# Kubernetes

## Cloud Insights

NetApp
April 16, 2024

# Sommario

# Kubernetes

## Panoramica del cluster Kubernetes

Cloud Insights Kubernetes Explorer è un potente strumento per visualizzare lo stato generale e l'utilizzo dei cluster Kubernetes e consente di analizzare facilmente le aree di ricerca.

Facendo clic su **Dashboards > Kubernetes Explorer** si apre la pagina Kubernetes Cluster. Questa pagina di panoramica contiene la tabella dei cluster Kubernetes nel tuo ambiente.


| Name | Overall Saturation (%) | CPU Saturation (%) | Memory Saturation (%) | Storage Saturation (%) | Nodes | Pods | Namespaces | Workloads |
|---|---|---|---|---|---|---|---|---|
| self | 56 | 25 | 56 | 31 | 2 | 63 | 18 | 68 |
| setoK3s | 4 | 2 | 3 | 4 | 2 | 9 | 5 | 7 |

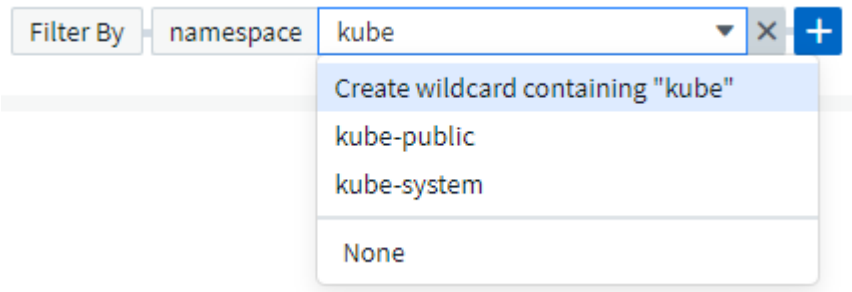
### Elenco dei cluster

L'elenco dei cluster visualizza le seguenti informazioni per ciascun cluster dell'ambiente:

- Cluster **Nome**. Facendo clic sul nome di un cluster, viene aperto il **"pagina dei dettagli"** per quel cluster.
- 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 l'operatore di monitoraggio NetApp Kubernetes

**Prerequisiti:**

- Se si utilizza un repository di docker privato o personalizzato, seguire le istruzioni nella sezione utilizzo di un repository di docker privato o personalizzato
- L'installazione di NetApp Kubernetes Monitoring Operator è supportata con Kubernetes versione 1.20 o successiva.
- Quando Cloud Insights sta monitorando lo storage back-end e Kubernetes viene utilizzato con il runtime del container Docker, Cloud Insights può visualizzare le mappature e le metriche pod-to-PV-to-storage per NFS e iSCSI; altre runtime mostrano solo NFS.
- A partire da agosto 2022, NetApp Kubernetes Monitoring Operator include il supporto per Pod Security Policy (PSP). Se il tuo ambiente utilizza PSP, devi eseguire l'aggiornamento all'ultimo NetApp Kubernetes Monitoring Operator.
- Se si utilizza OpenShift 4.6 o versione successiva, è necessario seguire le istruzioni di OpenShift riportate di seguito oltre a garantire che questi prerequisiti siano soddisfatti.
- Il monitoraggio viene installato solo sui nodi Linux Cloud 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:

| Piattaforma | Etichetta |
|---|---|
| Kubernetes v1.20 e versioni successive | Kubernetes.io/os = linux |
| Rancher + Cattle.io come piattaforma di orchestrazione/Kubernetes | cattle.io/os = linux |

- NetApp Kubernetes Monitoring Operator e le relative dipendenze (telegraf, kube-state-metrics, fluentbit, ecc.) non sono supportate sui nodi che eseguono l'architettura Arm64.
- Devono essere disponibili i seguenti comandi: Curl, kubectl. Il comando docker è necessario per una fase di installazione opzionale. Per ottenere risultati ottimali, aggiungere questi comandi al PERCORSO. Si noti che kubectl deve essere configurato con accesso minimo ai seguenti oggetti kubernetes: Agenti, clusterrolebinding, customresourcedefinizioni, implementazioni, namespace, ruoli, associazioni di ruoli, segreti, serviceaccounts, e servizi. Vedere qui per un file .yaml di esempio con questi privilegi minimi di

ruolo del clusterrole.

- L'host da utilizzare per l'installazione dell'operatore di monitoraggio Kubernetes di NetApp deve avere kubectl configurato per comunicare con il cluster K8s di destinazione e disporre di connettività Internet all'ambiente Cloud Insights.
- Se si utilizza un proxy durante l'installazione o quando si utilizza il cluster K8s da monitorare, seguire le istruzioni nella sezione Configurazione del supporto proxy.
- NetApp Kubernetes Monitoring Operator installa le proprie metriche di stato kube per evitare conflitti con altre istanze. 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).
- Se si sta ridistribuendo l'operatore (ovvero si sta aggiornando o sostituendo), non è necessario creare un token API *new*; è possibile riutilizzare il token precedente.
- Si noti inoltre che se si dispone di un recente NetApp Kubernetes Monitoring Operator installato e si utilizza un token di accesso API rinnovabile, i token in scadenza verranno sostituiti automaticamente da token di accesso API nuovi/aggiornati.
- Monitoraggio della rete:
  - Richiede il kernel Linux versione 4.18.0 e superiore
  - Il sistema operativo Photon non è supportato.

**Configurazione dell'operatore**

Nelle versioni più recenti dell'operatore, le impostazioni modificate più comunemente 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 commentati di alcune impostazioni. Vedere l'elenco di "impostazioni disponibili" per la versione più recente dell'operatore.

