_started,kube_pod_containe
r_status_terminated,kube_pod_container_status_terminated_reason,kube_pod_c
ontainer_status_last_terminated_reason,kube_pod_container_status_ready,kub
e_pod_container_status_restarts_total,kube_pod_overhead_cpu_cores,kube_pod
_overhead_memory_bytes,kube_pod_created,kube_pod_deletion_timestamp,kube_p
od_init_container_info,kube_pod_init_container_status_waiting,kube_pod_ini
t_container_status_waiting_reason,kube_pod_init_container_status_running,k
ube_pod_init_container_status_terminated,kube_pod_init_container_status_te
rminated_reason,kube_pod_init_container_status_last_terminated_reason,kube
_pod_init_container_status_ready,kube_pod_init_container_status_restarts_t
otal,kube_pod_status_scheduled_time,kube_pod_status_unschedulable,kube_pod
_spec_volumes_persistentvolumeclaims_readonly,kube_pod_container_resource_
requests_cpu_cores,kube_pod_container_resource_requests_memory_bytes,kube_
pod_container_resource_requests_storage_bytes,kube_pod_container_resource_
requests_ephemeral_storage_bytes,kube_pod_container_resource_limits_cpu_co
res,kube_pod_container_resource_limits_memory_bytes,kube_pod_container_res
ource_limits_storage_bytes,kube_pod_container_resource_limits_ephemeral_st
orage_bytes,kube_pod_init_container_resource_limits_cpu_cores,kube_pod_ini
t_container_resource_limits_memory_bytes,kube_pod_init_container_resource_
limits_storage_bytes,kube_pod_init_container_resource_limits_ephemeral_sto
rage_bytes,kube_pod_init_container_resource_requests_cpu_cores,kube_pod_in
it_container_resource_requests_memory_bytes,kube_pod_init_container_resour
ce_requests_storage_bytes,kube_pod_init_container_resource_requests_epheme
ral_storage_bytes,kube_replicaset_status_replicas,kube_replicaset_status_r
eady_replicas,kube_replicaset_status_observed_generation,kube_replicaset_s
pec_replicas,kube_replicaset_metadata_generation,kube_replicaset_labels,ku
```

```
be_replicaset_created,kube_replicaset_owner,kube_resourcequota,kube_resour
cequota_created,kube_service_info,kube_service_labels,kube_service_created
,kube_service_spec_type,kube_statefulset_status_replicas,kube_statefulset_
status_replicas_current,kube_statefulset_status_replicas_ready,kube_statef
ulset_status_replicas_updated,kube_statefulset_status_observed_generation,
kube_statefulset_replicas,kube_statefulset_metadata_generation,kube_statef
ulset_created,kube_statefulset_labels,kube_statefulset_status_current_revi
sion,kube_statefulset_status_update_revision,kube_node_status_capacity,kub
e_node_status_allocatable,kube_node_status_condition,kube_pod_container_re
source_requests,kube_pod_container_resource_limits,kube_pod_init_container
_resource_limits,kube_pod_init_container_resource_requests'

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

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

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

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

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

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

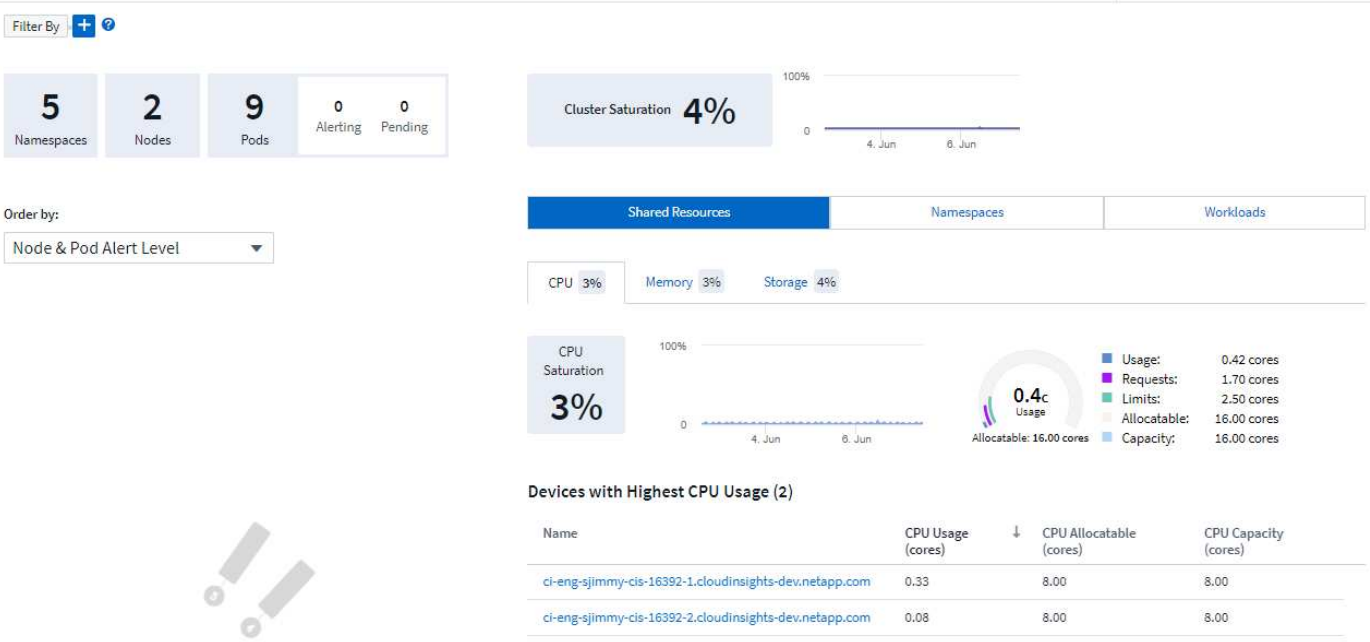
```
    # # DNS resolver that Fluent Bit should use: "LEGACY" or "ASYNC"
    # fluentBitDNSResolver: "LEGACY"

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

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

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

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

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

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

```
    # memLimit: '500Mi'
    # cpuRequest: '100m'
    # memRequest: '500Mi'

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

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

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

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

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

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

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

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

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

    # # A set of regular expressions used in env names and data maps whose
```

```
value will be redacted
    # termsToRedact: '"pwd", "password", "token", "apikey", "api-key",
"api_key", "jwt", "accesskey", "access_key", "access-key", "ca-file",
"key-file", "cert", "cafile", "keyfile", "tls", "crt", "salt",
".dockerconfigjson", "auth", "secret"'

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

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

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

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

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

Pagina dei dettagli del cluster Kubernetes

La pagina dei dettagli del cluster Kubernetes visualizza una panoramica dettagliata del cluster Kubernetes.



Namespace, Node e Pod Counts

I conteggi nella parte superiore della pagina mostrano il numero totale di spazi dei nomi, nodi e pod nel cluster, nonché il numero di pop-of che sono attualmente in stato di avviso e in sospeso.

