



Kubernetes

Cloud Insights

NetApp
September 13, 2024

This PDF was generated from https://docs.netapp.com/de-de/cloudinsights/kubernetes_landing_page.html on September 13, 2024. Always check docs.netapp.com for the latest.

Inhalt

- Kubernetes 1
 - Kubernetes-Cluster – Übersicht 1
 - Bevor Sie den NetApp Kubernetes Monitoring Operator installieren oder aktualisieren 2
 - Installation und Konfiguration des Kubernetes Monitoring Operator 6
 - Konfigurationsoptionen Für Kubernetes Monitoring Operator 26
 - Detailseite Zu Kubernetes Cluster 36
 - Performance-Monitoring und -Zuordnung des Kubernetes-Netzwerks 41
 - Kubernetes Change Analytics 49

Kubernetes

Kubernetes-Cluster – Übersicht

Der Cloud Insights Kubernetes Explorer ist ein leistungsstarkes Tool zur Anzeige des Gesamtzustands und der Verwendung der Kubernetes Cluster und ermöglicht es Ihnen, mühelos detaillierte Informationen zu den Bereichen der Untersuchung anzuzeigen.

Durch Klicken auf **Dashboards > Kubernetes Explorer** wird die Listenseite für Kubernetes-Cluster geöffnet. Diese Übersichtsseite enthält Tabellen der Kubernetes Cluster in Ihrer Umgebung.

Filter By  

Clusters (2)

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




Cluster-Liste

In der Cluster-Liste werden für jedes Cluster in Ihrer Umgebung die folgenden Informationen angezeigt:

- Cluster **Name**. Wenn Sie auf einen Cluster-Namen klicken, wird das geöffnet "**Detailseite**" Für diesen Cluster zu erstellen.
- **Sättigung** Prozentsätze. „Gesamteinlagerung“ entspricht dem höchsten Wert für CPU, Speicher oder Speichersättigung.
- Anzahl **Nodes** im Cluster. Wenn Sie auf diese Nummer klicken, wird die Seite Knotenliste geöffnet.
- Anzahl **Pods** im Cluster. Wenn Sie auf diese Nummer klicken, wird die Pod-Listenseite geöffnet.
- Anzahl **Namespaces** im Cluster. Wenn Sie auf diese Nummer klicken, wird die Namespace-Listenseite geöffnet.
- Anzahl **Workloads** im Cluster. Wenn Sie auf diese Nummer klicken, wird die Listenseite Workload geöffnet.

Verfeinern des Filters

Wenn Sie filtern, werden Sie beim Eingeben mit der Option angezeigt, basierend auf dem aktuellen Text einen **Platzhalterfilter** zu erstellen. Wenn Sie diese Option auswählen, werden alle Ergebnisse angezeigt, die dem Platzhalterausdruck entsprechen. Sie können auch **Expressions** mit NOT oder UND erstellen, oder Sie können die Option "Keine" auswählen, um nach Null-Werten im Feld zu filtern.

Filter By namespace kube   

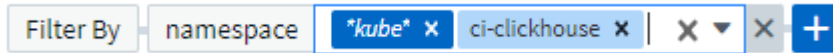
Create wildcard containing "kube"

kube-public

kube-system

None

Filter basierend auf Platzhalter oder Ausdrücken (z. B. NOT, AND, „None“ etc.) wird im Filterfeld dunkelblau angezeigt. Elemente, die Sie direkt aus der Liste auswählen, werden hellblau angezeigt.



Kubernetes-Filter sind kontextbezogen, d. h., wenn Sie sich beispielsweise auf einer bestimmten Knotenseite befinden, listet der Pod_Name-Filter nur die Pods auf, die mit diesem Node zusammenhängen. Wenn Sie darüber hinaus einen Filter für einen bestimmten Namespace anwenden, werden im Pod_Name-Filter nur Pods auf diesem Node *und* in diesem Namespace aufgelistet.

Beachten Sie, dass die Platzhalter- und Ausdrucksfilterung mit Text oder Listen funktioniert, jedoch nicht mit numerischen Werten, Daten oder Booleanen.

Bevor Sie den NetApp Kubernetes Monitoring Operator installieren oder aktualisieren

Lesen Sie diese Informationen, bevor Sie das installieren oder aktualisieren "[Kubernetes Monitoring Operator](#)".

Komponente	Anforderungen
Kubernetes-Version	Kubernetes v1.20 und höher
Kubernetes Distributionen	AWS Elastic Kubernetes Service (EKS) Azure Kubernetes-Service (AKS) Google Kubernetes Engine (GKE) Red hat OpenShift Rancher Kubernetes Engine (RKE) VMware Tanzu
Linux BS	Cloud Insights unterstützt keine Knoten, die mit Arm64-Architektur laufen. Netzwerküberwachung: Muss Linux Kernel Version 4.18.0 oder höher ausführen. Photon OS wird nicht unterstützt.
Etiketten	Cloud Insights unterstützt das Monitoring von Kubernetes-Nodes, auf denen Linux ausgeführt wird, indem eine Kubernetes-Node-Auswahl angegeben wird, die auf diesen Plattformen die folgenden Kubernetes-Labels berücksichtigt: Kubernetes v1.20 und höher: Kubernetes.io/os = linux Rancher + Cattle.io als Orchestrierungs-/Kubernetes-Plattform: Cattle.io/os = linux
Befehle	Die Befehle Curl und kubectl müssen verfügbar sein.; für optimale Ergebnisse fügen Sie diese Befehle dem PFAD hinzu.