È inoltre 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.
```

## Cose importanti da notare prima di iniziare

Se si utilizza un proxy, hanno un repository personalizzato, o stanno utilizzando OpenShift, leggere attentamente le seguenti sezioni.

Leggi anche di Permessi.

Se si sta eseguendo l'aggiornamento da un'installazione precedente, leggere la Aggiornamento in corso

informazioni.

### Configurazione del supporto proxy

Per installare NetApp Kubernetes Monitoring Operator, è possibile utilizzare un proxy nel proprio ambiente in due punti. 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 viene eseguito il frammento all'ambiente Cloud Insights
- Proxy necessario dal cluster Kubernetes di destinazione per comunicare con l'ambiente Cloud Insights

Se si utilizza un proxy per uno o entrambi questi, per installare il monitor operativo Kubernetes di NetApp è necessario prima assicurarsi che il proxy sia configurato per consentire una buona comunicazione con l'ambiente Cloud Insights. Ad esempio, dai server/VM da cui si desidera installare l'operatore, è necessario poter accedere a Cloud Insights e scaricare i file binari da Cloud 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 la comunicazione del cluster Kubernetes con l'ambiente Cloud Insights, installare l'operatore di monitoraggio Kubernetes dopo aver letto tutte le 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 monitoraggio di NetApp Kubernetes estrarrà le immagini container dal repository Cloud Insights. 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 effettua l'accesso al repository Cloud Insights, inserisce tutte le dipendenze dell'immagine per l'operatore e si disconnette dal repository Cloud 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. Assicurarsi che i tag delle immagini e i percorsi delle directory per queste immagini nel repository siano coerenti con quelli nel repository Cloud 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 ulteriori 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: https://docs.netapp.com/us-
en/cloudinsights/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.

#### Permessi

Se il cluster che si sta 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 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

#### 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 tipi di contamini personalizzati sui nodi, impedendo l'esecuzione dei pod su ogni nodo, è possibile creare una **tolleranza** per tali tipi di contamini "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.

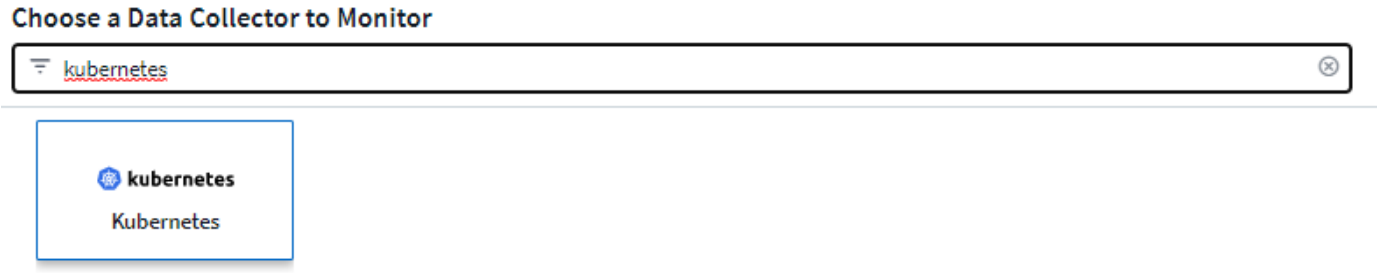
Scopri di più su Kubernetes "Contamini e pedaggi".

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

# Installazione e configurazione dell'operatore di monitoraggio Kubernetes

Cloud Insights offre la raccolta * NetApp Kubernetes Monitoring Operator* (NKMO) per Kubernetes. Quando si aggiunge un data collector, è sufficiente selezionare la sezione Kubernetes.

> Se si dispone dell'edizione federale di Cloud Insights, le istruzioni di installazione e configurazione potrebbero essere diverse da quelle riportate su questa pagina. Seguire le istruzioni in Cloud Insights per installare l'operatore di monitoraggio NetApp Kubernetes.

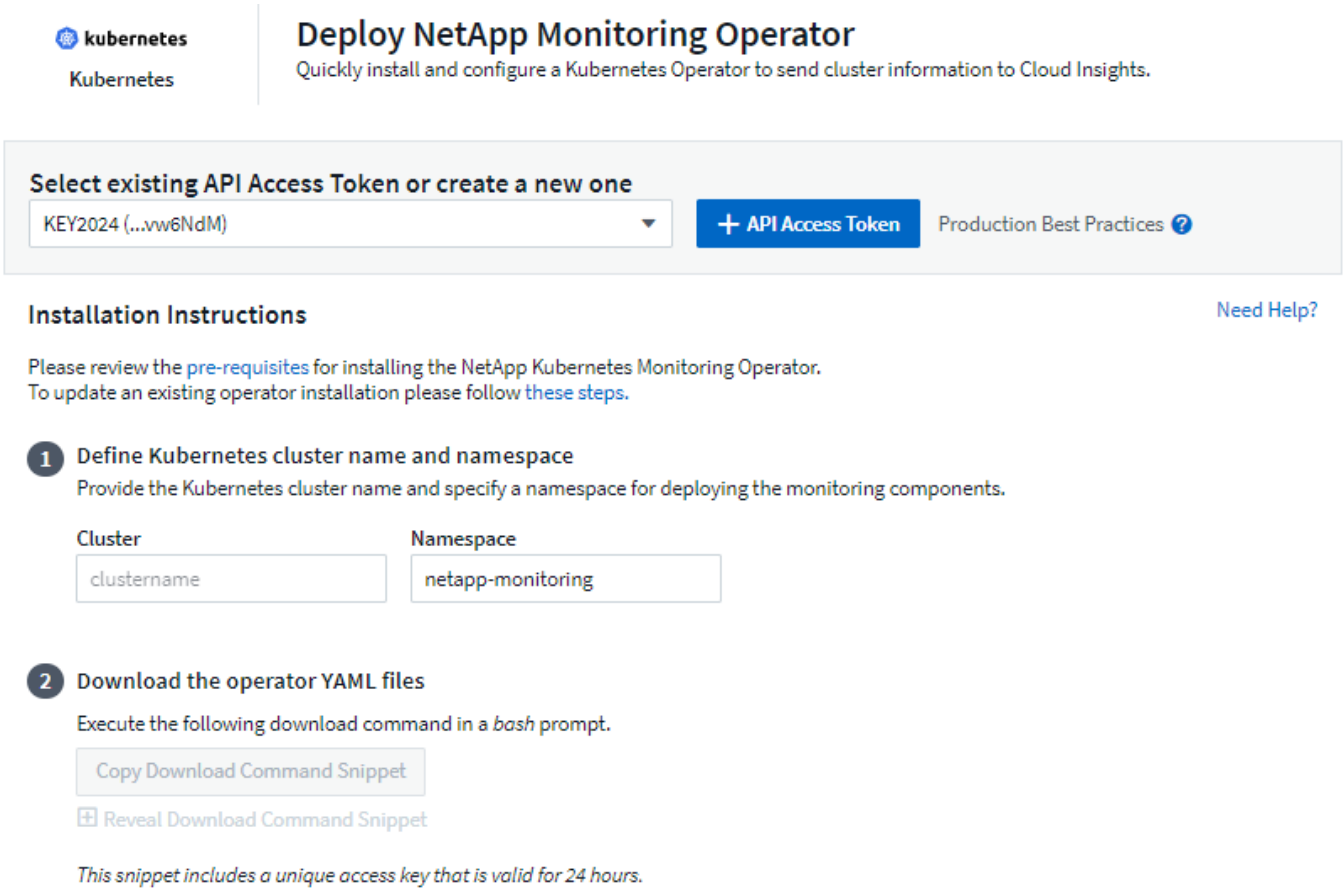



L'operatore Kubernetes e i data collector vengono scaricati dal Registro di Docker di Cloud Insights. Una volta installato, l'operatore gestisce quindi tutti i collettori compatibili con l'operatore implementati nei nodi del cluster Kubernetes per acquisire i dati, inclusa la gestione del ciclo di vita di tali collettori. In seguito a questa catena, i dati vengono acquisiti dai collettori e inviati a Cloud Insights.

## Prima di installare NetApp Kubernetes Monitoring Operator

Leggere il **"Prima dell'installazione o dell'aggiornamento"** Documentazione pre-requisiti prima di installare o aggiornare l'operatore di monitoraggio Kubernetes NetApp.

## Installazione di NetApp Kubernetes Monitoring Operator



**3 Optional: Upload the operator images to your private repository**

By default, the operator pulls container images from the Cloud Insights repository. To use a private repository, download the required images using the Image Pull command. Then upload them to your private repository maintaining the same tags and directory structure.
Finally, update the image paths in operator-deployment.yaml and the docker repository settings in operator-config.yaml.
For more information review the documentation.

Copy Image Pull Snippet

Reveal Image Pull Snippet

Copy Repository Password

Reveal Repository Password

*This password is valid for 24 hours.*

**4 Optional: Review available configuration options**

Configure custom options such as proxy and private repository settings. Review the instructions and available options.

**5 Deploy the operator (create new or upgrade existing)**

Execute the *kubectl* snippet to apply the following operator YAML files.

- operator-setup.yaml - Create the operator's dependencies.
- operator-secrets.yaml - Create secrets holding your API key.
- operator-deployment.yaml, operator-cr.yaml - Deploy the NetApp Kubernetes Monitoring Operator.
- operator-config.yaml - Apply the configuration settings if not already present.

Copy kubectl Apply Snippet

Reveal kubectl Apply Snippet

After deploying the operator, **delete or securely store operator-secrets.yaml.**

**6** Next

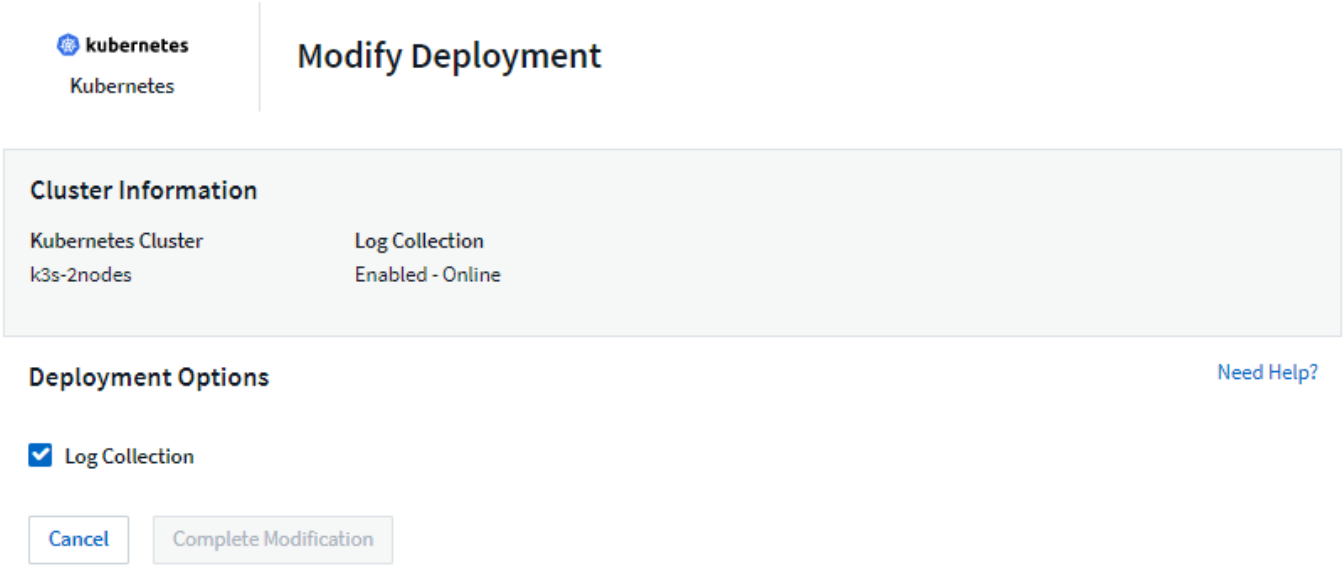
**Procedura per installare NetApp Kubernetes Monitoring Operator Agent su Kubernetes:**

1. Immettere un nome cluster e uno spazio dei nomi univoci. Se lo sei aggiornamento in corso Da un operatore Kubernetes precedente, utilizzare lo stesso nome del 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. 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*.

Scopri di più configurazione del proxy.

Scopri di più utilizzando un repository di docker personalizzato/privato.

La raccolta dei log EMS di Kubernetes è attivata per impostazione predefinita quando si installa NetApp Kubernetes Monitoring Operator. Per disattivare questa raccolta dopo l'installazione, fare clic sul pulsante **Modify Deployment** (Modifica distribuzione) nella parte superiore della pagina dei dettagli del cluster Kubernetes e deselezionare "Log collection" (raccolta log).




Questa schermata mostra anche lo stato corrente della raccolta dei log. Di seguito sono riportati i possibili stati:

- Disattivato
- Attivato
- Enabled (attivato) - Installazione in corso
- Abilitato - non in linea
- Abilitato - Online
- Errore - le autorizzazioni della chiave API non sono sufficienti

## Aggiornamento in corso

### Aggiornamento all'ultimo NetApp 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.
  - Assicurati di sì estrarre le immagini container più recenti se si utilizza un repository personalizzato.

Se AgentConfiguration non esiste:

- Prendere nota del nome del cluster riconosciuto da Cloud Insights (se lo spazio dei nomi non è il monitoraggio netapp predefinito, sostituire lo spazio dei nomi 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-netapp-kubernetes-monitoring-operator,Disinstallare>>
L'operatore esistente.
* <<installing-the-netapp-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.
  - Assicurati di sì estrarre le immagini container più recenti se si utilizza un repository personalizzato.

## Arresto e avvio di NetApp Kubernetes Monitoring Operator

Per arrestare NetApp Kubernetes Monitoring Operator:

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

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

## Disinstallazione in corso

### Per rimuovere NetApp Kubernetes Monitoring Operator

Si noti che lo spazio dei nomi predefinito per NetApp 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 automaticamente le metriche dello stato kube, senza richiedere alcuna interazione da parte dell'utente.

### Contatori di metriche di stato kube

Utilizzare i seguenti collegamenti per accedere alle informazioni relative ai contatori delle metriche di stato del kube:

1. "Metriche di ConfigMap"
2. "Metriche DemonSet"
3. "Metriche di implementazione"
4. "Metriche di ingresso"
5. "Metriche dello spazio dei nomi"
6. "Metriche del nodo"
7. "Metriche di volume persistenti"
8. "Metriche delle richieste di rimborso per volumi persistenti"
9. "Metriche pod"
10. "Metriche ReplicaSet"
11. "Metriche segrete"
12. "Metriche del servizio"
13. "Metriche StatefulSet"

---

```
 == Configuring the Operator
Nelle versioni più recenti dell'operatore, le impostazioni modificate più
comunemente 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 commentati di alcune impostazioni. Vedere
l'elenco di
xref:{relative_path}telegraf_agent_k8s_config_options.html["impostazioni
disponibili"] per la versione più recente dell'operatore.
```

È inoltre 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

Per installare NetApp Kubernetes Monitoring Operator, è possibile utilizzare un proxy nel proprio ambiente in

due punti. 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 viene eseguito il frammento all'ambiente Cloud Insights
- Proxy necessario dal cluster Kubernetes di destinazione per comunicare con l'ambiente Cloud Insights

Se si utilizza un proxy per uno o entrambi questi, per installare il monitor operativo di NetApp Kubernetes è necessario prima assicurarsi che il proxy sia configurato per consentire una buona comunicazione con l'ambiente Cloud Insights. Se si dispone di un proxy e si può accedere a Cloud 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 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 la comunicazione del cluster Kubernetes con l'ambiente Cloud Insights, installare l'operatore di monitoraggio Kubernetes dopo aver letto tutte le 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 monitoraggio di NetApp Kubernetes estrarrà le immagini container dal repository Cloud Insights. 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 effettua l'accesso al repository Cloud Insights, inserisce tutte le dipendenze dell'immagine per l'operatore e si disconnette dal repository Cloud 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. Assicurarsi che i tag delle immagini e i percorsi delle directory per queste immagini nel repository siano coerenti con quelli nel repository Cloud 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:<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 ulteriori 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: https://docs.netapp.com/us-
en/cloudinsights/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.

## Una nota sui segreti

Per rimuovere l'autorizzazione per l'operatore di monitoraggio Kubernetes NetApp per 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 dei checksum di Kubernetes

Il programma di installazione dell'agente Cloud Insights esegue controlli di integrità, ma alcuni utenti potrebbero voler eseguire le proprie verifiche prima di installare o applicare gli artefatti scaricati. Per eseguire un'operazione di solo download (invece del download e dell'installazione predefiniti), questi utenti possono modificare il comando di installazione dell'agente ottenuto dall'interfaccia utente e rimuovere l'opzione finale di "installazione".

Attenersi alla seguente procedura:

1. Copiare il frammento del programma di installazione dell'agente come indicato.
2. Invece di incollare il frammento in una finestra di comando, incollarlo in un editor di testo.
3. Rimuovere il file "--install" finale dal comando.
4. Copiare l'intero comando dall'editor di testo.
5. Incollarlo nella finestra di comando (in una directory di lavoro) ed eseguirlo.
    - Download e installazione (impostazione predefinita):

```
 installerName=cloudinsights-rhel_centos.sh … && sudo -E -H
./$installerName --download --install
** Solo download:
```