Risorse condivise e saturazione

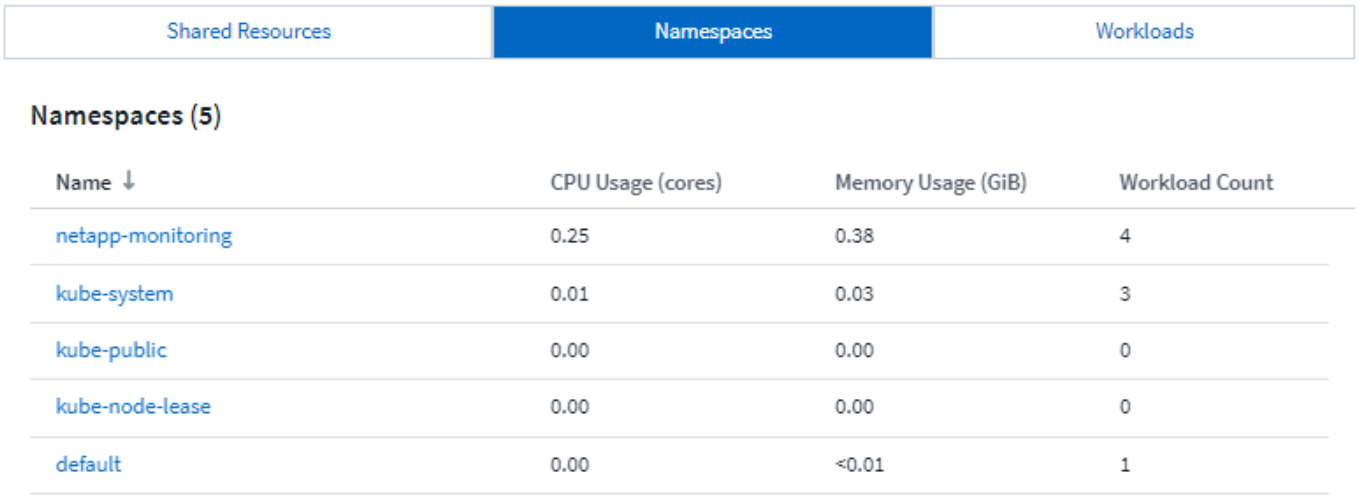
Nella parte superiore destra della pagina dei dettagli si trova la saturazione del cluster come percentuale corrente e un grafico che mostra la tendenza recente nel tempo. La saturazione del cluster è la più alta tra CPU, memoria o saturazione dello storage in ogni punto del tempo.

Di seguito, la pagina mostra per impostazione predefinita l'utilizzo di **risorse condivise**, con schede per CPU, memoria e storage. Ogni scheda mostra la percentuale di saturazione e l'andamento nel tempo, con ulteriori dettagli sull'utilizzo. Per lo storage, il valore mostrato è maggiore tra il backend e la saturazione del file system, che vengono calcolati in modo indipendente.

I dispositivi con il massimo utilizzo sono mostrati in una tabella nella parte inferiore. Fare clic su un collegamento qualsiasi per esplorare questi dispositivi.

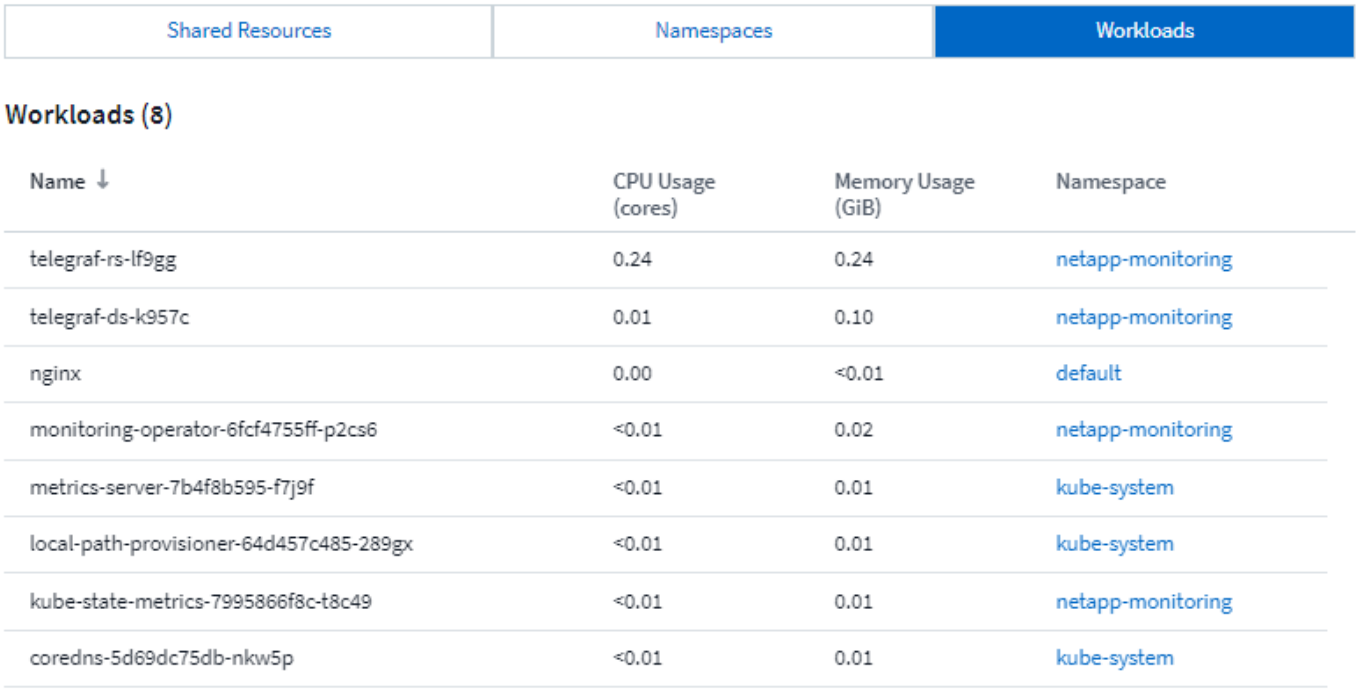
Spazi dei nomi

La scheda Namespaces visualizza un elenco di tutti gli spazi dei nomi nell'ambiente Kubernetes, mostrando l'utilizzo di CPU e memoria e il numero di carichi di lavoro in ogni spazio dei nomi. Fare clic sui link Name (Nome) per esplorare ciascun namespace.



Carichi di lavoro

Allo stesso modo, la scheda workload visualizza un elenco dei carichi di lavoro in ogni namespace, mostrando nuovamente l'utilizzo di CPU e memoria. Facendo clic sullo spazio dei nomi, è possibile accedere a ciascuno di essi.