Komponente	Anforderungen
Konnektivität	<p>Kubectl cli ist für die Kommunikation mit dem Ziel-K8s-Cluster konfiguriert und verfügt über eine Internetverbindung zu Ihrer Cloud Insights-Umgebung.</p> <p>Wenn Sie während der Installation hinter einem Proxy stehen, befolgen Sie die Anweisungen im "Proxy-Unterstützung Wird Konfiguriert" Abschnitt der Bedienerinstallation.</p> <p>Für genaue Audit- und Datenberichte synchronisieren Sie die Zeit auf dem Agent-Computer mit Network Time Protocol (NTP) oder Simple Network Time Protocol (SNTP).</p>
Andere	<p>Wenn Sie OpenShift 4.6 oder höher verwenden, müssen Sie die folgenden Schritte ausführen "OpenShift-Anweisungen" Zusätzlich zur Sicherstellung, dass diese Voraussetzungen erfüllt sind.</p>
API-Token	<p>Wenn Sie den Operator neu bereitstellen (d. h. aktualisieren oder ersetzen), müssen Sie kein neues API-Token erstellen; Sie können das vorherige Token erneut verwenden.</p>

Wichtige Dinge, die Sie beachten sollten, bevor Sie beginnen

Wenn Sie mit einem laufen [Proxy](#), Haben Sie eine [Benutzerdefiniertes Repository](#), Oder verwenden [OpenShift](#), Lesen Sie die folgenden Abschnitte sorgfältig.

Lesen Sie auch darüber [Berechtigungen](#).

Proxy-Unterstützung Wird Konfiguriert

An zwei Stellen können Sie in Ihrer Umgebung einen Proxy verwenden, um den NetApp Kubernetes Monitoring Operator zu installieren. Es kann sich um dieselben oder separate Proxy-Systeme handelt:

- Proxy benötigt bei Ausführung des Installationscodes Snippet (mit "Curl"), um das System, an dem das Snippet ausgeführt wird, mit Ihrer Cloud Insights-Umgebung zu verbinden
- Proxy für die Kommunikation mit Ihrer Cloud Insights Umgebung durch das Ziel-Kubernetes-Cluster

Wenn Sie einen Proxy für eine oder beide dieser Optionen verwenden, müssen Sie zuerst sicherstellen, dass Ihr Proxy für eine gute Kommunikation mit Ihrer Cloud Insights-Umgebung konfiguriert ist, um den NetApp Kubernetes-Betriebsmonitor zu installieren. Beispielsweise müssen Sie von den Servern/VMs, von denen Sie den Operator installieren möchten, auf Cloud Insights zugreifen und Binärdateien von Cloud Insights herunterladen können.

Legen Sie für den Proxy, der zur Installation des NetApp Kubernetes Operating Monitor verwendet wurde, vor der Installation des Operators die Umgebungsvariablen `http_Proxy/https_Proxy` fest. In einigen Proxy-Umgebungen müssen Sie möglicherweise auch die Variable `no_Proxy Environment` festlegen.

Um die Variable(en) festzulegen, führen Sie auf Ihrem System **vor** der Installation des NetApp Kubernetes

Monitoring Operators folgende Schritte aus:

1. Legen Sie die Umgebungsvariable `https_Proxy` und/oder `http_Proxy` für den aktuellen Benutzer fest:
 - a. Wenn der Proxy, der eingerichtet wird, keine Authentifizierung (Benutzername/Passwort) aufweist, führen Sie den folgenden Befehl aus:

```
export https_proxy=<proxy_server>:<proxy_port>
.. Wenn der Proxy, der eingerichtet wird, über Authentifizierung
(Benutzername/Passwort) verfügt, führen Sie folgenden Befehl aus:
```

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

Nachdem Sie alle diese Anweisungen gelesen haben, installieren Sie den Proxy, der für die Kommunikation Ihres Kubernetes Clusters mit Ihrer Cloud Insights-Umgebung verwendet wurde.

Konfigurieren Sie den Proxy-Abschnitt von AgentConfiguration in Operator-config.yaml, bevor Sie den NetApp Kubernetes Monitoring Operator bereitstellen.

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

Verwenden eines benutzerdefinierten oder privaten Docker Repositorys

Standardmäßig sendet der NetApp Kubernetes Monitoring Operator Container-Images aus dem Cloud

Insights-Repository. Wenn Sie ein Kubernetes-Cluster als Ziel für das Monitoring verwenden und der Cluster so konfiguriert ist, dass er nur Container-Images aus einem benutzerdefinierten oder privaten Docker-Repository oder der Container-Registrierung zieht, müssen Sie den Zugriff auf die Container konfigurieren, die vom NetApp Kubernetes Monitoring Operator benötigt werden.

Führen Sie das „Image Pull Snippet“ aus der NetApp Monitoring Operator Installationskachel aus. Dieser Befehl meldet sich beim Cloud Insights-Repository an, zieht alle Image-Abhängigkeiten für den Operator und meldet sich vom Cloud Insights-Repository ab. Wenn Sie dazu aufgefordert werden, geben Sie das angegebene temporäre Repository-Passwort ein. Mit diesem Befehl werden alle vom Bediener verwendeten Bilder heruntergeladen, einschließlich optionaler Funktionen. Nachfolgend sehen Sie, für welche Funktionen diese Bilder verwendet werden.

Core Operator-Funktionalität und Kubernetes Monitoring

- netapp Monitoring
- kube-rbac-Proxy
- status-Kennzahlen von kube
- telegraf
- Distroless-root-user

Ereignisprotokoll

- Fluent-Bit
- kubernetes Event Exporter

Netzwerkleistung und -Zuordnung

- ci-Netz-Beobachter

Übertragen Sie das Operator-Docker-Image gemäß Ihren Unternehmensrichtlinien in das private/lokale/unternehmenseigene Docker-Repository. Stellen Sie sicher, dass die Bild-Tags und Verzeichnispfade zu diesen Bildern in Ihrem Repository mit denen im Cloud Insights-Repository übereinstimmen.