```
installerName=cloudinsights-rhel_centos.sh … && sudo -E -H
./$installerName --download
```

Il comando di solo download scaricherà tutti gli artefatti richiesti da Cloud Insights nella directory di lavoro. Gli artefatti includono, ma non possono essere limitati a:

- uno script di installazione
- un file di ambiente
- File YAML
- un file checksum firmato (sha256.signed)
- Un file PEM (netapp_cert.pem) per la verifica della firma

Lo script di installazione, il file di ambiente e i file YAML possono essere verificati utilizzando l'ispezione visiva.

Il file PEM può essere verificato confermando che l'impronta digitale è la seguente:

```
 1A918038E8E127BB5C87A202DF173B97A05B4996
In particolare,
```

```
 openssl x509 -fingerprint -sha1 -noout -inform pem -in netapp_cert.pem
Il file checksum firmato può essere verificato utilizzando il file PEM:
```

```
 openssl smime -verify -in sha256.signed -CAfile netapp_cert.pem -purpose
any
Una volta verificati correttamente tutti gli artefatti, l'installazione
dell'agente può essere avviata eseguendo:
```

```
sudo -E -H ./<installation_script_name> --install
```

## Risoluzione dei problemi

Alcune cose da provare in caso di problemi durante la configurazione dell'operatore di monitoraggio di NetApp 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. È necessario utilizzare Telegraf versione 2.0 o successiva e lo storage del cluster Kubernetes deve essere monitorato attivamente da Cloud Insights. |

| Problema: | Prova: |
| --- | --- |
| Nei registri vengono visualizzati messaggi simili a quelli riportati di seguito:<br><br>E0901 15:21:39,962145 1 Reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: Impossibile elencare *v1.MutatingWebhookConfigurazione: Il server non ha trovato la risorsa richiesta<br>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 ha trovato la risorsa richiesta (get leases.Coordination.k8s.io)<br>ecc. | Questi messaggi possono verificarsi se si utilizza kube-state-metrics versione 2.0.0 o superiore con versioni di Kubernetes inferiori alla 1.20.<br><br>Per ottenere la versione di Kubernetes:<br><br>*kubectl version*<br><br>Per ottenere la versione kube-state-metrics:<br><br>*kubectl get deploy/kube-state-metrics -o jsonpath='{..image}'*<br><br>Per evitare che questi messaggi si verifichino, gli utenti possono modificare la distribuzione delle metriche dello stato-kube per disabilitare i seguenti leasing:<br><br>*mutatingwebhookconfigurations validatingwebhookconfigurations volumeattachments resources*<br><br>In particolare, possono utilizzare il seguente argomento CLI:<br><br>resources=certificatesigningrequires,configmaps,cronjob,daemonset, deployments,endpoints,horizontalpodautoscalers,ingresses,job,limitrange, namespace,networkpolicy,node,persistentvolumeclaims<br><br>L'elenco delle risorse predefinito è:<br><br>"certificatesigningrequests,configmaps,cronjob,daemonsets,deployments, endpoint,horizontalpodautoscalers,ingresses,job,leases,limitrange, mutatingwebhookconfigurations,namespaces,network policy,nodi, persistentvolumeclaimes,durentvolumetsets,poddisruptionbudgets,pods,replicasets, replicationstoricasets,replicationfors,storeforcsets,servizi,storeforcsets,storeforcsets convalidatingwebhookconfigurations,volumeattachments" |

| Problema: | Prova: |
| --- | --- |
| Vengono visualizzati messaggi di errore di Telegraf simili ai seguenti, ma Telegraf si avvia ed esegue:<br><br>Oct 11 14:23:41:00 ip-172-31-39-47 systemd[1]: Avviato l'agente server basato su plugin per la generazione di rapporti sulle metriche in InfluxDB.<br>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<br>che: permesso negato. Ignorato\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120"<br>Ott 11 14:23:41:00 ip-172-31-39-47 telegraf[1827]: Time="2021-10-11T14:23:41Z" level=error msg="Impossibile aprire. Ignorato. aprire /etc/telegraf/.cache/snowflake/ocsp_response_cache.j son: no<br>File o directory\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120"<br>Oct:23:41:ip-172-31-39-47:11 14 telegraf[1827]: 2021-10-11T14:23:41Z i! Avvio di Telegraf 1.19.3 | Si tratta di un problema noto. Fare riferimento a. "Questo articolo di GitHub" per ulteriori dettagli. Finché Telegraf è in funzione, gli utenti possono ignorare questi messaggi di errore. |
| In Kubernetes, i pod Telegraf riportano il seguente errore:<br>"Errore durante l'elaborazione delle informazioni sui mount stats: 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 a: https://docs.netapp.com/us-en/cloudinsights/task_config_telegraf_agent_k8s.html#openshift-instructions. |
| In Kubernetes, il pod Telegraf ReplicaSet riporta il seguente errore:<br><br>[inputs.prometheus] errore nel plugin: Impossibile caricare keypair /etc/kuowski/pki/etcd/server.crt:/etc/kuowski/pki/etcd/s erver.key: Aprire /etc/kuowski/pki/etcd/server.crt: Nessun file o directory di questo tipo | 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.<br><br>Ad esempio, modificare ReplicaSet…<br><br>kubectl edit rs telegraf-rs<br><br>…e aggiungere le tolleranze appropriate alle specifiche. Quindi, riavviare il pod ReplicaSet. |

| Problema: | Prova: |
| --- | --- |
| 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'ultimo NetApp Kubernetes Monitoring Operator. Per eseguire l'aggiornamento all'NKMO corrente con il supporto per PSP/PSA, procedere come segue:<br><br>1. Disinstallare l'operatore di monitoraggio precedente:<br><br>kubectl delete agent-monitoring-netapp -n monitoring<br>kubectl elimina ns monitoraggio netapp<br>kubectl cancella crd agents.monitoring.netapp.com<br>kubectl elimina agente-manager-ruolo-agente-proxy-ruolo-agente-metrica-lettore<br>kubectl elimina agente di associazione-manager-agente di legame-proxy-agente di legame-cluster-admin-rolebinding<br><br>2. Installare la versione più recente dell'operatore di monitoraggio. |
| Ho riscontrato problemi nel tentativo di implementare NKMO e ho utilizzato PSP/PSA. | 1. Modificare l'agente utilizzando il seguente comando:<br><br>kubectl -n <name-space> edit agent<br><br>2. Contrassegnare 'sicurezza-policy-enabled' come 'false'. In questo modo verranno disabilitati i criteri di sicurezza Pod e l'ammissione alla sicurezza Pod e verrà consentito l'implementazione di NKMO. Confermare utilizzando i seguenti comandi:<br><br>Kubectl Prendi psp (dovrebbe mostrare la politica di sicurezza del Pod rimossa)<br>kubectl get all -n <namespace> |
| grep -i psp (dovrebbe mostrare che non si trova nulla) | Errori "ImagePullBackoff" rilevati |
| Questi errori possono essere rilevati se si dispone di un repository di docker personalizzato o privato e non si è ancora configurato NetApp Kubernetes Monitoring Operator per 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.<br><br>`kubectl -n netapp-monitoring get all`<br>`kubectl -n netapp-monitoring describe all`<br>`kubectl -n netapp-monitoring logs <monitoring-operator-pod> --all-containers=true`<br>`kubectl -n netapp-monitoring logs <telegraf-pod> --all-containers=true` | I pod Net-Observer (Workload Map) nello spazio dei nomi NKMO si trovano in CrashLoopBackOff |
| Questi pod corrispondono al data collector Workload Map per l'osservabilità della rete. Provare a effettuare le seguenti operazioni:<br>• Controllare i log di uno dei pod per confermare la versione minima del kernel. Ad esempio:<br><br>----<br>{"ci-tenant-id":"your-tenant-id","collector-cluster":"your-k8s-cluster-name","environment":"prod","level":"error","msg":"failed in validation. Motivo: La versione del kernel 3.10.0 è inferiore alla versione minima del kernel 4.18.0","Time":"2022-11-09T08:23:08Z"}<br>----<br><br>• 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 nello spazio dei nomi NKMO (impostazione predefinita: monitoraggio netapp), ma non vengono visualizzati dati nell'interfaccia utente per la mappa del carico 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-osservatore nello spazio dei nomi NKMO sono in stato Pending |

| Problema: | Prova: |
| --- | --- |
| NET-osservatore è un DemonSet che esegue un pod in ogni nodo del cluster k8s.<br>• Notare il pod che si trova nello stato in sospeso e controllare 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. | Vedo quanto segue nei miei log subito dopo l'installazione dell'operatore di monitoraggio NetApp Kubernetes:<br><br>[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: lookube-state-metrics.<namespace>.svc.cluster.local: no tale host |
| 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 di metriche dello stato kube implementata con l'operatore di monitoraggio Kubernetes di NetApp 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). |

| Problema: | Prova: |
| --- | --- |
| Modificare la sezione telegraf in *AgentConfiguration* e impostare *dockerMetricCollectionEnabled* su false. Per ulteriori dettagli, fare riferimento al manuale dell'operatore "opzioni di configurazione".<br><br>NOTA: se si utilizza l'Edizione Federale di Cloud Insights, gli utenti con restrizioni sull'uso di *su* non potranno raccogliere metriche di docker perché l'accesso al socket di docker richiede l'esecuzione del contenitore di telegraf come root o l'utilizzo di *su* per aggiungere l'utente di telegraf al gruppo di docker. La raccolta di metriche Docker e l'utilizzo di *su* sono attivati per impostazione predefinita; per disabilitare entrambi, rimuovere la voce *telegraf.docker* nel file *AgentConfiguration*:<br><br>…<br>specifiche:<br>…<br>telegraf:<br>…<br>- nome: docker<br>modalità di esecuzione:<br>- DaemonSet<br>sostituzioni:<br>CHIAVE:<br>DOCKER_UNIX_SOCK_PLACEHOLDER<br>valore: unix://run/docker.sock<br>…<br>… | Nei registri di Telegraf vengono visualizzati messaggi di errore ricorrenti simili a quelli riportati di seguito:<br> <br> E! [Agent] Error writing to outputs.http: Post "<a href="https://<tenant_url>/rest/v1/lake/ingest/influxdb":" class="bare">https://<tenant_url>/rest/v1/lake/ingest/influxdb":</a> Scadenza contesto superata (timeout client superato in attesa di intestazioni) |
| Modificare la sezione telegraf in *AgentConfiguration* e impostare *dockerMetricCollectionEnabled* su false. Per ulteriori dettagli, fare riferimento al manuale dell'operatore "opzioni di configurazione". | Mancano i dati *involvedobject* per alcuni registri eventi. |
| Assicurarsi di aver seguito i passaggi descritti in "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, Cloud Insights ha ridefinito l'operatore per servire meglio i nostri utenti; affinché tali modifiche siano completamente adottate, è necessario rimuovere il vecchio operatore e. installare il nuovo. | I miei eventi kuowski hanno inaspettatamente smesso di segnalare a Cloud Insights. |
| Recuperare il nome del pod dell'esportatore di eventi:<br><br>`kubectl -n netapp-monitoring get pods | grep event-exporter |

| Problema: | Prova: |
| --- | --- |
| awk '{print $1}' | sed 's/event-exporter./event-exporter/'`<br>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"<br><br>….<br>fluent-bit:<br>…<br>- name: event-exporter-ci<br>substitutions:<br>- key: LOG_FILE<br>values:<br>- /var/log/containers/netapp-ci-event-exporter*.log<br>…<br>….<br>In alternativa, si può anche disinstallazione e. reinstallare l'agente. |
| Sto vedendo i pod implementati dal crash dell'operatore di monitoring NetApp Kubernetes a causa di risorse insufficienti. | Fare riferimento all'operatore di monitoraggio Kubernetes NetApp "opzioni di configurazione" Per aumentare i limiti di CPU e/o memoria in base alle esigenze. |

Per ulteriori informazioni, consultare "Supporto" o in "Matrice di supporto Data Collector".

# Opzioni di configurazione dell'operatore di monitoraggio NetApp Kubernetes

Il "NetApp Kubernetes Monitoring Operator" è possibile personalizzare l'installazione e la configurazione.

La tabella seguente elenca le possibili opzioni 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 override dockerRepo per estrarre le immagini dai repos del docker privato dei clienti rispetto a Cloud Insights docker repo. Il valore predefinito è Cloud Insights docker repo |
| | DockerImagePullSecret | Facoltativo: Un segreto per i clienti privati |

| Componente | Opzione | Descrizione |
| --- | --- | --- |
| | Nome cluster | Campo di testo libero che identifica in modo univoco un cluster in tutti i cluster dei clienti. Questo dovrebbe essere unico in un tenant Cloud Insights. Il valore predefinito è quello che il cliente inserisce nell'interfaccia utente per il campo "Cluster Name" (Nome cluster) |
| | proxy<br><br>Formato:<br><br>proxy:<br><br>server:<br>porta:<br>nome utente:<br>password:<br>NoProxy:<br>IsTelegrafProxyEnabled:<br>IsAuxyEnabled:<br>IsFluentbitProxyEnabled:<br>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 |
| | RsMemRequest | Richiesta di memoria per telegraf rs |
| | DockerMountPoint | Un override per il percorso dockerMountPoint. Questo vale per le installazioni di docker non standard su piattaforme k8s come la fonderia cloud |
| | DockerUnixSocket | Un override per il percorso dockerUnixSocket. Questo vale per le installazioni di docker non standard su piattaforme k8s come la fonderia cloud. |
| | CrioSockPath | Un override per il percorso di crioSockPath. Questo vale per le installazioni di docker non standard su piattaforme k8s come la fonderia cloud. |

| Componente | Opzione | Descrizione |
| --- | --- | --- |
| | RunPriveged | Eseguire il container telegraf in modalità privilegiata. Impostare questa opzione su true se SELinux è attivato sui nodi k8s |
| | 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" |
| | DockerMetricCollectionEnabled | Raccogli le metriche di Docker. Per impostazione predefinita, questa opzione è impostata su true e le metriche del docker verranno raccolte per le implementazioni k8s on-premise e basate su docker. Per disattivare la raccolta di metriche docker, impostarla su false. |
| | DsTolerazioni | teletegraf-ds tollerazioni aggiuntive. |
| | RsTolerazioni | tollerazioni aggiuntive di telegraf-rs. |
| 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 |
| | 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 |

| Componente | Opzione | Descrizione |
| --- | --- | --- |
| | risorse | un elenco separato da virgole di risorse da acquisire. esempio: cronjobs,demonset,implementazioni,inserimenti,job,namespaces,nodi,persistentvolumeclaims, persistentvolumes,pod,replicasets,resourcequotas,servizi,statefulsets |
| | tollerazioni | tolleranze aggiuntive delle metriche dello stato del kube. |
| | etichette | un elenco separato da virgole di risorse che kube-state-metrics dovrebbe acquisire<br><br>esempio: cronjobs=[*],demonsets=[*],deployments=[*],ingresses=[*],jobs=[*],namespaces=[*],nodes=[*], persistentvolumeclaims=[*],persistentvolumes=[*],pods=[*],replicasets=[*],resourcequotas=[*],services=[*],statefulsets=[*] |
| 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. |
| mappa del carico di lavoro | | Opzioni di configurazione che consentono di personalizzare la raccolta e l'installazione della mappa del carico di lavoro 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 |
| | l4-tollerazioni | tolleranza aggiuntiva net-observer-l4-ds. |
| | RunPriveged | Vero/falso - impostare runPrivileged su true se SELinux è abilitato sui tuoi nodi Kubernetes. |

| Componente | Opzione | Descrizione |
|---|---|---|
| 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 | Insieme di espressioni regolari utilizzate nei nomi env e nelle mappe di dati il cui valore verrà rivisto 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 esempio di file AgentConfiguration.

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