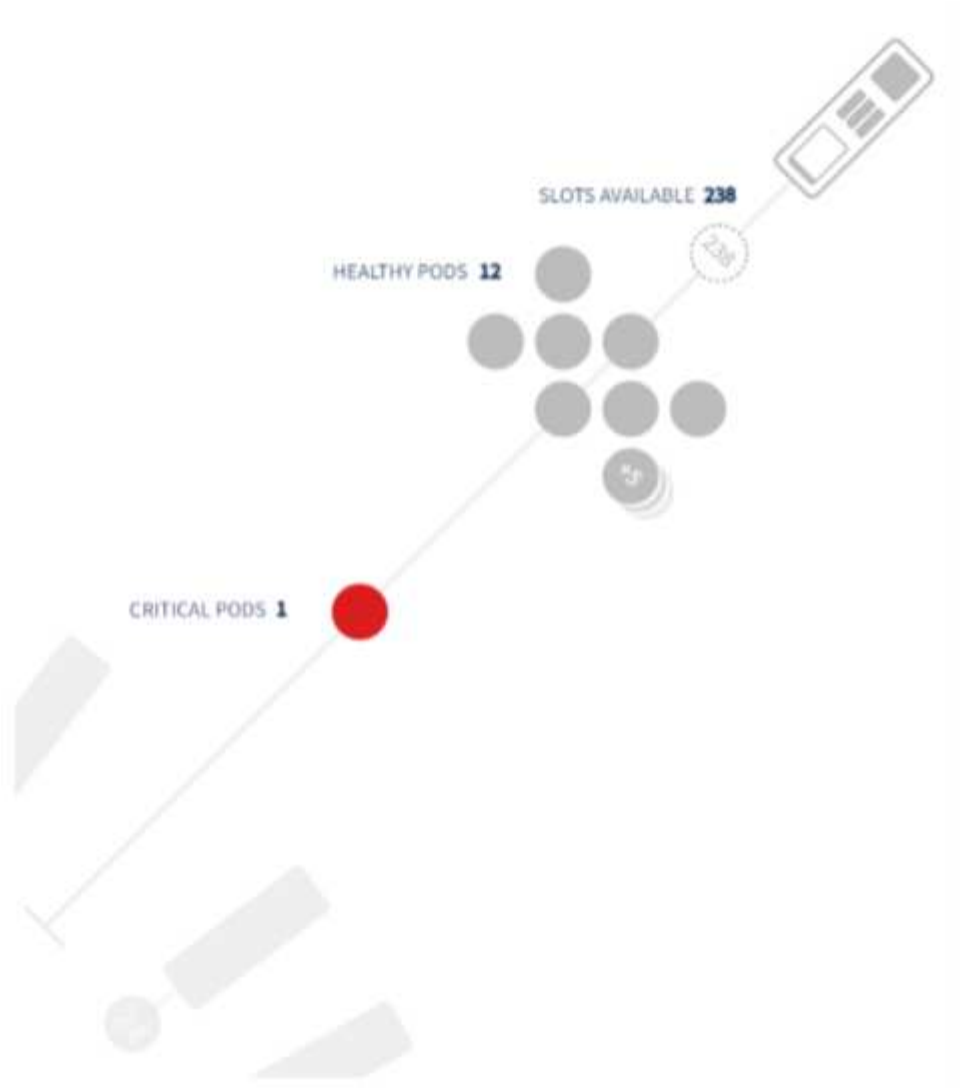
La "ruota" del cluster

La sezione "ruota" del cluster fornisce informazioni sullo stato dei nodi e dei pod, che è possibile analizzare per ulteriori informazioni. Se il cluster contiene più nodi di quelli visualizzabili in quest'area della pagina, sarà possibile ruotare la manopola utilizzando i pulsanti disponibili.

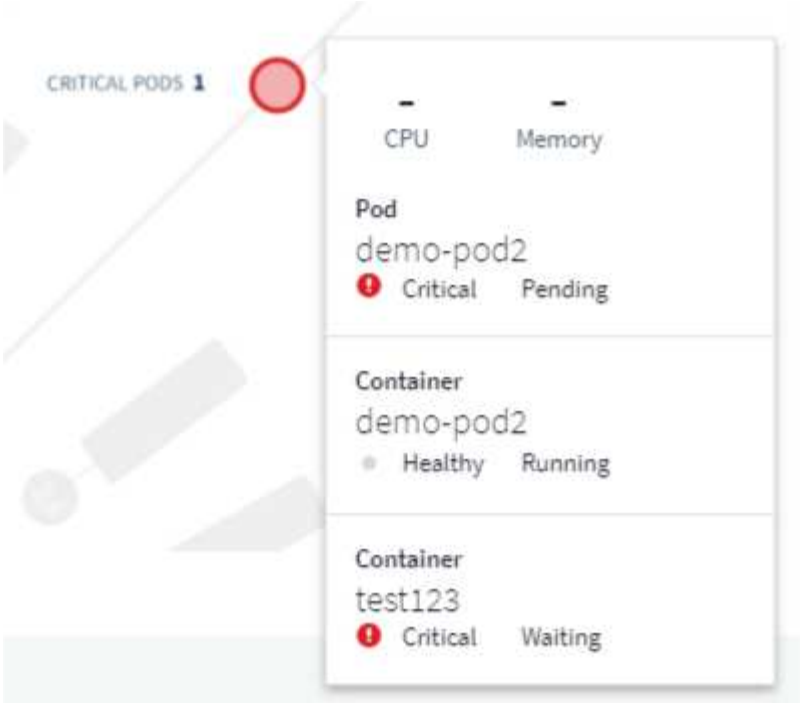
I pod o i nodi di avviso vengono visualizzati in rosso. Le aree di "avvertenza" sono visualizzate in arancione. I pod non pianificati (ovvero non collegati) vengono visualizzati nell'angolo inferiore della "ruota" del cluster.

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

Facendo clic sul pod o sul nodo in tale vista, viene eseguito lo zoom avanti nella vista Expanded Node (nodo espanso).



Da qui, è possibile passare il mouse su un elemento per visualizzare i dettagli relativi a tale elemento. Ad esempio, passando il mouse sul pod critico in questo esempio vengono visualizzati i dettagli relativi a tale pod.



È possibile visualizzare le informazioni relative a filesystem, memoria e CPU passando il mouse sugli elementi Node.

Una nota sugli indicatori

Gli indicatori della memoria e della CPU mostrano tre colori, in quanto indicano *used* in relazione alla *capacità allocabile* e alla *capacità totale*.

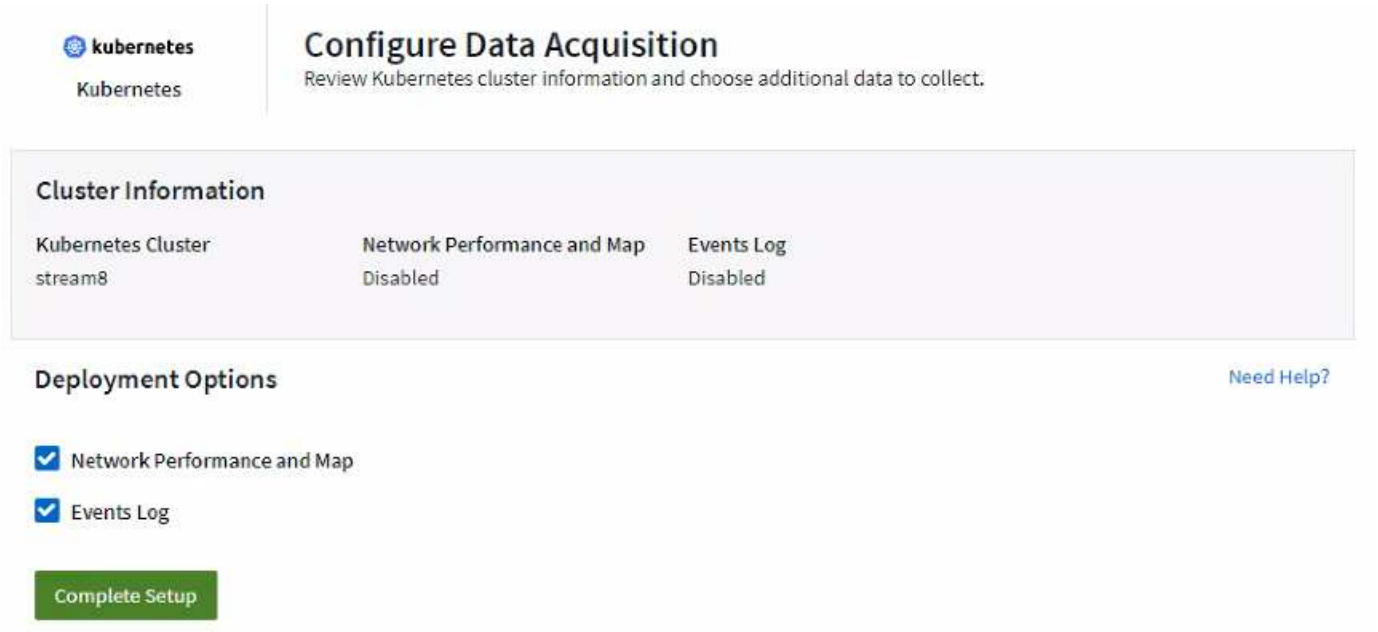
Kubernetes Network Performance Monitoring and Map

Le funzionalità MAP e di Kubernetes Network Performance Monitoring semplificano il troubleshooting mappando le dipendenze tra i servizi (anche denominati workload) e offrono visibilità real-time sulle latenze delle performance di rete e sulle anomalie per identificare i problemi di performance prima che incidano sugli utenti. Questa funzionalità aiuta le organizzazioni a ridurre i costi complessivi analizzando e revisionando i flussi di traffico Kubernetes.

Caratteristiche principali: • La mappa del carico di lavoro presenta le dipendenze e i flussi dei carichi di lavoro di Kubernetes e evidenzia i problemi di rete e di performance. • Monitora il traffico di rete tra pod, carichi di lavoro e nodi Kubernetes; identifica l'origine dei problemi di traffico e latenza. • Riduci i costi complessivi analizzando il traffico di rete in entrata, in uscita, cross-region e cross-zone.