Bearbeiten Sie die Bereitstellung des Monitoring-Operators in Operator-Deployment.yaml, und ändern Sie alle Bildverweise, um Ihr privates Docker-Repository zu verwenden.

```
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>
```

Bearbeiten Sie die AgentConfiguration in Operator-config.yaml, um die neue Position des Docker-Repo zu berücksichtigen. Erstellen Sie ein neues imagePullSecret für Ihr privates Repository. Weitere Informationen finden Sie unter <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
```

OpenShift-Anweisungen

Wenn Sie OpenShift 4.6 oder höher ausführen, müssen Sie die AgentConfiguration in *Operator-config.yaml* bearbeiten, um die Einstellung *runPrivileged* zu aktivieren:

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

OpenShift kann zusätzliche Sicherheitsstufen implementieren, die den Zugriff auf einige Kubernetes-Komponenten blockieren könnten.

Berechtigungen

Wenn das zu überwachende Cluster benutzerdefinierte Ressourcen enthält, für die keine ClusterRole vorhanden ist "[AnzeigeEinblick in Aggregate](#)" Sie müssen dem Bediener manuell Zugriff auf diese Ressourcen gewähren, um sie mit Ereignisprotokollen zu überwachen.

1. Bearbeiten Sie *Operator-additional-permissions.yaml* vor der Installation oder nach der Installation bearbeiten Sie die Ressource *ClusterRole/<namespace>-additional-permissions*
2. Erstellen Sie eine neue Regel für die gewünschten apiGroups und Ressourcen mit den Verben ["get", "watch", "list"]. Siehe <https://kubernetes.io/docs/reference/access-authn-authz/rbac/>
3. Übernehmen Sie die Änderungen auf das Cluster

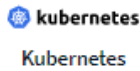
Installation und Konfiguration des Kubernetes Monitoring Operator

Cloud Insights bietet die **Kubernetes Monitoring Operator** für Kubernetes an. Navigieren Sie zu **Kubernetes > Collectors > +Kubernetes Collector**, um einen neuen Operator bereitzustellen.

Bevor Sie den Kubernetes Monitoring Operator installieren

Siehe "[Voraussetzungen](#)" Dokumentation vor der Installation oder dem Upgrade des Kubernetes Monitoring Operator.

Installieren des Kubernetes Monitoring Operator



Deploy NetApp Monitoring Operator

Quickly install and configure a Kubernetes Operator to send cluster information to Cloud Insights.

Select existing API Access Token or create a new one

KEY2024 (...vw6NdM) ▼

+ API Access Token

Production Best Practices ?

Installation Instructions

[Need Help?](#)

Please review the [pre-requisites](#) for installing the NetApp Kubernetes Monitoring Operator.
To update an existing operator installation please follow [these steps](#).

1 Define Kubernetes cluster name and namespace

Provide the Kubernetes cluster name and specify a namespace for deploying the monitoring components.

Cluster	Namespace
<input type="text" value="clustername"/>	<input type="text" value="netapp-monitoring"/>

2 Download the operator YAML files

Execute the following download command in a *bash* prompt.

Copy Download Command Snippet

⊞ Reveal Download Command Snippet

This snippet includes a unique access key that is valid for 24 hours.

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

Schritte zum Installieren des Kubernetes Monitoring Operator Agent auf Kubernetes:

1. Geben Sie einen eindeutigen Cluster-Namen und einen eindeutigen Namespace ein. Wenn Sie es sind [Aktualisierung](#) Verwenden Sie aus einem früheren Kubernetes-Operator den gleichen Cluster-Namen und Namespace.
2. Sobald diese eingegeben wurden, können Sie den Download-Befehl-Snippet in die Zwischenablage kopieren.
3. Fügen Sie das Snippet in ein `bash` Fenster ein und führen Sie es aus. Die Installationsdateien des Bedieners werden heruntergeladen. Beachten Sie, dass das Snippet einen eindeutigen Schlüssel hat und für 24 Stunden gültig ist.
4. Wenn Sie ein benutzerdefiniertes oder privates Repository haben, kopieren Sie das optionale Bild-Pull-Snippet, fügen Sie es in eine `bash`-Shell ein und führen Sie es aus. Nachdem die Bilder gezogen wurden, kopieren Sie sie in Ihr privates Repository. Stellen Sie sicher, dass Sie dieselben Tags und Ordnerstrukturen beibehalten. Aktualisieren Sie die Pfade in `Operator-Deployment.yaml` sowie die Einstellungen des Docker-Repository in `Operator-config.yaml`.
5. Prüfen Sie bei Bedarf die verfügbaren Konfigurationsoptionen, z. B. Proxy- oder private Repository-Einstellungen. Sie können mehr über lesen "[Konfigurationsoptionen](#)".
6. Wenn Sie bereit sind, stellen Sie den Operator bereit, indem Sie den kubectl Apply-Snippet kopieren, herunterladen und ausführen.
7. Die Installation wird automatisch ausgeführt. Klicken Sie anschließend auf die Schaltfläche „Next“.

8. Wenn die Installation abgeschlossen ist, klicken Sie auf die Schaltfläche „Next“. Achten Sie darauf, auch die Datei *Operator-Secrets.yaml* zu löschen oder sicher zu speichern.

Wenn Sie einen Proxy verwenden, lesen Sie mehr über [Proxy wird konfiguriert](#).

Wenn Sie über ein benutzerdefiniertes Repository verfügen, lesen Sie mehr über [Ein benutzerdefiniertes/privates Docker-Repository verwenden](#).

Kubernetes-Monitoring-Komponenten

Cloud Insights-Kubernetes-Monitoring besteht aus vier Monitoring-Komponenten:

- Cluster-Kennzahlen
- Netzwerkleistung und -Zuordnung (optional)
- Ereignisprotokolle (optional)
- Änderungsanalyse (optional)

Die oben aufgeführten optionalen Komponenten sind standardmäßig für jeden Kubernetes-Collector aktiviert. Wenn Sie sich entscheiden, keine Komponente für einen bestimmten Collector zu benötigen, können Sie sie deaktivieren, indem Sie zu **Kubernetes > Collectors** navigieren und im Collector-Menü „drei Punkte“ rechts auf dem Bildschirm *Modify Deployment* auswählen.

NetApp / Observability / Collectors

Data Collectors 21 Acquisition Units 4 Kubernetes Collectors				
Kubernetes Collectors (13)				
View Upgrade/Delete Documentation + Kubernetes Collector Filter...				
Cluster Name ↑	Status	Operator Version	Network Performance and Map	Change Analysis
au-pod	Outdated	1.1540.0	1.347.0	1.162.0
jks-troublemaker	Latest	1.1579.0	N/A	1.201.0
oom-test	Outdated	1.1555.0	N/A	1.161.0

Der Bildschirm zeigt den aktuellen Status jeder Komponente an und ermöglicht es Ihnen, Komponenten für diesen Collector nach Bedarf zu deaktivieren oder zu aktivieren.

Cluster Information

Kubernetes Cluster
ci-demo-01

Network Performance and Map
Enabled - Online

Event Logs
Enabled - Online

Change Analysis
Enabled - Online

Deployment Options

[Need Help?](#)

☒ Network Performance and Map

☒ Event Logs

☒ Change Analysis

Cancel

Complete Modification

Aktualisierung

Upgrade auf den neuesten Kubernetes Monitoring Operator

Ermitteln Sie, ob eine AgentConfiguration bei dem vorhandenen Operator vorhanden ist (wenn Ihr Namespace nicht der Standardwert *netapp-Monitoring* ist, ersetzen Sie den entsprechenden Namespace):