```
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 Cloud
Insights environment.
    clusterName: "CLUSTERNAME_PLACEHOLDER"

    # # Proxy settings. The proxy that the operator should use to send
metrics to Cloud 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_REPO_PLACEHOLDER'
    # # [Required Field] The name of the imagePullSecret for dockerRepo.
    # # If you are using a private repository, change this field from
'docker' to the name of your secret.
    {{ if not (contains .Values.config.cloudType "aws") }}# {{ end -}}
    dockerImagePullSecret: 'docker'

    # # Allow the operator to automatically rotate its ApiKey before
expiration.
    # tokenRotationEnabled: '{{
.Values.telegraf_installer.kubernetes.rs.shim_token_rotation  }}'
    # # Number of days before expiration that the ApiKey should be
```

```
rotated. This must be less than the total ApiKey duration.
    # tokenRotationThresholdDays: '{{
.Values.telegraf_installer.kubernetes.rs.shim_token_rotation_threshold_day
s  }}'

  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: '{{
.Values.telegraf_installer.agent_resources.collection_interval }}'
    # # Maximum number of records per output that telegraf will write in
one batch (metric_batch_size).
    # batchSize: '{{
.Values.telegraf_installer.agent_resources.metric_batch_size }}'
    # # Maximum number of records per output that telegraf will cache
pending a successful write (metric_buffer_limit).
    # bufferLimit: '{{
.Values.telegraf_installer.agent_resources.metric_buffer_limit }}'
    # # Collect metrics on multiples of interval (round_interval).
    # roundInterval: '{{
.Values.telegraf_installer.agent_resources.round_interval }}'
    # # Each plugin waits a random amount of time between the scheduled
collection time and that time + collection_jitter before collecting inputs
(collection_jitter).
    # collectionJitter: '{{
.Values.telegraf_installer.agent_resources.collection_jitter }}'
    # # Collected metrics are rounded to the precision specified. When set
to "0s" precision will be set by the units specified by interval
(precision).
    # precision: '{{ .Values.telegraf_installer.agent_resources.precision
}}'
    # # Time telegraf will wait between writing outputs (flush_interval).
Max=collectionInterval
    # flushInterval: '{{
.Values.telegraf_installer.agent_resources.flush_interval }}'
    # # Each output waits a random amount of time between the scheduled
write time and that time + flush_jitter before writing outputs
(flush_jitter).
    # flushJitter: '{{
.Values.telegraf_installer.agent_resources.flush_jitter }}'
```

```
    # # Timeout for writing to outputs (timeout).
    # outputTimeout: '{{
.Values.telegraf_installer.http_output_plugin.timeout }}'

    # # telegraf-ds CPU/Mem limits and requests.
    # # See https://kubernetes.io/docs/concepts/configuration/manage-
resources-containers/
    dsCpuLimit: '{{
.Values.telegraf_installer.telegraf_resources.ds_cpu_limits  }}'
    dsMemLimit: '{{
.Values.telegraf_installer.telegraf_resources.ds_mem_limits  }}'
    dsCpuRequest: '{{
.Values.telegraf_installer.telegraf_resources.ds_cpu_request  }}'
    dsMemRequest: '{{
.Values.telegraf_installer.telegraf_resources.ds_mem_request  }}'

    # # telegraf-rs CPU/Mem limits and requests.
    rsCpuLimit: '{{
.Values.telegraf_installer.telegraf_resources.rs_cpu_limits  }}'
    rsMemLimit: '{{
.Values.telegraf_installer.telegraf_resources.rs_mem_limits  }}'
    rsCpuRequest: '{{
.Values.telegraf_installer.telegraf_resources.rs_cpu_request  }}'
    rsMemRequest: '{{
.Values.telegraf_installer.telegraf_resources.rs_mem_request  }}'

    # # 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: ''

    # # Set runPrivileged to true if SELinux is enabled on your Kubernetes
nodes.
    # runPrivileged: 'false'

    # # Collect NFS IO metrics.
    # dsNfsIOEnabled: '{{
.Values.telegraf_installer.kubernetes.ds.shim_nfs_io_processing }}'

    # # 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: '{{
.Values.telegraf_installer.kubernetes.shim_managed_k8s_system_metric_colle
ction }}'

    # # Collect kubernetes.pod_volume (pod ephemeral storage) metrics.
Set this to true if you want to collect these metrics.
    # podVolumeMetricCollectionEnabled: '{{
.Values.telegraf_installer.kubernetes.shim_pod_volume_metric_collection
}}'

    # # Declare Rancher cluster as managed.  Set this to true if your
Rancher cluster is managed as opposed to on-premise.
    # isManagedRancher: '{{
.Values.telegraf_installer.kubernetes.is_managed_rancher }}'

  # kube-state-metrics:
    # # kube-state-metrics CPU/Mem limits and requests. By default, when
unset, kube-state-metrics has no CPU/Mem limits nor request.
    # cpuLimit:
    # memLimit:
    # cpuRequest:
    # memRequest:

    # # Comma-separated list of metrics to enable.
    # # See metric-allowlist 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 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}'
```

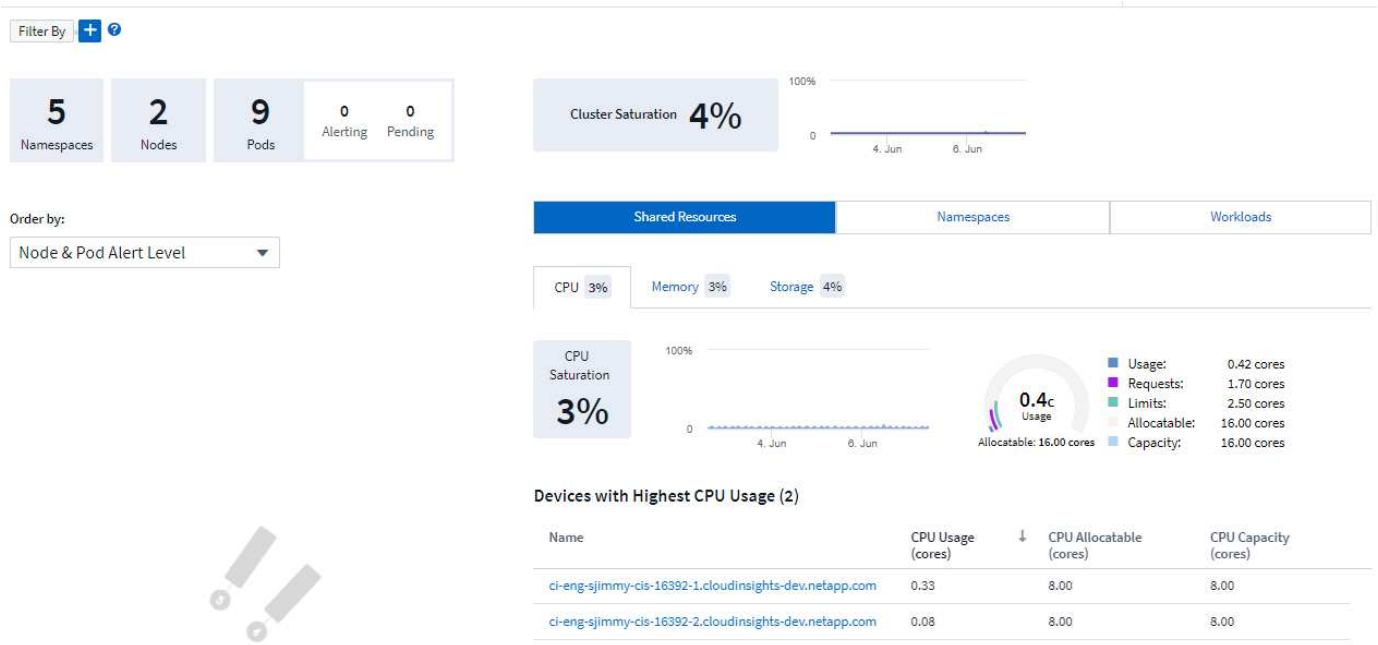
```
    # tolerations: ''

  # # Settings for the Events Log feature.
  # logs:
    # # 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"

    # # 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: ''

  # # Settings for the Network Performance and Map feature.
  # workload-map:
    # # 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'

    # # net-observer-l4-ds additional tolerations. Use the following
abbreviated single line format only.
    # # Inspect 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: '500m'
    # memLimit: '500Mi'
    # cpuRequest: '100m'
    # 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",
"jwt"'

    # # 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 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'
    # 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.