Prerequisiti

Prima di poter utilizzare la mappa e il monitoraggio delle performance di rete di Kubernetes, è necessario aver configurato "NetApp Kubernetes Monitoring Operator"per abilitare questa opzione. Durante l'implementazione dell'operatore, selezionare la casella di controllo "Network Performance and Map" (prestazioni di rete e mappa) per attivarla. È inoltre possibile attivare questa opzione accedendo a una landing page di Kubernetes e selezionando "Modify Deployment" (Modifica distribuzione).



Monitor

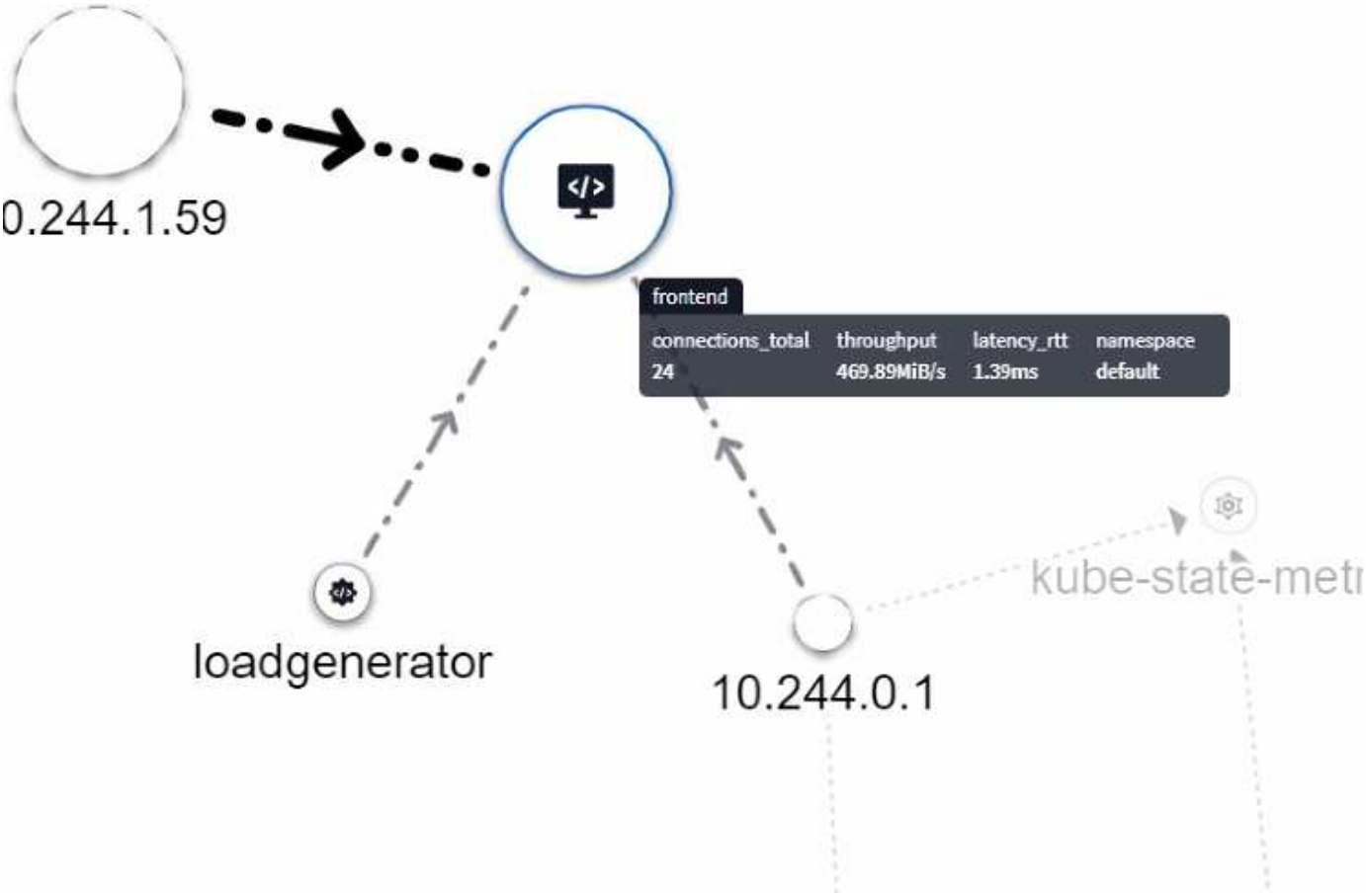
La mappa del carico di lavoro utilizza "monitor" per derivare le informazioni. Data Infrastructure Insights fornisce alcuni monitor Kubernetes predefiniti (si noti che per impostazione predefinita questi possono essere *Paused*). È possibile *riprendere* (ad esempio attivare) i monitor desiderati) oppure creare monitor personalizzati per gli oggetti Kubernetes, che verranno utilizzati anche dalla mappa del carico di lavoro.

È possibile creare avvisi metrici di Data Infrastructure Insights per uno qualsiasi dei tipi di oggetto seguenti. Assicurarsi che i dati siano raggruppati in base al tipo di oggetto predefinito.

- kubernetes.workload
- kubernetes.demonset
- kubernetes.deployment
- 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 tra loro. Le frecce indicano le direzioni del traffico. Passando il mouse su un carico di lavoro vengono visualizzate informazioni riepilogative per tale carico di lavoro, come si può vedere in questo esempio:



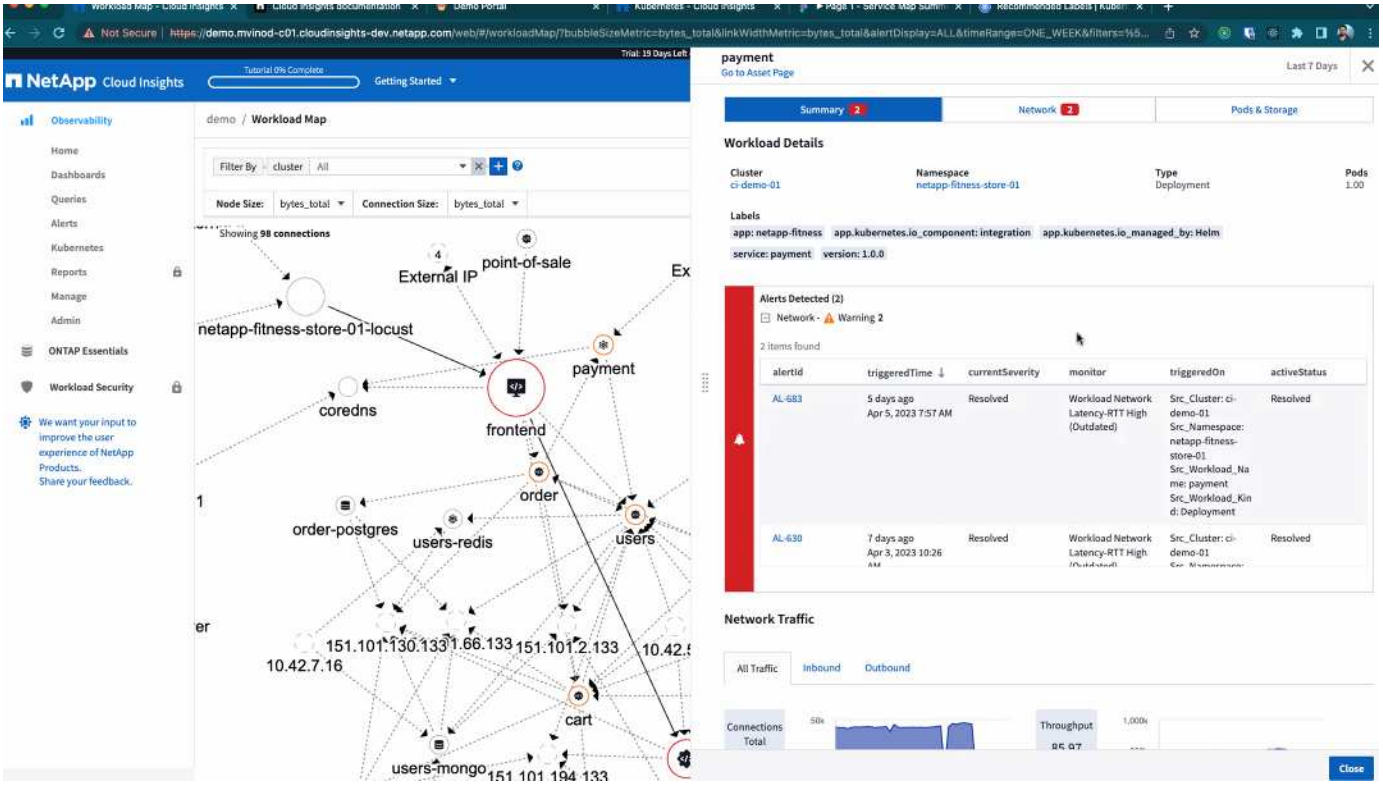
Le icone all'interno dei cerchi rappresentano diversi tipi di servizio. Si noti che le icone sono visibili solo se gli oggetti sottostanti hanno etichette.