```
kubectl -n netapp-monitoring get agentconfiguration netapp-monitoring-configuration
```

Wenn eine AgentConfiguration vorhanden ist:

- [Installieren](#) Der letzte Operator über den vorhandenen Operator.
 - Stellen Sie sicher, dass Sie es sind [Die neuesten Container-Bilder werden angezeigt](#) Wenn Sie ein benutzerdefiniertes Repository verwenden.

Wenn AgentConfiguration nicht vorhanden ist:

- Notieren Sie sich den von Cloud Insights erkannten Cluster-Namen (wenn Ihr Namespace nicht der standardmäßige netapp-Monitoring ist, ersetzen Sie den entsprechenden Namespace):

```
kubectl -n netapp-monitoring get agent -o  
jsonpath='{.items[0].spec.cluster-name}'
```

* Erstellen Sie eine Sicherung des bestehenden Operators (wenn Ihr Namespace nicht der Standard-netapp-Überwachung ist, ersetzen Sie den entsprechenden Namespace):

```
kubectl -n netapp-monitoring get agent -o yaml > agent_backup.yaml
* <<to-remove-the-kubernetes-monitoring-operator,Deinstallieren>> Der vorhandene Operator.
* <<installing-the-kubernetes-monitoring-operator,Installieren>> Der neueste Operator.
```

- Verwenden Sie denselben Cluster-Namen.
- Nachdem Sie die neuesten Operator YAML-Dateien heruntergeladen haben, können Sie alle in Agent_Backup.yaml gefundenen Anpassungen vor der Bereitstellung an den heruntergeladenen Operator-config.yaml übertragen.
- Stellen Sie sicher, dass Sie es sind [Die neuesten Container-Bilder werden angezeigt](#) Wenn Sie ein benutzerdefiniertes Repository verwenden.

Anhalten und Starten des Kubernetes Monitoring Operator

So beenden Sie den Kubernetes Monitoring Operator:

```
kubectl -n netapp-monitoring scale deploy monitoring-operator --replicas=0
```

So starten Sie den Kubernetes Monitoring Operator:

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

Deinstallation

Um den Kubernetes Monitoring Operator zu entfernen

Beachten Sie, dass der Standard-Namespace für den Kubernetes Monitoring Operator „netapp-Monitoring“ ist. Wenn Sie Ihren eigenen Namespace festgelegt haben, ersetzen Sie diesen Namespace in diesen und allen nachfolgenden Befehlen und Dateien.

Neuere Versionen des Überwachungsoperators können mit den folgenden Befehlen deinstalliert werden:

```
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>
```

Wenn der Überwachungsoperator in seinem eigenen dedizierten Namespace bereitgestellt wurde, löschen Sie den Namespace:

```
kubectl delete ns <NAMESPACE>
```

Wenn der erste Befehl „Keine Ressourcen gefunden“ zurückgibt, verwenden Sie die folgenden Anweisungen, um ältere Versionen des Überwachungsoperators zu deinstallieren.

Führen Sie jeden der folgenden Befehle in der Reihenfolge aus. Abhängig von Ihrer aktuellen Installation können einige dieser Befehle Nachrichten 'object not found' zurückgeben. Diese Meldungen können sicher ignoriert werden.

```
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>
```

Wenn zuvor eine Sicherheitskontextbeschränkung erstellt wurde:

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

Über Kube-State-Metrics

Der NetApp Kubernetes Monitoring Operator installiert seine eigenen kube-State-Metriken, um Konflikte mit anderen Instanzen zu vermeiden.

Informationen über Kube-State-Metrics finden Sie unter ["Auf dieser Seite"](#).

Konfigurieren/Anpassen des Bedieners

Diese Abschnitte enthalten Informationen zur Anpassung Ihrer Bedienerkonfiguration, zur Arbeit mit Proxy, zur Verwendung eines benutzerdefinierten oder privaten Docker-Repositorys oder zur Arbeit mit OpenShift.

Konfigurationsoptionen

Die am häufigsten geänderten Einstellungen können in der benutzerdefinierten Ressource *AgentConfiguration* konfiguriert werden. Sie können diese Ressource bearbeiten, bevor Sie den Operator bereitstellen, indem Sie die Datei *Operator-config.yaml* bearbeiten. Diese Datei enthält kommentierte Beispiele für Einstellungen. Siehe Liste von ["Verfügbare Einstellungen"](#) Für die neueste Version des Bedieners.

Sie können diese Ressource auch bearbeiten, nachdem der Operator bereitgestellt wurde, indem Sie den

folgenden Befehl verwenden:

```
kubectl -n netapp-monitoring edit AgentConfiguration
```

Um festzustellen, ob die bereitgestellte Version des Operators AgentConfiguration unterstützt, führen Sie den folgenden Befehl aus:

```
kubectl get crd agentconfigurations.monitoring.netapp.com
```

Wenn die Meldung „Fehler vom Server (notfound)“ angezeigt wird, muss Ihr Bediener aktualisiert werden, bevor Sie die AgentConfiguration verwenden können.

Proxy-Unterstützung Wird Konfiguriert

Es gibt zwei Stellen, an denen Sie einen Proxy in Ihrer Umgebung verwenden können, um den Kubernetes Monitoring Operator zu installieren. Es kann sich um dieselben oder separate Proxy-Systeme handeln:

- Proxy benötigt bei Ausführung des Installationscodes Snippet (mit "Curl"), um das System, an dem das Snippet ausgeführt wird, mit Ihrer Cloud Insights-Umgebung zu verbinden
- Proxy für die Kommunikation mit Ihrer Cloud Insights Umgebung durch das Ziel-Kubernetes-Cluster

Wenn Sie einen Proxy für eine oder beide dieser Optionen verwenden, müssen Sie zuerst sicherstellen, dass Ihr Proxy für eine gute Kommunikation mit Ihrer Cloud Insights-Umgebung konfiguriert ist, um den Kubernetes Operating Monitor zu installieren. Wenn Sie über einen Proxy verfügen und über den Server/die VM auf Cloud Insights zugreifen können, von dem aus Sie den Operator installieren möchten, wird Ihr Proxy wahrscheinlich richtig konfiguriert.

Für den Proxy, der zur Installation des Kubernetes Operating Monitor verwendet wird, legen Sie vor der Installation des Operators die Umgebungsvariablen `http_Proxy`/`https_Proxy` fest. In einigen Proxy-Umgebungen müssen Sie möglicherweise auch die Variable `no_Proxy Environment` festlegen.

Um die Variablen festzulegen, führen Sie die folgenden Schritte auf Ihrem System aus * bevor* den Kubernetes Monitoring Operator installiert:

1. Legen Sie die Umgebungsvariable `https_Proxy` und/oder `http_Proxy` für den aktuellen Benutzer fest:
 - a. Wenn der Proxy, der eingerichtet wird, keine Authentifizierung (Benutzername/Passwort) aufweist, führen Sie den folgenden Befehl aus:

```
export https_proxy=<proxy_server>:<proxy_port>
.. Wenn der Proxy, der eingerichtet wird, über Authentifizierung
(Benutzername/Passwort) verfügt, führen Sie folgenden Befehl aus:
```

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

Damit der Proxy, der für das Kubernetes-Cluster zur Kommunikation mit der Cloud Insights-Umgebung verwendet wird, den Kubernetes Monitoring Operator installieren kann, nachdem alle diese Anweisungen gelesen wurden.

Konfigurieren Sie den Proxy-Abschnitt von AgentConfiguration in Operator-config.yaml, bevor Sie den Kubernetes Monitoring Operator bereitstellen.

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

Verwenden eines benutzerdefinierten oder privaten Docker Repositorys

Standardmäßig zieht der Kubernetes Monitoring Operator Container-Images aus dem Cloud Insights-Repository. Wenn Sie ein Kubernetes-Cluster als Ziel für das Monitoring verwenden und der Cluster so konfiguriert ist, dass er nur Container-Images aus einem benutzerdefinierten oder privaten Docker-Repository oder der Container-Registrierung zieht, müssen Sie den Zugriff auf die Container konfigurieren, die vom Kubernetes Monitoring Operator benötigt werden.

Führen Sie das „Image Pull Snippet“ aus der NetApp Monitoring Operator Installationskachel aus. Dieser Befehl meldet sich beim Cloud Insights-Repository an, zieht alle Image-Abhängigkeiten für den Operator und meldet sich vom Cloud Insights-Repository ab. Wenn Sie dazu aufgefordert werden, geben Sie das angegebene temporäre Repository-Passwort ein. Mit diesem Befehl werden alle vom Bediener verwendeten Bilder heruntergeladen, einschließlich optionaler Funktionen. Nachfolgend sehen Sie, für welche Funktionen diese Bilder verwendet werden.

Core Operator-Funktionalität und Kubernetes Monitoring

- netapp Monitoring
- ci-kube-rbac-Proxy
- ci-ksm

- ci-telegraf
- Distroless-root-user

Ereignisprotokoll

- ci-Fluent-Bit
- ci-kubernetes-Event-Exporteur

Netzwerkleistung und -Zuordnung

- ci-Netz-Beobachter

Übertragen Sie das Operator-Docker-Image gemäß Ihren Unternehmensrichtlinien in das private/lokale/unternehmenseigene Docker-Repository. Stellen Sie sicher, dass die Bild-Tags und Verzeichnispfade zu diesen Bildern in Ihrem Repository mit denen im Cloud Insights-Repository übereinstimmen.

Bearbeiten Sie die Bereitstellung des Monitoring-Operators in `Operator-Deployment.yaml`, und ändern Sie alle Bildverweise, um Ihr privates Docker-Repository zu verwenden.

```
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>
```

Bearbeiten Sie die AgentConfiguration in `Operator-config.yaml`, um die neue Position des Docker-Repo zu berücksichtigen. Erstellen Sie ein neues `imagePullSecret` für Ihr privates Repository. Weitere Informationen finden Sie unter <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
```

OpenShift-Anweisungen

Wenn Sie OpenShift 4.6 oder höher ausführen, müssen Sie die AgentConfiguration in `Operator-config.yaml` bearbeiten, um die Einstellung `runPrivileged` zu aktivieren:

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

OpenShift kann zusätzliche Sicherheitsstufen implementieren, die den Zugriff auf einige Kubernetes-Komponenten blockieren könnten.