Shared Resources | **Namespaces** | Workloads

**Namespaces (5)**

| Name ↓ | CPU Usage (cores) | Memory Usage (GiB) | Workload Count |
|---|---|---|---|
| netapp-monitoring | 0.25 | 0.38 | 4 |
| kube-system | 0.01 | 0.03 | 3 |
| kube-public | 0.00 | 0.00 | 0 |
| kube-node-lease | 0.00 | 0.00 | 0 |
| default | 0.00 | <0.01 | 1 |

## 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.

Shared Resources | Namespaces | **Workloads**

**Workloads (8)**

| Name ↓ | CPU Usage (cores) | Memory Usage (GiB) | Namespace |
|---|---|---|---|
| telegraf-rs-lf9gg | 0.24 | 0.24 | netapp-monitoring |
| telegraf-ds-k957c | 0.01 | 0.10 | netapp-monitoring |
| nginx | 0.00 | <0.01 | default |
| monitoring-operator-6fcf4755ff-p2cs6 | <0.01 | 0.02 | netapp-monitoring |
| metrics-server-7b4f8b595-f7j9f | <0.01 | 0.01 | kube-system |
| local-path-provisioner-64d457c485-289gx | <0.01 | 0.01 | kube-system |
| kube-state-metrics-7995866f8c-t8c49 | <0.01 | 0.01 | netapp-monitoring |
| coredns-5d69dc75db-nkw5p | <0.01 | 0.01 | kube-system |

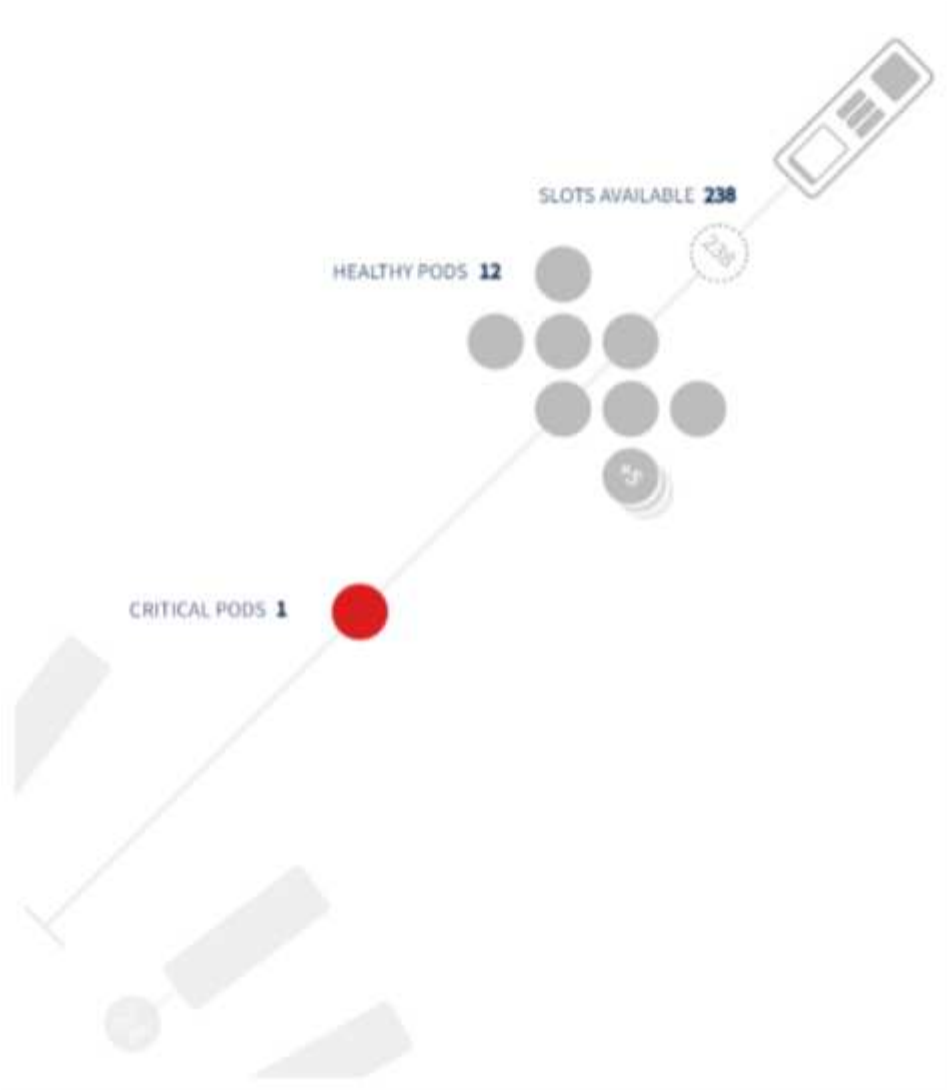
## La "ruota" del cluster


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.

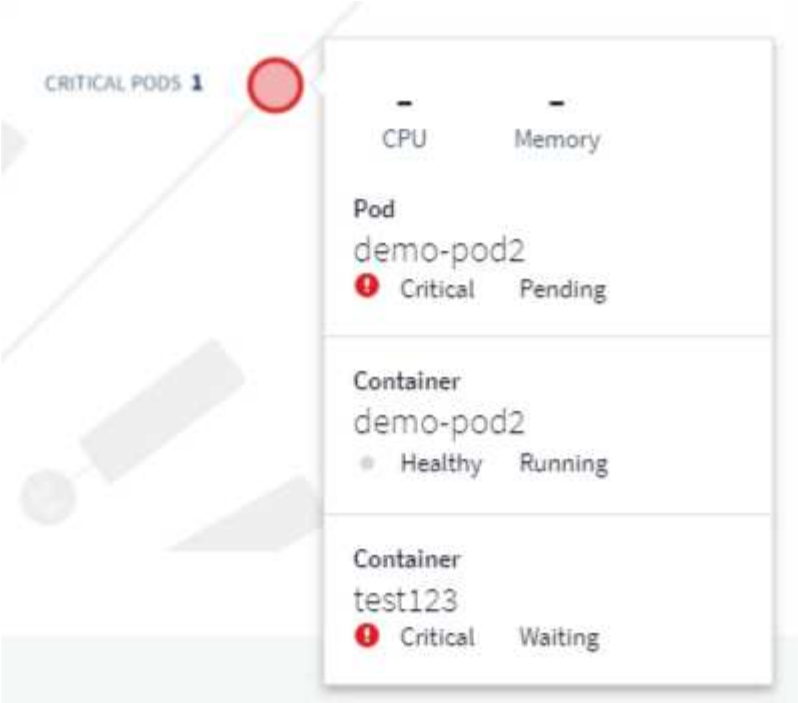
I pod o i nodi di avviso vengono visualizzati in rosso. Le aree di "avvertenza" sono visualizzate in arancione. I pod non pianificati (ovvero non collegati) vengono visualizzati nell'angolo inferiore della "ruota" del cluster.

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*.

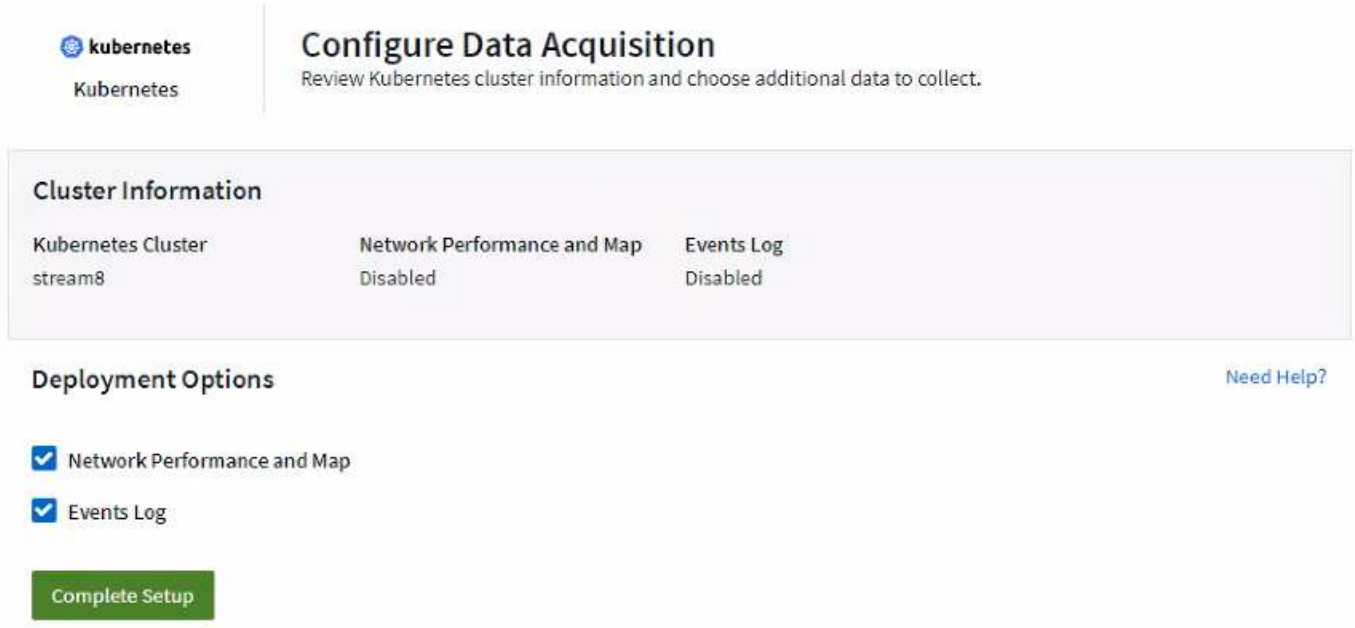
# Kubernetes Network Performance Monitoring and Map

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 Kubernetes Network Performance Monitoring and Map, è necessario aver configurato "NetApp Kubernetes Monitoring Operator" per attivare 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