La dimensione di ciascun cerchio indica la dimensione del nodo. Si noti che queste dimensioni sono relative, il livello di zoom del browser o le dimensioni dello schermo potrebbero influire sulle dimensioni effettive dei cerchi. Allo stesso modo, lo stile della linea di traffico offre una vista a colpo d'occhio delle dimensioni della connessione; le linee solide in grassetto sono un traffico elevato, mentre le linee tratteggiate sono un traffico minore.

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

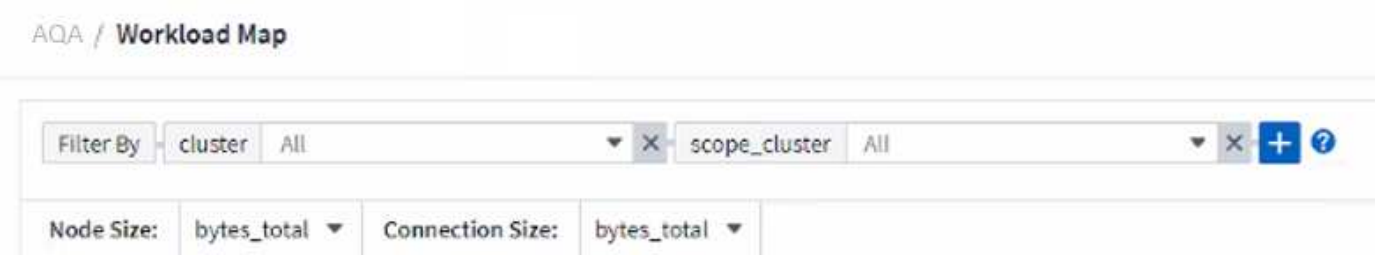
Avvisi e dettagli sul carico di lavoro

I cerchi visualizzati a colori indicano un avviso o un avviso di livello critico per il carico di lavoro. Passare il puntatore del mouse sul cerchio per visualizzare un riepilogo del problema oppure fare clic sul cerchio per aprire un pannello a scorrimento con maggiori dettagli.

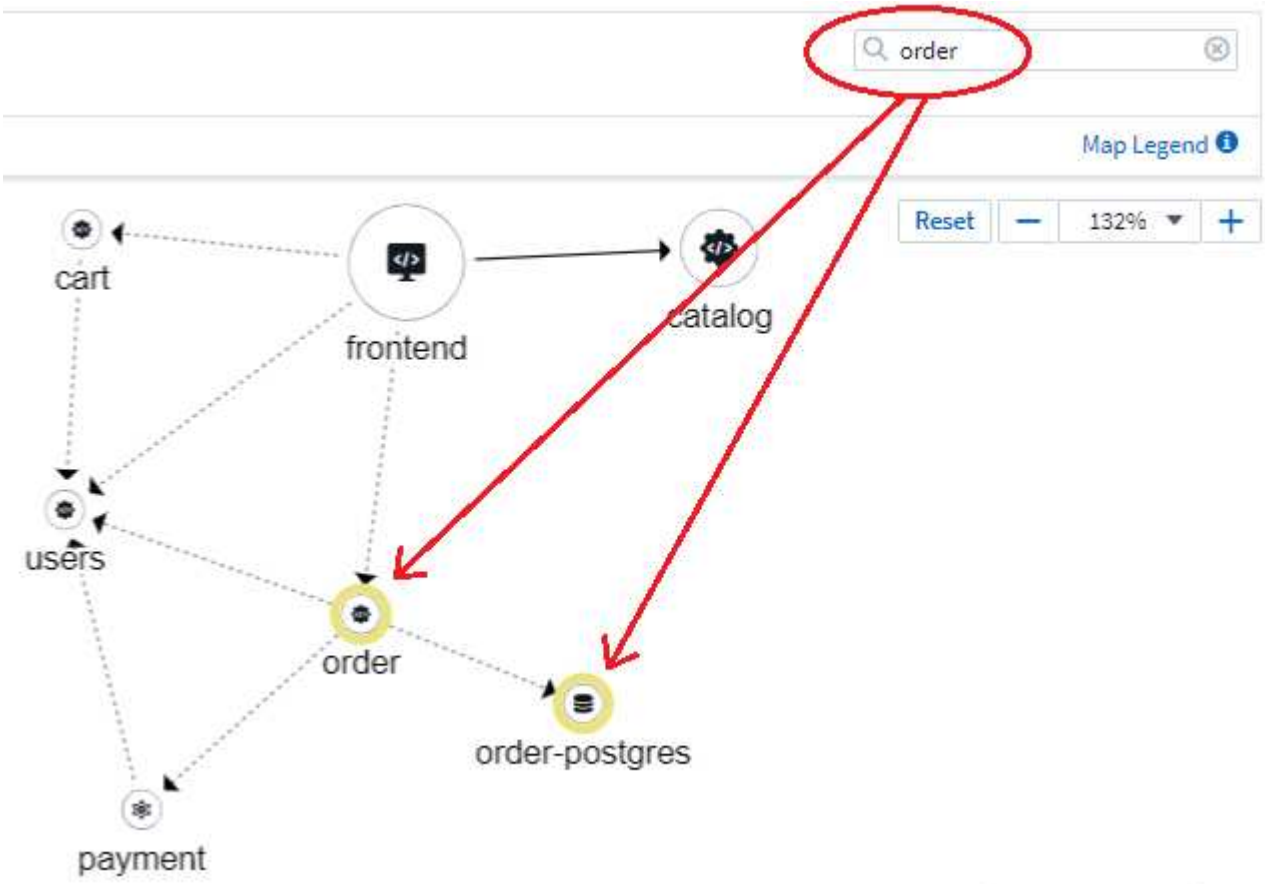


Ricerca e filtraggio

Come per le altre funzionalità di Data Infrastructure Insights, puoi facilmente impostare filtri per concentrarsi su oggetti o attributi specifici del carico di lavoro che desideri.



Allo stesso modo, digitando una stringa nel campo *Find* si evidenzieranno i carichi di lavoro corrispondenti.