Ein Hinweis über Geheimnisse

Um die Berechtigung für den Kubernetes Monitoring Operator zum Anzeigen der geheimen Daten im gesamten Cluster zu entfernen, löschen Sie vor der Installation die folgenden Ressourcen aus der Datei *Operator-Setup.yaml*:

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

Wenn es sich um ein Upgrade handelt, löschen Sie auch die Ressourcen aus Ihrem Cluster:

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

Wenn die Änderungsanalyse aktiviert ist, ändern Sie die Optionen *AgentConfiguration* oder *Operator-config.yaml*, um den Änderungsmanagementabschnitt zu entkommentieren und *kindsToIgnoreFromWatch*: *"Secrets"* im Bereich Change-Management aufzunehmen. Notieren Sie sich das Vorhandensein und die Position von einfachen und doppelten Anführungszeichen in dieser Zeile.

```
# 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"'
...
```

Überprüfen Von Kubernetes Prüfsummen

Das Cloud Insights Agent-Installationsprogramm führt Integritätsprüfungen durch. Einige Benutzer müssen jedoch vor der Installation oder Anwendung heruntergeladener Artefakte möglicherweise ihre eigenen Überprüfungen durchführen. Um einen nur-Download-Vorgang durchzuführen (im Gegensatz zum Standard-Download-and-install), können diese Benutzer den Agent-Installation Befehl erhalten von der UI und entfernen Sie die nachhängbare "Installation" Option.

Führen Sie hierzu folgende Schritte aus:

1. Kopieren Sie das Agent Installer-Snippet wie angewiesen.
2. Anstatt das Snippet in ein Befehlsfenster einzufügen, fügen Sie es in einen Texteditor ein.
3. Entfernen Sie den nachfolgenden „--install“ aus dem Befehl.
4. Kopieren Sie den gesamten Befehl aus dem Texteditor.
5. Fügen Sie es nun in Ihr Befehlsfenster ein (in einem Arbeitsverzeichnis) und führen Sie es aus.
 - Download und Installation (Standard):

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

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

Der Download-Only-Befehl lädt alle erforderlichen Artefakte vom Cloud Insights in das Arbeitsverzeichnis herunter. Die Artefakte umfassen, dürfen aber nicht beschränkt sein auf:

- Ein Installationsskript
- Einer Umgebungsdatei
- YAML-Dateien
- Eine signierte Prüfsumme-Datei (sha256.signed)
- Eine PEM-Datei (netapp_cert.pem) zur Signaturverifizierung

Das Installationsskript, die Umgebungsdatei und die YAML-Dateien können mittels Sichtprüfung verifiziert werden.

Die PEM-Datei kann durch Bestätigung des Fingerabdrucks wie folgt verifiziert werden:

```
1A918038E8E127BB5C87A202DF173B97A05B4996  
Genauer gesagt,
```

```
openssl x509 -fingerprint -sha1 -noout -inform pem -in netapp_cert.pem  
Die signierte Prüfsummendatei kann mit der PEM-Datei verifiziert werden:
```

```
openssl smime -verify -in sha256.signed -CAfile netapp_cert.pem -purpose  
any
```

Sobald alle Artefakte zufriedenstellend überprüft wurden, kann die Agenteninstallation durch Ausführen von gestartet werden:

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

Toleranzen und Verfleckungen

Die DemonSets *netapp-ci-telegraf-ds*, *netapp-ci-Fluent-Bit-ds* und *netapp-ci-net-Observer-l4-ds* müssen für jeden Node im Cluster einen Pod planen, damit Daten auf allen Nodes korrekt erfasst werden. Der Operator wurde so konfiguriert, dass er einige bekannte **Fehler** toleriert. Wenn Sie auf Ihren Nodes benutzerdefinierte Taints konfiguriert haben und damit verhindern, dass Pods auf jedem Knoten ausgeführt werden, können Sie für diese Taints eine **Toleration** erstellen "[In der AgentConfiguration](#)". Wenn Sie auf alle Nodes im Cluster benutzerdefinierte Taints angewendet haben, müssen Sie der Operator-Bereitstellung auch die erforderlichen Toleranzen hinzufügen, damit der Operator-Pod geplant und ausgeführt werden kann.

Weitere Informationen zu Kubernetes "[Tönungen und Tolerationen](#)".

Kehren Sie zum zurück "[NetApp Kubernetes Monitoring Operator Installation Seite](#)"

Fehlerbehebung

Bei Problemen beim Einrichten des Kubernetes Monitoring Operator sollten Sie Folgendes versuchen:

Problem:	Versuchen Sie dies:
Ich sehe keinen Hyperlink/Verbindung zwischen meinem Kubernetes Persistent Volume und dem entsprechenden Back-End Storage-Gerät. Mein Kubernetes Persistent Volume wird mit dem Hostnamen des Storage-Servers konfiguriert.	Befolgen Sie die Schritte, um den bestehenden Telegraf-Agent zu deinstallieren, und installieren Sie dann den neuesten Telegraf-Agent erneut. Sie müssen Telegraf Version 2.0 oder höher verwenden, und Ihr Kubernetes Cluster Storage muss von Cloud Insights aktiv überwacht werden.

Problem:	Versuchen Sie dies:
<p>Ich sehe Nachrichten in den Protokollen, die folgendermaßen aussehen:</p> <p>E0901 15:21:39.962145 1 Reflector.go:178] k8s.io/kube-State-metrics/internal/Store/Builder.go:352: Konnte *v1.MutatingWebhookKonfiguration: Der Server konnte die angeforderte Ressource nicht finden E0901 15:21:43.168161 1 Reflector.go:178] k8s.io/kube-State-metrics/internal/Store/Builder.go:352: Fehler beim Auflisten von *v1.Lease: Der Server konnte die angeforderte Ressource nicht finden (get Leases.Coordination.k8s.io) Usw.</p>	<p>Diese Nachrichten können auftreten, wenn Sie kube-State-Metrics Version 2.0.0 oder höher mit Kubernetes-Versionen unter 1.20 ausführen.</p> <p>So erhalten Sie die Kubernetes-Version:</p> <p><i>Kubectl Version</i></p> <p>So erhalten Sie die kube-State-metrics-Version:</p> <p><i>Kubectl get deploy/kube-State-metrics -o jsonpath='{..image}'</i></p> <p>Um zu verhindern, dass diese Meldungen stattfinden, können Benutzer ihre Bereitstellung von kube-State-Metrics ändern, um die folgenden Leasings zu deaktivieren:</p> <p><i>Mutatingwebhookkonfigurationen</i> <i>Validatingwebhookkonfigurationen</i> <i>Volumeattachments-Ressourcen</i></p> <p>Genauer gesagt können sie das folgende CLI-Argument verwenden:</p> <p>Ressourcen=zertifiziertigningrequests,configmaps,cronjobs,demonsets, Bereitstellungen,Endpunkte,horizontalpodautoscalers,ingresses,Jobs,limitranges, Namespaces,Netzwerkrichtlinien,Nodes,persistent Volumeclaims,persistent Volumes, poddisruptionbudgets,Pods,Replikasets,Replikationcontroller,resourcequotas, Secrets,Services,Statefulsets,Storageclasses</p> <p>Die Standardressourcenliste lautet:</p> <p>„Zertificatizingrequest,configmaps,cronjobs,demonsets,Bereitstellungen, Endpunkte,horizontalpodautoscalers,ingresses,Jobs,Leases,limitranges, mutatingwebhookkonfigurationen,Namespaces,Netzwerkrichtlinien,Nodes,persistent Volumeclaims,persistent Volumes,poddisruptionbudgets,Pods,Replikasets,resourcequotas,Secrets,Services, statefulsets,storageclasses,validatingwebhookkonfigurationen, volumeattachments“</p>