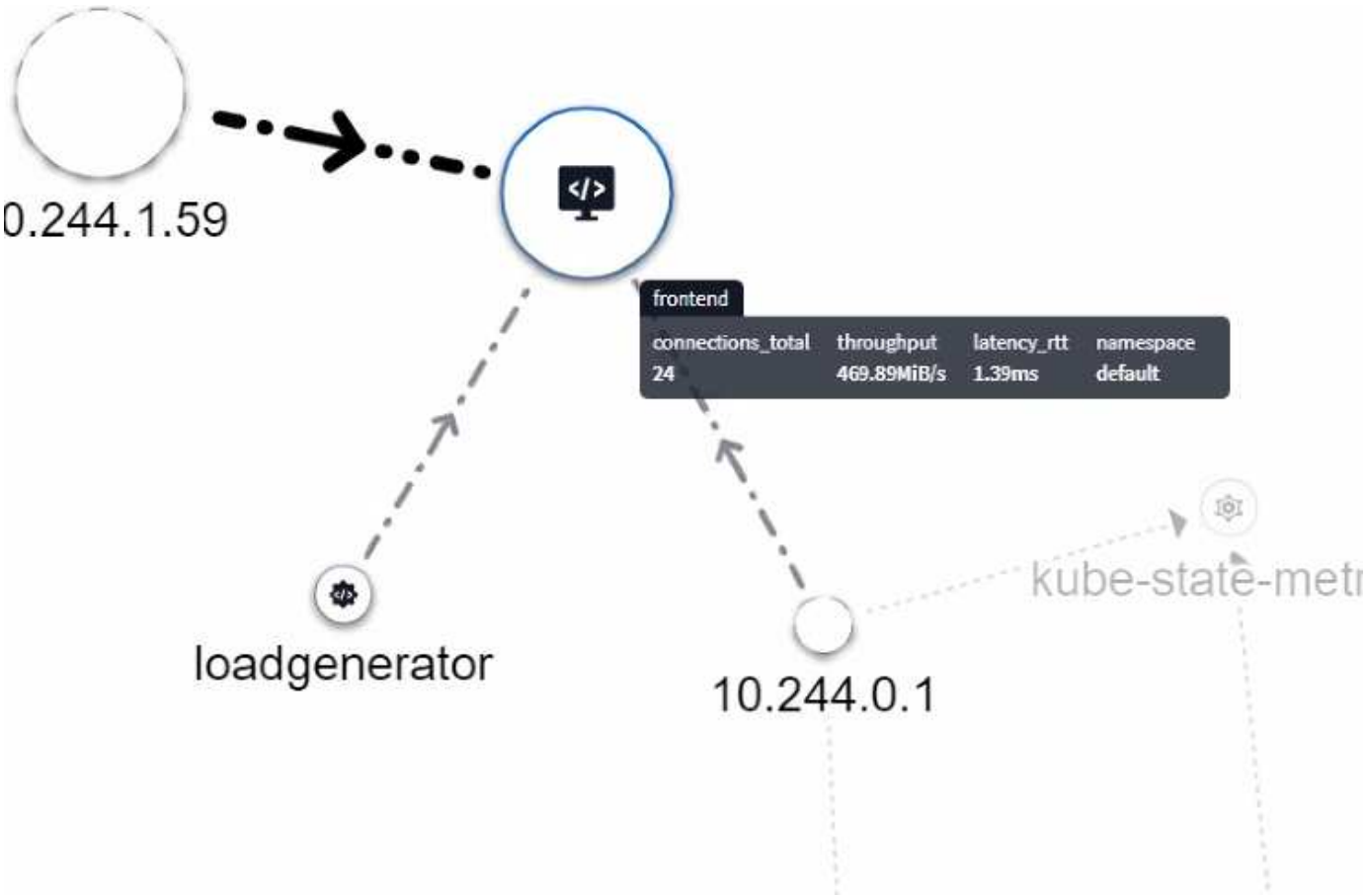
La mappa del carico di lavoro utilizza "monitor" per ricavare informazioni. Cloud Insights fornisce una serie di monitor Kubernetes predefiniti (si noti che per impostazione predefinita potrebbero essere *in pausa*). È possibile *riprendere* (ad esempio attivare) i monitor desiderati) oppure creare monitor personalizzati per gli oggetti Kubernetes, che verranno utilizzati anche dalla mappa del carico di lavoro.

È possibile creare avvisi Cloud Insights metric su uno qualsiasi dei tipi di oggetto riportati di seguito. Assicurarsi che i dati siano raggruppati in base al tipo di oggetto predefinito.

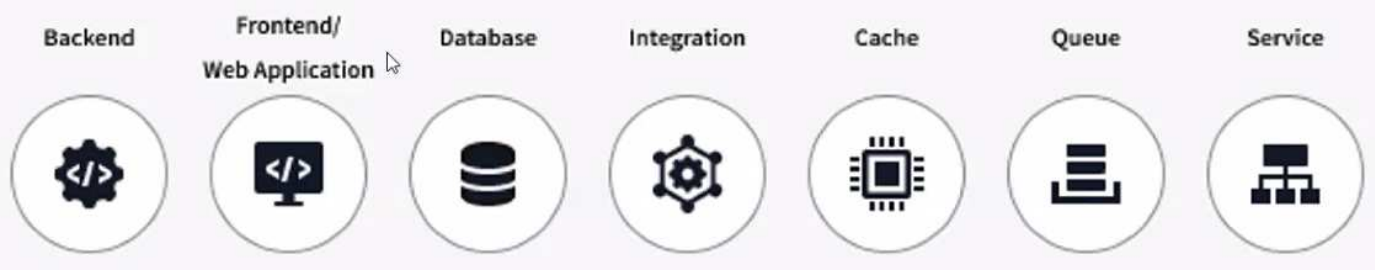
- kubernetes.workload
- kubernetes.demonset
- kubernetes.deployment
- kubernetes.cronjob
- kubernetes.job
- kubernetes.replicaset
- kubernetes.statefulset
- kubernetes.pod
- kubernetes.network_traffic_l4

## 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:

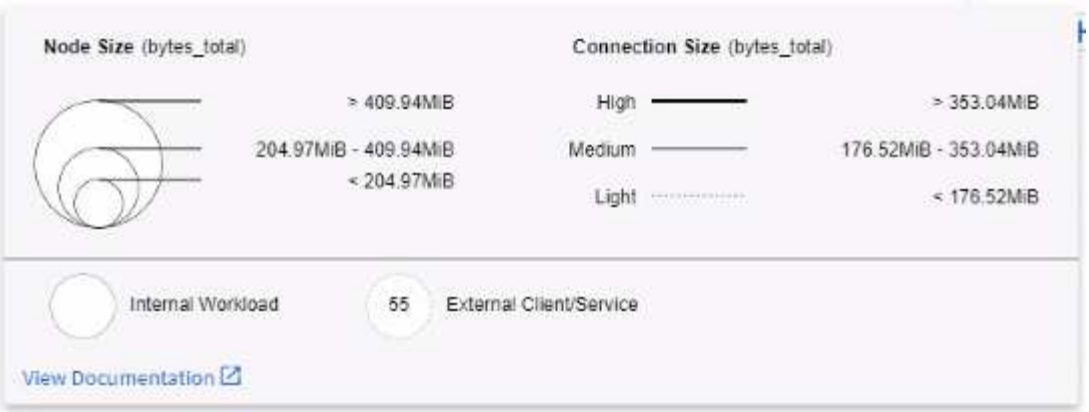



Le icone all'interno dei cerchi rappresentano diversi tipi di servizio. Si noti che le icone sono visibili solo se sono presenti gli oggetti sottostanti etichette.



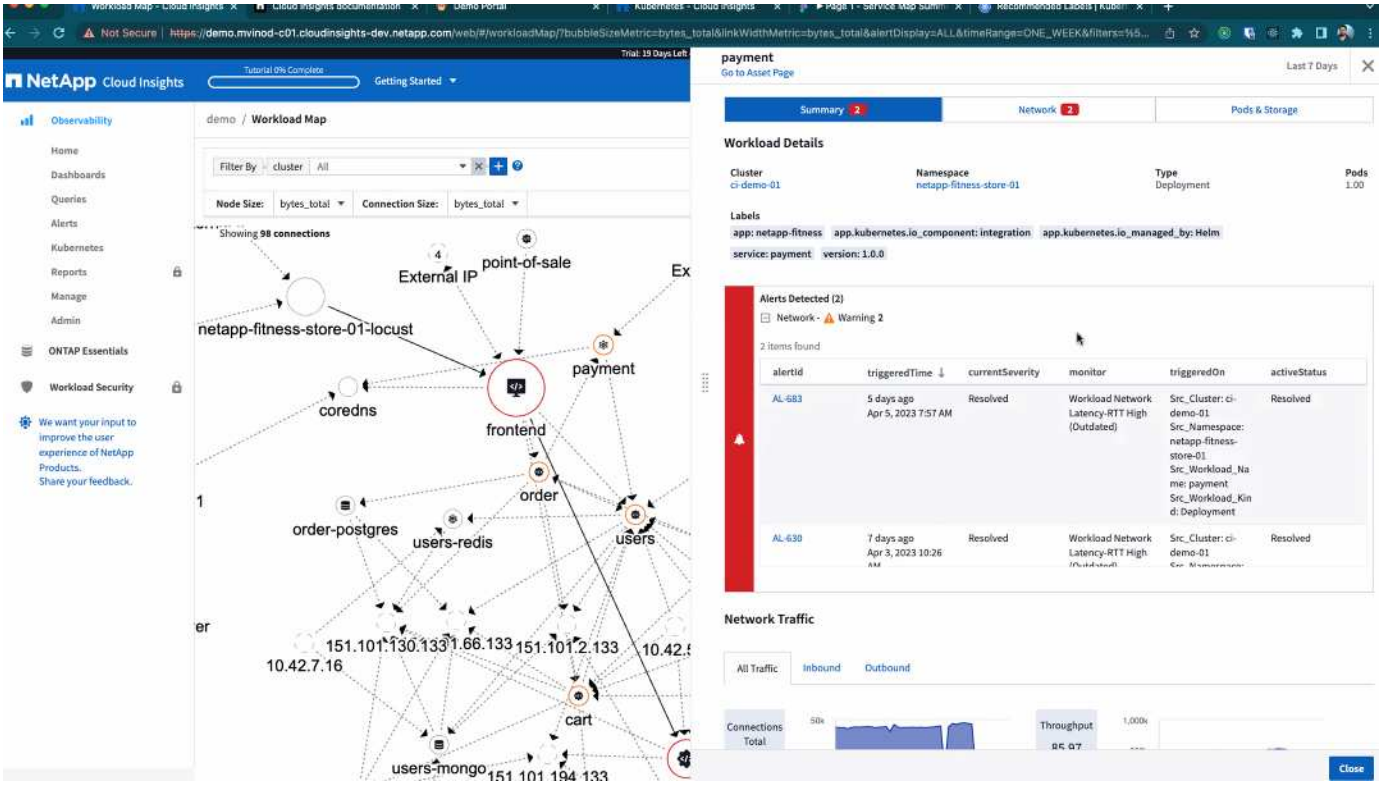

La dimensione di ciascun cerchio indica la dimensione del nodo. Si noti che queste dimensioni sono relative, il livello di zoom del browser o le dimensioni dello schermo potrebbero influire sulle dimensioni effettive dei cerchi. Allo stesso modo, lo stile della linea di traffico offre una vista a colpo 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.



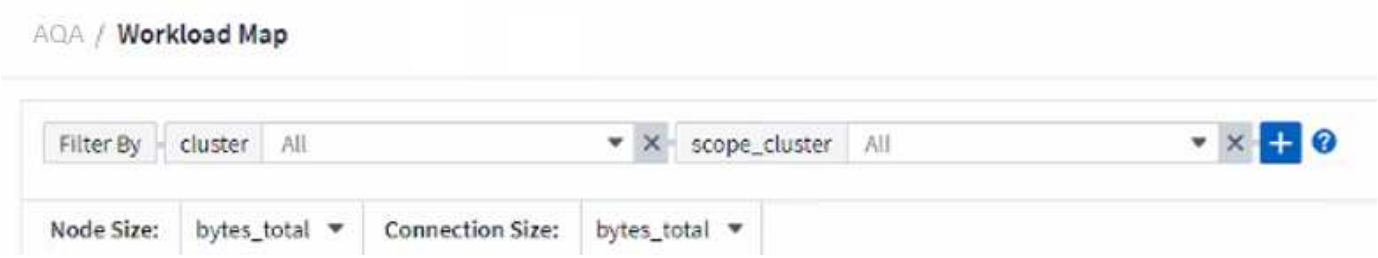

## Avvisi e dettagli sul carico di lavoro

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.