Etichette dei carichi 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 dei cerchi). Le etichette sono derivate come segue:

- Nome del servizio/applicazione in esecuzione in termini generici
- Se l'origine 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:
    - frontend

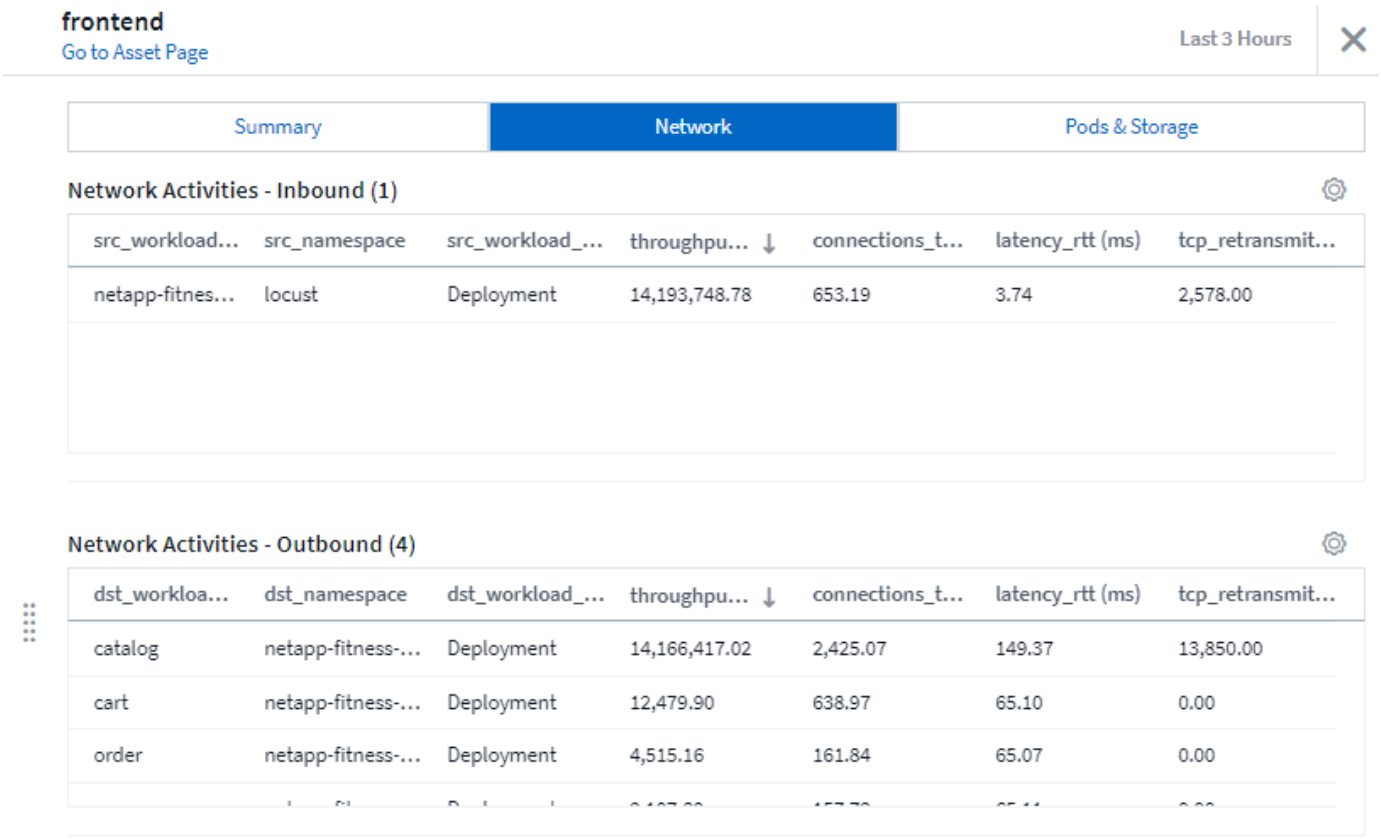
    - back-end
    - database
    - cache
    - coda
    - kafka
- Se l'origine è esterna al cluster kubernetes:
    - Data Infrastructure Insights tenterà di analizzare il nome DNS risolto per estrarre il tipo di servizio.

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

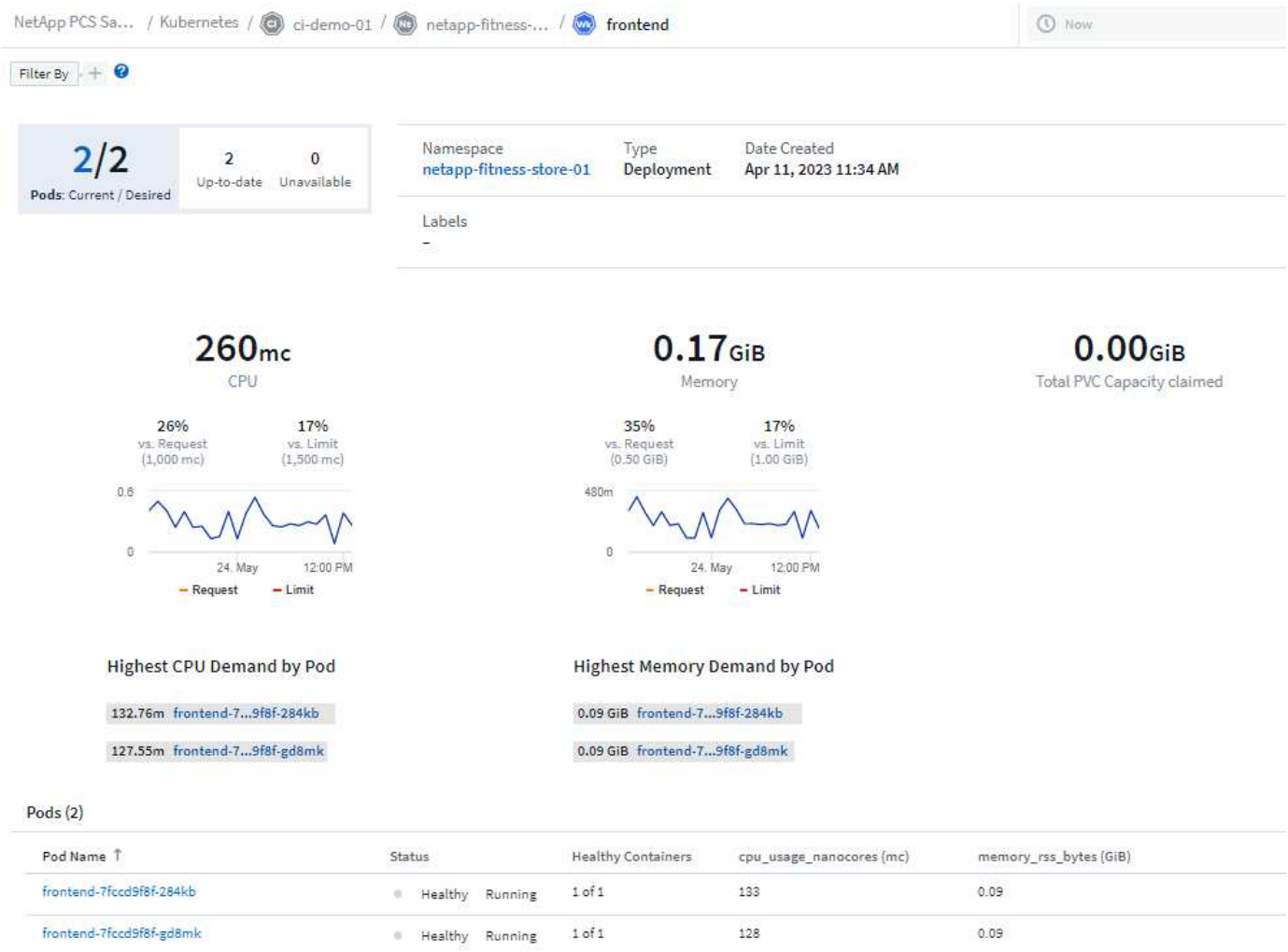
Tuffati in profondità

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

In alternativa, puoi aprire il pannello a scorrimento dei dettagli per visualizzare direttamente la scheda *Summary*, *Network* o *Pod & Storage*.



Infine, selezionando *Go to Asset Page* si apre la landing page dettagliata delle risorse per il carico di lavoro.