Problem:	Versuchen Sie dies:
<p>Ich sehe Fehlermeldungen von Telegraf wie die folgenden, aber Telegraf startet und läuft:</p> <pre>Oct 11 14:23:41 ip-172-31-39-47 systemd[1]: Startete den Plugin-gesteuerten Server-Agent für die Berichterstattung von Kennzahlen in InfluxDB. Okt 11 14:23:41 ip-172-31-39-47 telegraf[1827]: Time=„2021-10-11T14:23:41Z“ Level=error msg=„konnte kein Cache-Verzeichnis erstellen. /Etc/telegraf/.Cache/snowflake, err: Mkdir /etc/telegraf/.ca Che: Erlaubnis verweigert. Ignored\n“ func=„gosnowflake.(*defaultLogger).Errorf“ file=„log.go:120“ Okt. 11 14:23:41 ip-172-31-39-47 telegraf[1827]: Time=„2021-10-11T14:23:41Z“ Level=error msg=„Öffnen fehlgeschlagen. Ignoriert. Open /etc/telegraf/.Cache/snowflake/ocsp_response_Cache.json: Nicht so Datei oder Verzeichnis\n“ func=„gosnowflake.(*defaultLogger).Errorf“ file=„log.go:120“ Okt. 11 14:23:41 ip-172-31-39-47 telegraf[1827]: 2021-10-11T14:23:41Z ! Telegraf 1.19.3 Starten</pre>	<p>Dies ist ein bekanntes Problem. Siehe "Dieser GitHub-Artikel" Entnehmen. Solange Telegraf läuft, können Benutzer diese Fehlermeldungen ignorieren.</p>
<p>Auf Kubernetes berichten meine Telegraf POD(s) die folgende Fehlermeldung: "Fehler bei der Verarbeitung von mountstats-Info: Mountstats-Datei konnte nicht geöffnet werden: /Hostfs/proc/1/mountstats, Fehler: Open /hostfs/proc/1/mountstats: Berechtigung verweigert"</p>	<p>Wenn SELinux aktiviert und durchgesetzt wird, wird wahrscheinlich verhindert, dass die Telegraf PODs auf die Datei /proc/1/mountstats auf dem Kubernetes-Knoten zugreifen. Um diese Einschränkung zu überwinden, bearbeiten Sie die Agentkonfiguration und aktivieren Sie die runPrivileged-Einstellung. Weitere Informationen finden Sie im "OpenShift-Anweisungen".</p>
<p>Auf Kubernetes meldet mein Telegraf ReplicaSet POD den folgenden Fehler:</p> <pre>[inputs.prometheus] Fehler im Plugin: Konnte keypair /etc/kubernetes/pki/etcd/Server.crt:/etc/kubernetes/pki/etcd/Server.key nicht laden: Öffnen /etc/kubernetes/pki/etcd/Server.crt: Datei oder Verzeichnis nicht vorhanden</pre>	<p>Der Pod Telegraf ReplicaSet soll auf einem Knoten ausgeführt werden, der als Master oder für etc bestimmt ist. Wenn der ReplicaSet-Pod auf einem dieser Knoten nicht ausgeführt wird, werden diese Fehler angezeigt. Überprüfen Sie, ob Ihre Master/etcd-Knoten eine Tönungswalle haben. Fügen Sie in diesem Fall die erforderlichen Verträgen in das Telegraf ReplicaSet, telegraf-rs ein.</p> <p>Bearbeiten Sie beispielsweise das ReplicaSet...</p> <p>Kubectrl bearbeiten rs telegraf-rs</p> <p>...Und fügen Sie die entsprechenden Toleranzen in die Spezifikation ein. Starten Sie anschließend den Pod ReplicaSet neu.</p>