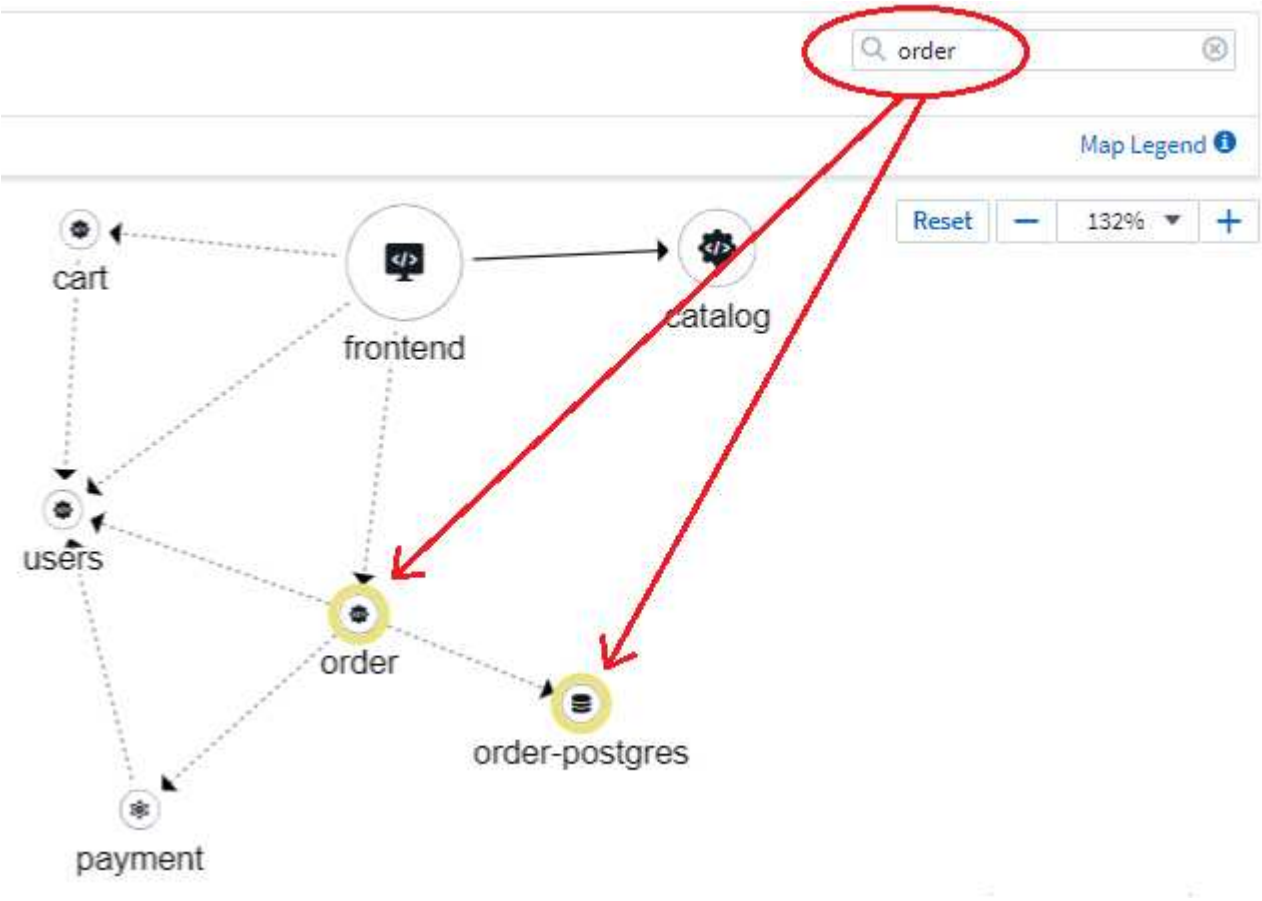



## Ricerca e filtraggio

Come per le altre funzionalità di Cloud Insights, è possibile impostare facilmente i filtri in modo che si concentrino sugli oggetti o sugli attributi dei carichi di lavoro specifici desiderati.




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

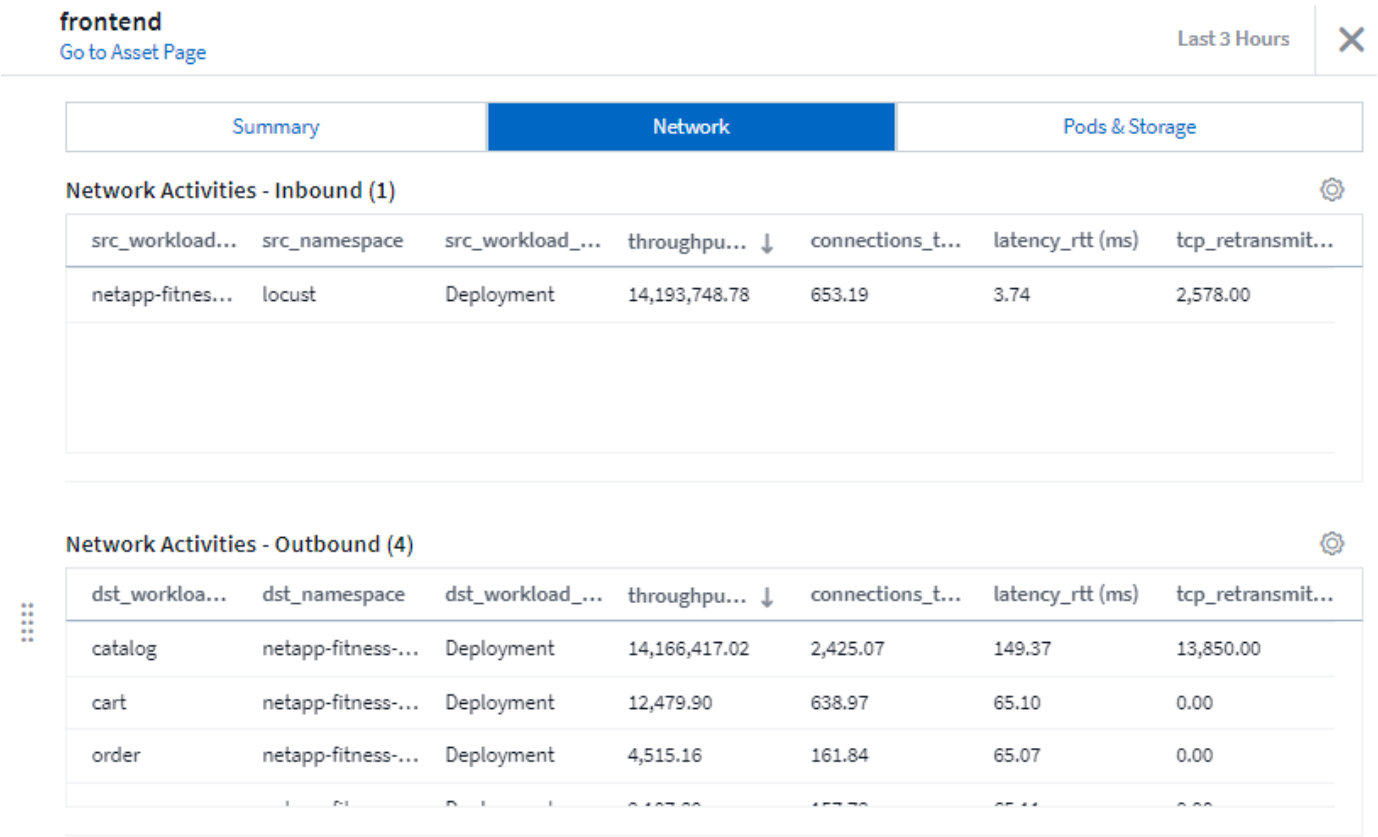
    - back-end
    - database
    - cache
    - coda
    - kafka
- Se l'origine è esterna al cluster kubernetes:
  - Cloud Insights tenterà di analizzare il nome DNS risolto per estrarre il tipo di servizio.

    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.

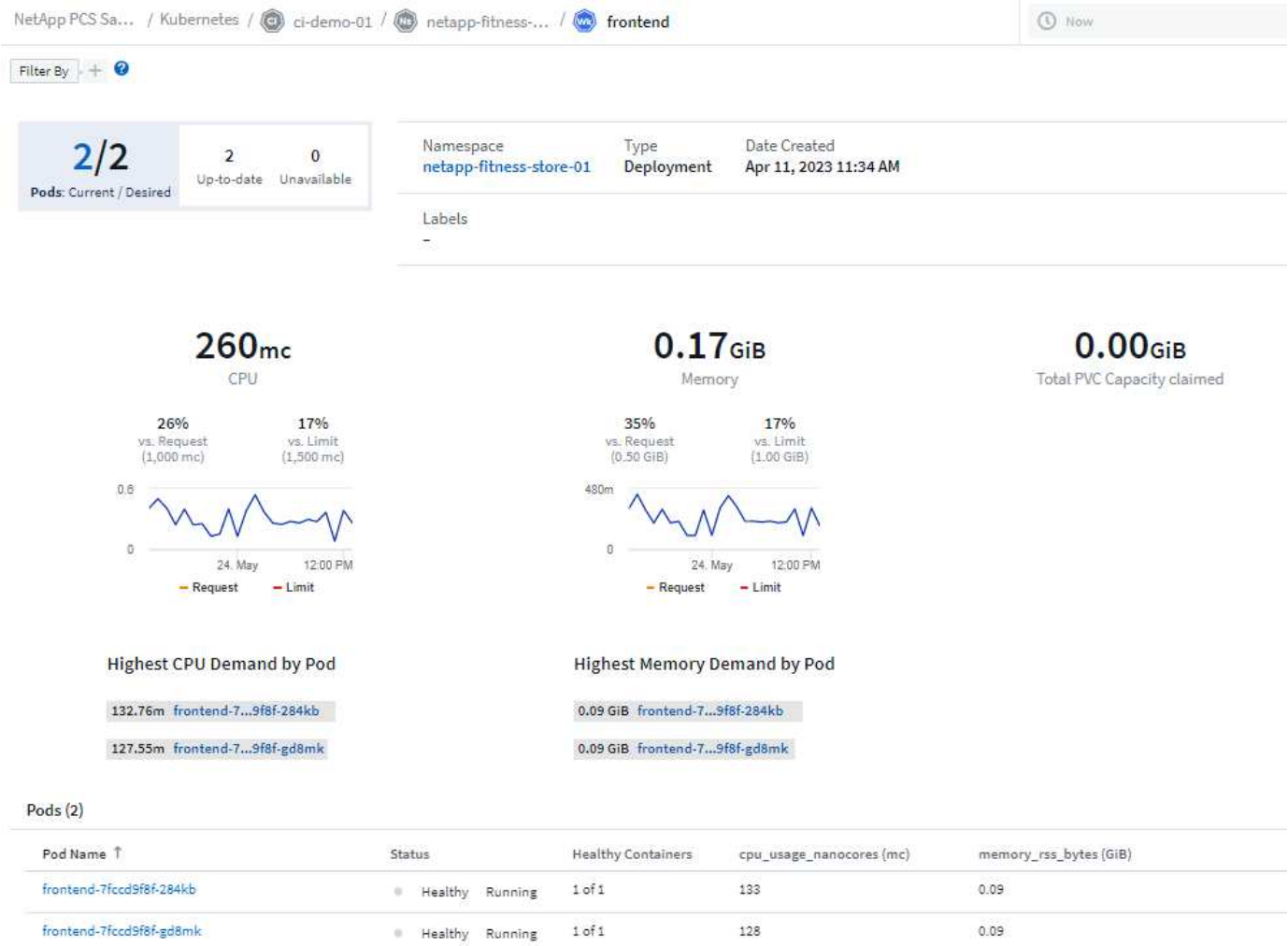
## Tuffati in profondità

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.