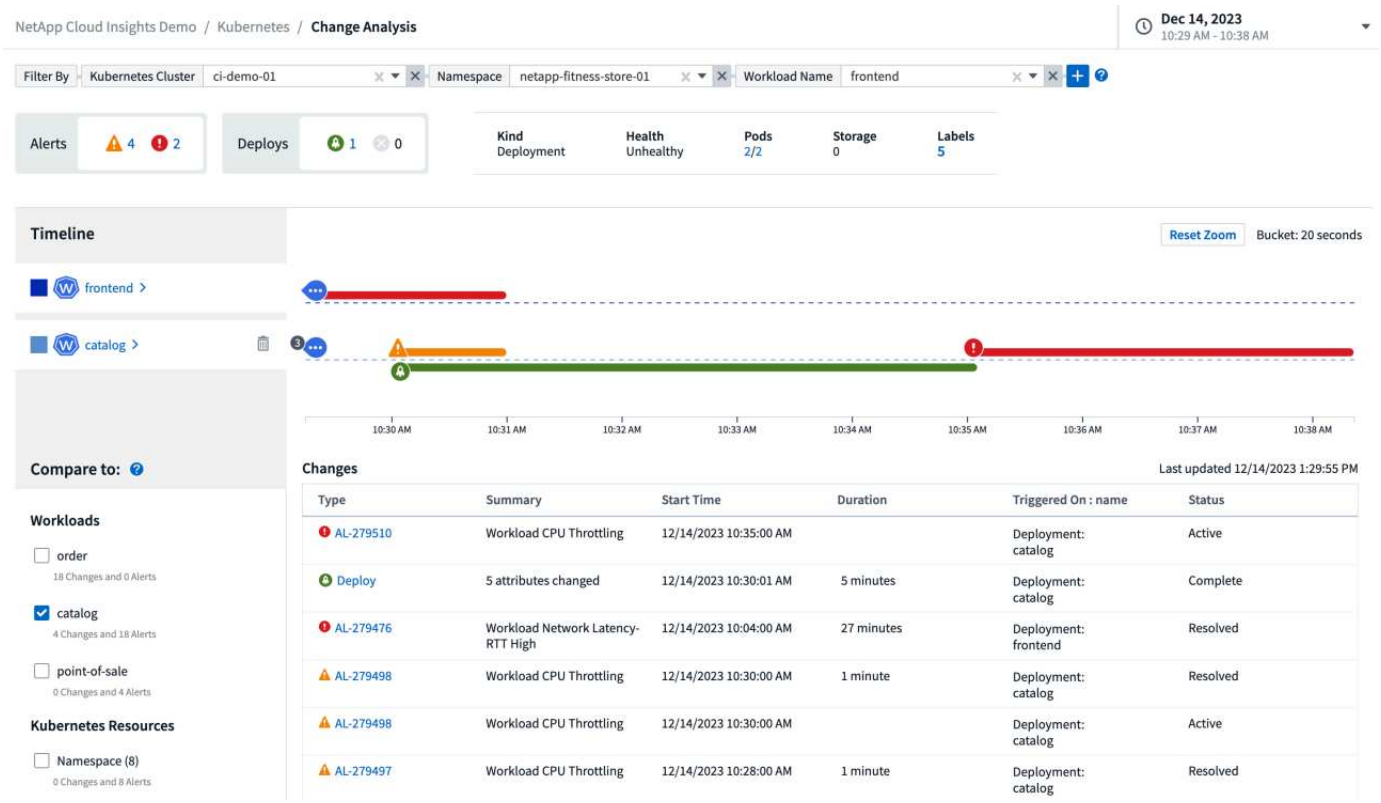
Analytics delle modifiche di Kubernetes

Kubernetes Change Analytics offre una vista completa delle recenti modifiche all'ambiente K8s. Gli avvisi e lo stato dell'implementazione sono a portata di mano. Con Change Analytics, puoi monitorare ogni modifica di implementazione e configurazione e correlarla con lo stato e le performance dei servizi, dell'infrastruttura e dei cluster K8s.

In che modo Change Analysis aiuta?

- Negli ambienti Kubernetes multi-tenant, le interruzioni possono verificarsi a causa di modifiche non configurate correttamente. L'analisi delle modifiche aiuta a questo scopo fornendo un singolo riquadro per visualizzare e correlare lo stato di salute dei carichi di lavoro e le modifiche alla configurazione. Ciò può risultare utile nella risoluzione dei problemi degli ambienti Kubernetes dinamici.

Per visualizzare Kubernetes Change Analytics, accedere a **Kubernetes > Change Analysis**.



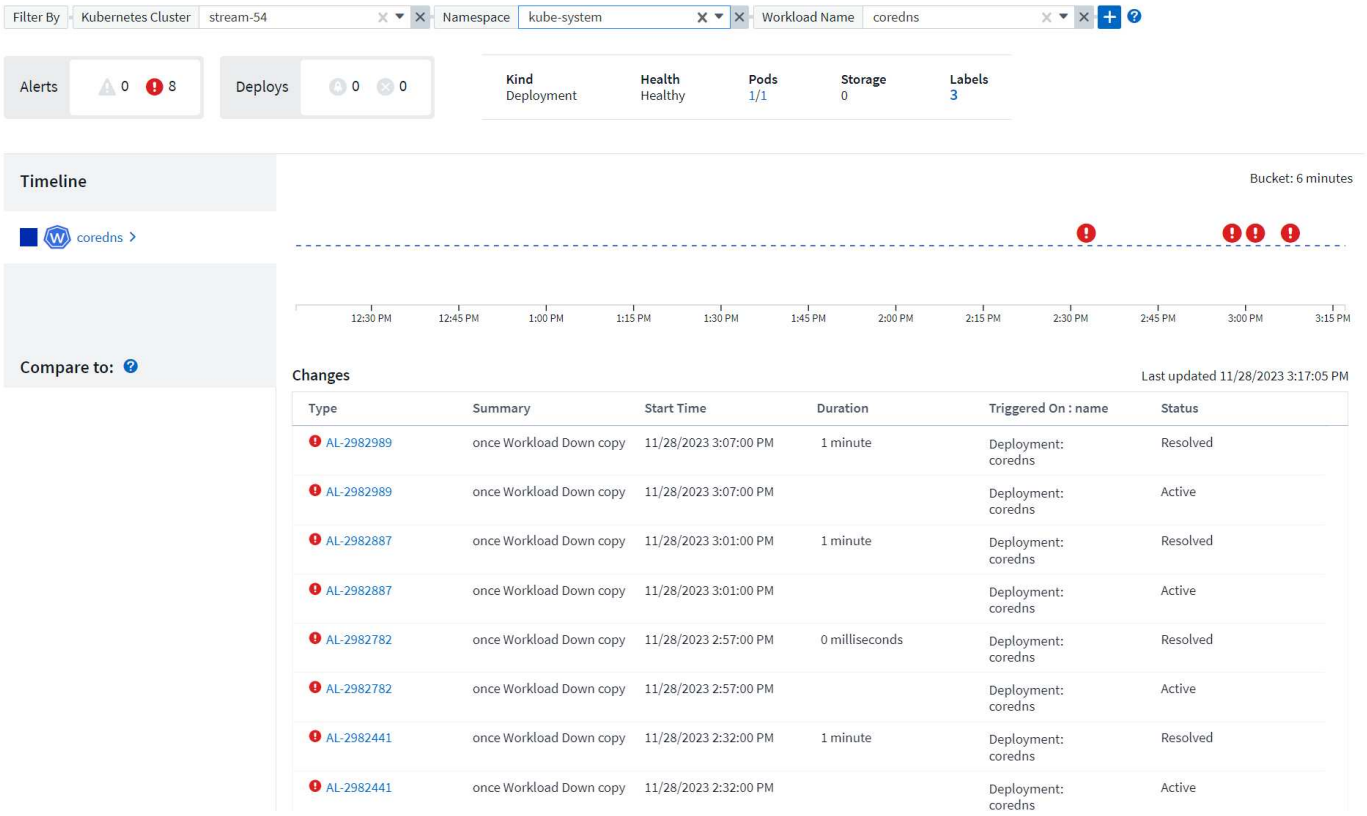
.

La pagina viene aggiornata automaticamente in base all'intervallo di tempo Data Infrastructure Insights attualmente selezionato. Intervalli di tempo più piccoli significano un aggiornamento dello schermo più frequente.

Filtraggio

Come per tutte le funzionalità di Data Infrastructure Insights, il filtraggio della lista di modifiche è intuitivo: Nella parte superiore della pagina, immettere o selezionare valori per il cluster Kubernetes, lo spazio dei nomi o il carico di lavoro oppure aggiungere i propri filtri selezionando il pulsante {+].

Quando si applica un filtro a un cluster, uno spazio dei nomi e un carico di lavoro specifici (insieme agli altri filtri impostati), viene visualizzata una timeline di distribuzione e avvisi per il carico di lavoro nello spazio dei nomi in quel cluster. Ingrandire ulteriormente facendo clic e trascinando il grafico per concentrarsi su un intervallo di tempo più specifico.

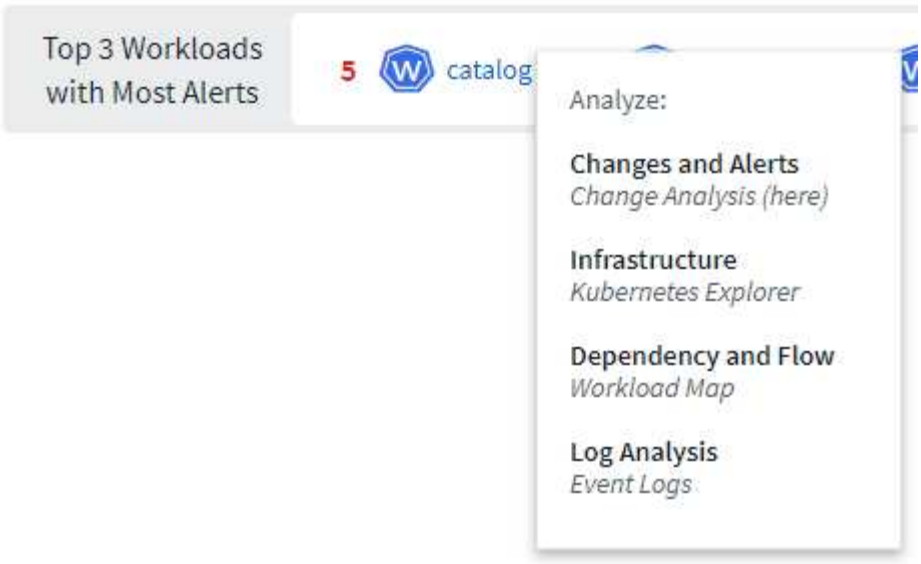


Stato rapido

Al di sotto dell'area di filtraggio sono presenti diversi indicatori di livello alto. A sinistra si trova il numero di avvisi (attenzione e critico). Questo numero include gli avvisi *Active* e *Resolved*. Per visualizzare solo gli avvisi *attivi*, imposta un filtro per "Stato" e scegli "attivo".

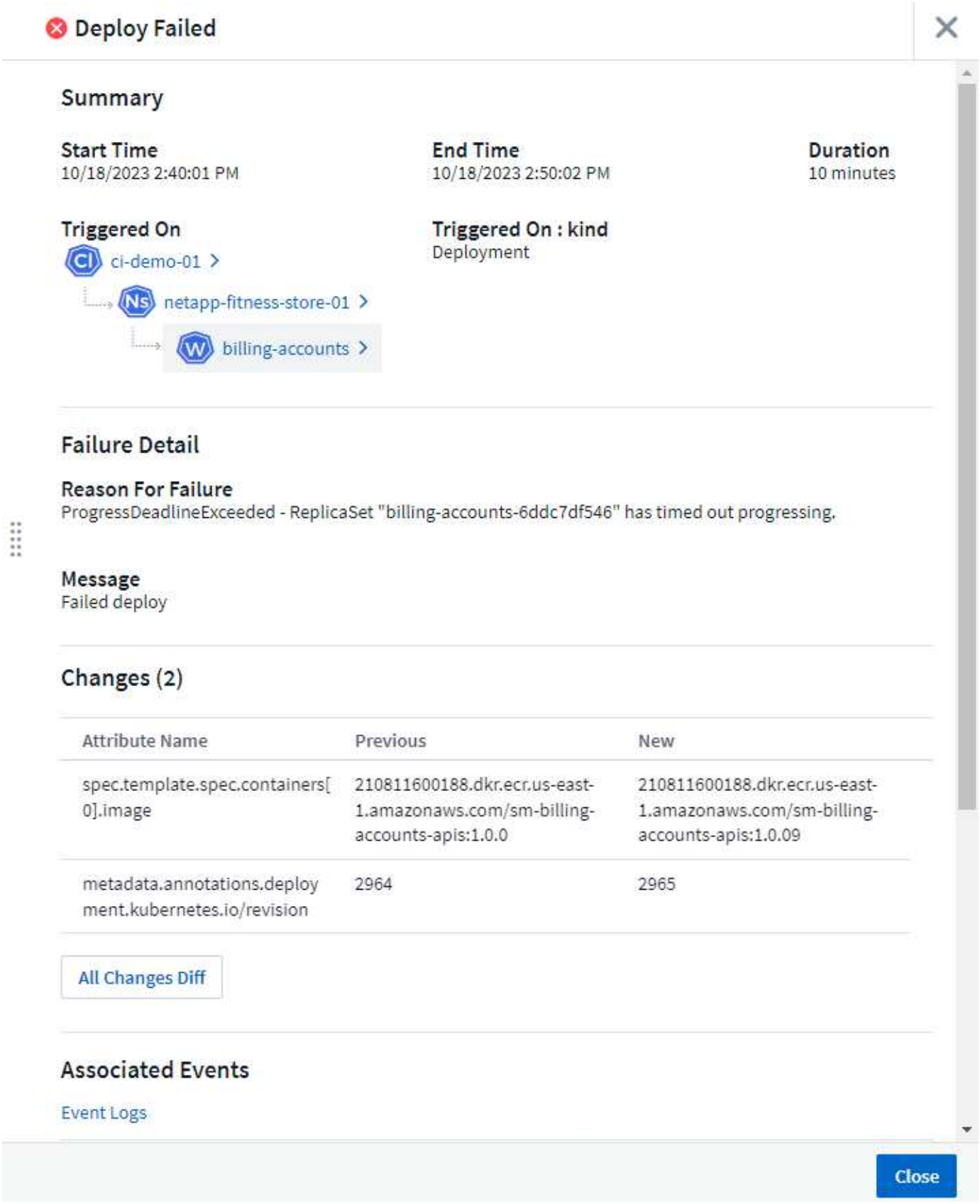
Qui viene visualizzato anche lo stato di distribuzione. Anche in questo caso, l'impostazione predefinita è quella di mostrare il numero di implementazioni *started*, *complete* e *Failed*. Per visualizzare solo le distribuzioni *non riuscite*, impostare un filtro per "Stato" e selezionare "non riuscito".

I primi 3 carichi di lavoro con un maggior numero di avvisi sono i prossimi. Il numero in rosso accanto a ciascun carico di lavoro indica il numero di avvisi relativi a tale carico di lavoro. Fare clic sul collegamento del carico di lavoro per esplorare tramite l'infrastruttura (Kubernetes Explorer), le dipendenze (Mappa del carico di lavoro) o l'analisi del registro (registri eventi).



Pannello Dettagli

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



La selezione di un avviso fornisce dettagli sull'avviso, compreso il monitor che ha attivato l'avviso, nonché un grafico che mostra una timeline visiva per l'avviso.

Informazioni sul copyright

Informazioni sul marchio commerciale