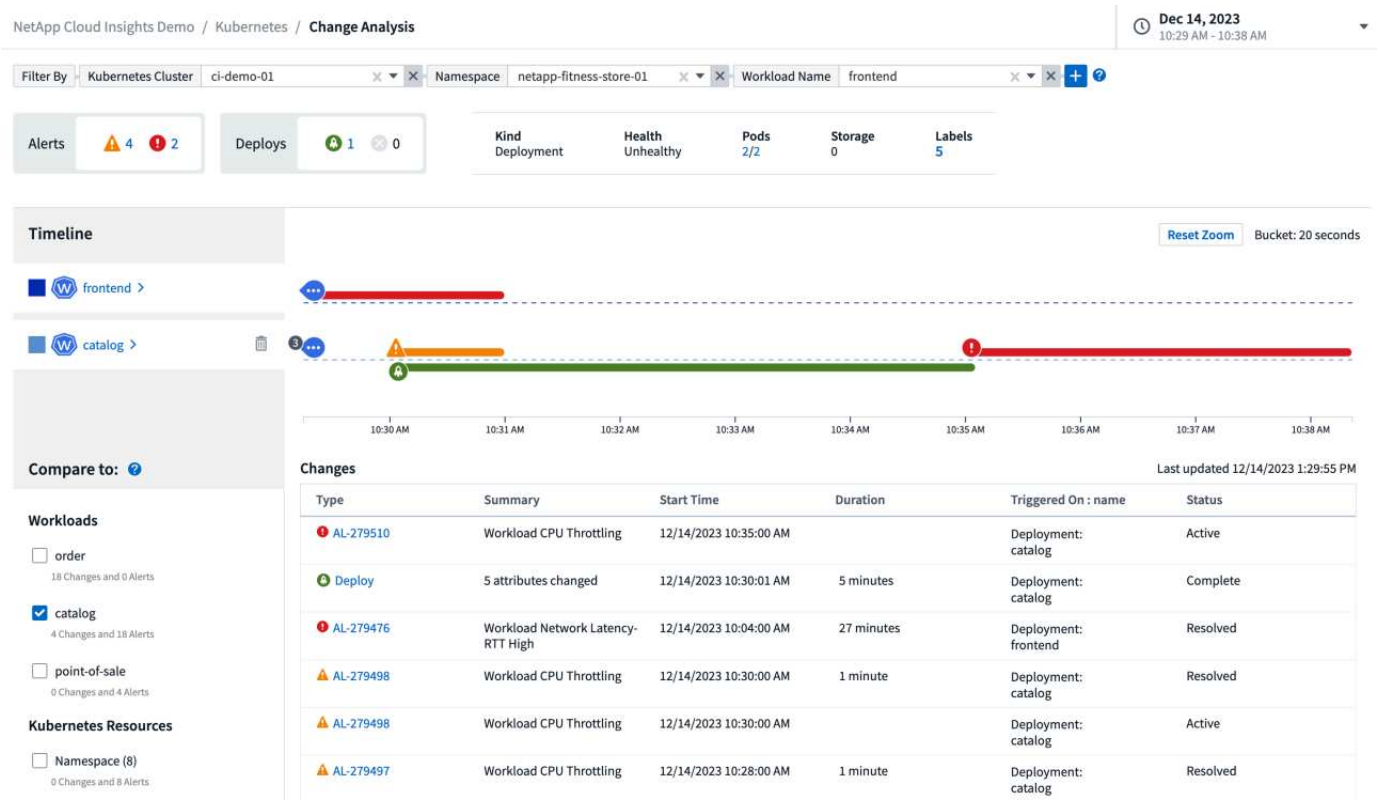

# Analytics delle modifiche di Kubernetes

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.

Tenere presente quanto segue:

- Negli ambienti multi-tenant, è possibile che si verifichino interruzioni a causa di modifiche non configurate correttamente. In ambienti molto dinamici, l'analisi delle modifiche Cloud Insights potrebbe non essere in grado di tenere traccia correttamente di tutte le modifiche.
- Change Analytics offre un singolo riquadro per visualizzare e correlare lo stato dei carichi di lavoro e le modifiche alla configurazione. Questo può essere utile nella risoluzione dei problemi degli ambienti dinamici.

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



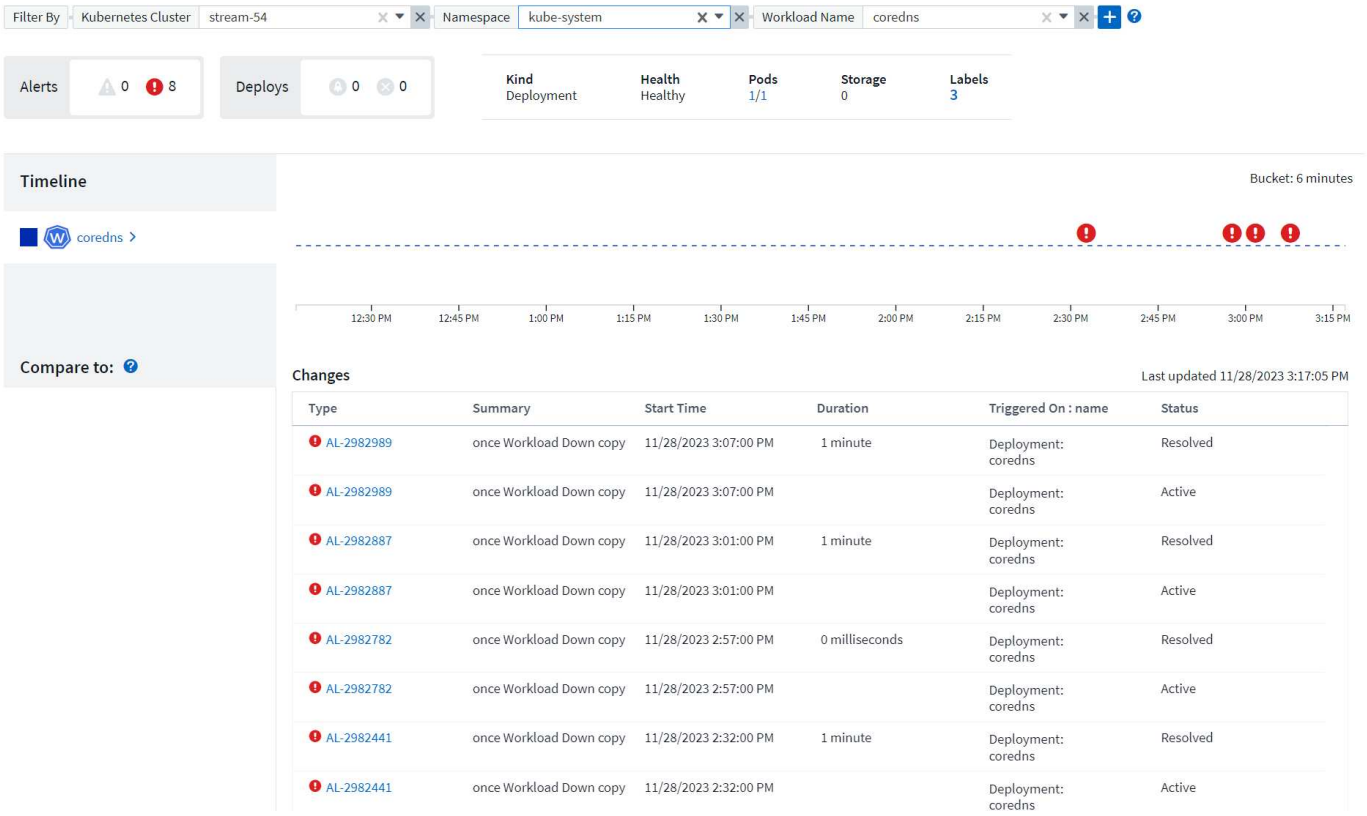

.

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

## Filtraggio

Come per tutte le funzionalità di Cloud Insights, filtrare l'elenco di modifiche è intuitivo: Nella parte superiore della pagina, immettere o selezionare i valori per il cluster Kubernetes, lo spazio dei nomi o il workload 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.

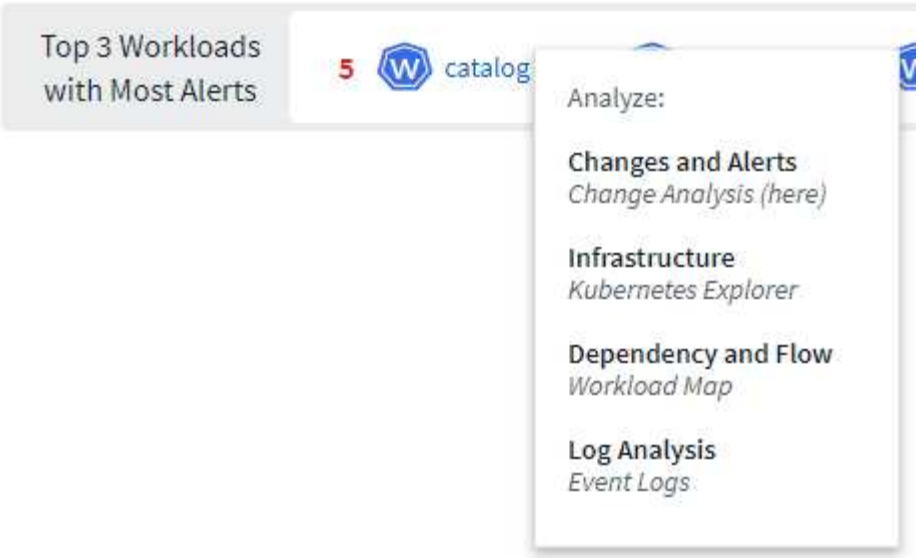



## Stato rapido

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.

## Deploy Failed

### Summary

**Start Time**
10/18/2023 2:40:01 PM

**End Time**
10/18/2023 2:50:02 PM

**Duration**
10 minutes

**Triggered On**
ci-demo-01 >
netapp-fitness-store-01 >
billing-accounts >

**Triggered On : kind**
Deployment

### Failure Detail

**Reason For Failure**
ProgressDeadlineExceeded - ReplicaSet "billing-accounts-6ddc7df546" has timed out progressing.

**Message**
Failed deploy

### Changes (2)

| Attribute Name | Previous | New |
|---|---|---|
| spec.template.spec.containers[0].image | 210811600188.dkr.ecr.us-east-1.amazonaws.com/sm-billing-accounts-apis:1.0.0 | 210811600188.dkr.ecr.us-east-1.amazonaws.com/sm-billing-accounts-apis:1.0.09 |
| metadata.annotations.deployment.kubernetes.io/revision | 2964 | 2965 |

All Changes Diff

### Associated Events

Event Logs

Close

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.