Problem:	Versuchen Sie dies:
Ich habe eine PSP/PSA Umgebung. Hat dies Auswirkungen auf meinen Überwachungsoperator?	<p>Wenn Ihr Kubernetes-Cluster mit Pod-Sicherheitsrichtlinie (PSP) oder Pod Security Admission (PSA) ausgeführt wird, müssen Sie ein Upgrade auf den aktuellen Kubernetes Monitoring Operator durchführen. Führen Sie die folgenden Schritte aus, um auf den aktuellen Bediener mit Unterstützung für PSP/PSA zu aktualisieren:</p> <p>1. Deinstallieren Der vorherige Überwachungsoperator:</p> <pre> Kubectrl delete Agent-Monitoring-netapp -n netapp-Monitoring Kubectrl löschen ns netapp-Monitoring Kubectrl löschen crd agents.monitoring.netapp.com Kubectrl delete clusterrole Agent-Manager-role Agent-Proxy-role Agent-metrics-reader Kubectrl delete clusterrolebinding Agent-Manager-rolebinding Agent-Proxy-rolebinding Agent-Cluster-admin-rolebinding </pre> <p>2. Installieren Die neueste Version des Überwachungsbedieners.</p>
Ich habe Probleme beim Versuch, den Operator bereitzustellen, und ich habe PSP/PSA in Gebrauch.	<p>1. Bearbeiten Sie den Agenten mit dem folgenden Befehl:</p> <pre> Kubectrl -n <name-space>-Bearbeitungsagent </pre> <p>2. Markieren Sie "Sicherheit-Politik-aktiviert" als "falsch". Dadurch werden Pod-Sicherheitsrichtlinien und Pod-Sicherheitszulassung deaktiviert und der Bediener kann die Bereitstellung durchführen. Bestätigen Sie die Bestätigung mit folgenden Befehlen:</p> <pre> Kubectrl get psp (sollte zeigen, dass die Pod-Sicherheitsrichtlinie entfernt wurde) Kubectrl get all -n <namespace> (sollte zeigen, dass nichts gefunden wird) </pre>
„ImagePullBackoff“-Fehler erkannt	<p>Diese Fehler können auftreten, wenn Sie über ein benutzerdefiniertes oder privates Docker-Repository verfügen und den Kubernetes Monitoring Operator noch nicht so konfiguriert haben, dass er es richtig erkennt. Weitere Informationen Info zur Konfiguration für benutzerdefinierte/private Repo.</p>

Problem:	Versuchen Sie dies:
<p>Ich habe ein Problem mit der Installation meines Monitoring-Bedieners, und die aktuelle Dokumentation hilft mir nicht, es zu lösen.</p>	<p>Erfassen oder notieren Sie die Ausgabe der folgenden Befehle, und wenden Sie sich an den technischen Support.</p> <pre> kubect1 -n netapp-monitoring get all kubect1 -n netapp-monitoring describe all kubect1 -n netapp-monitoring logs <monitoring-operator-pod> --all -containers=true kubect1 -n netapp-monitoring logs <telegraf-pod> --all -containers=true </pre>
<p>NET-Observer (Workload Map)-Pods im Operator Namespace befinden sich in CrashLoopBackOff</p>	<p>Diese Pods entsprechen dem Workload Map-Datensammler für Network Observability. Versuchen Sie Folgendes:</p> <ul style="list-style-type: none"> • Überprüfen Sie die Protokolle eines der Pods, um die minimale Kernel-Version zu bestätigen. Beispiel: <pre> ---- {"CI-Tenant-id":"your-Tenant-id","Collector-Cluster":"your-k8s-Cluster-Name","environment":"prod","Level":"error","msg":"failed in validation. Grund: Kernelversion 3.10.0 ist kleiner als die minimale Kernelversion von 4.18.0","Time":"2022-11-09T08:23:08Z"} ---- </pre> <ul style="list-style-type: none"> • Net-Observer PODs benötigen die Linux Kernel Version mindestens 4.18.0. Überprüfen Sie die Kernel-Version mit dem Befehl „uname -r“ und stellen Sie sicher, dass sie >= 4.18.0 sind
<p>Pods werden im Operator Namespace ausgeführt (Standard: netapp-Monitoring), es werden jedoch keine Daten in der UI für die Workload-Zuordnung oder Kubernetes-Metriken in Abfragen angezeigt</p>	<p>Überprüfen Sie die Zeiteinstellung auf den Knoten des K8S-Clusters. Für eine genaue Prüfung und Datenberichterstattung wird dringend empfohlen, die Zeit auf dem Agent-Rechner mit Network Time Protocol (NTP) oder Simple Network Time Protocol (SNTP) zu synchronisieren.</p>

Problem:	Versuchen Sie dies:
<p>Einige der Net-Observer-Pods im Namespace Operator befinden sich im Status „Ausstehend“</p>	<p>NET-Observer ist ein DaemonSet und führt in jedem Knoten des K8s-Clusters einen Pod aus.</p> <ul style="list-style-type: none"> • Beachten Sie den Pod, der sich im Status „Ausstehend“ befindet, und prüfen Sie, ob ein Ressourcenproblem für CPU oder Speicher vorliegt. Stellen Sie sicher, dass der erforderliche Arbeitsspeicher und die erforderliche CPU im Knoten verfügbar sind.
<p>Ich sehe Folgendes in meinen Protokollen sofort nach der Installation des Kubernetes Monitoring Operator:</p> <p>[inputs.prometheus] Fehler im Plugin: Fehler beim Erstellen einer HTTP-Anforderung an 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: Kein solcher Host</p>	<p>Diese Meldung wird normalerweise nur angezeigt, wenn ein neuer Operator installiert ist und der Pod „<i>telegraf-rs</i>“ vor dem Einschalten des Pod „<i>ksm</i>“ steht. Diese Meldungen sollten beendet werden, sobald alle Pods ausgeführt werden.</p>
<p>Ich sehe keine Kennzahlen für die Kubernetes-Kronjobs, die in meinem Cluster vorhanden sind, erfasst.</p>	<p>Überprüfen Ihrer Kubernetes Version (d. h. <code>kubectl version</code>). Wenn es v1.20.x oder niedriger ist, ist dies eine erwartete Einschränkung. Die mit dem Kubernetes Monitoring Operator implementierte Version von kube-State-Metrics unterstützt nur v1.cronjob. Bei Kubernetes 1.20.x und niedriger befindet sich die Ressource cronjob unter v1beta.cronjob. Daher können kube-State-Metriken die Ressource cronjob nicht finden.</p>

Problem:	Versuchen Sie dies:
<p>Nach der Installation des Bedieners geben die telegraf-ds-Pods CrashLoopBackOff ein und die POD-Protokolle zeigen „su: Authentication failure“ an.</p>	<p>Bearbeiten Sie den Abschnitt telegraf in <i>AgentConfiguration</i>, und setzen Sie <i>dockerMetricCollectionEnabled</i> auf false. Weitere Informationen finden Sie im Abschnitt des Bedieners "Konfigurationsoptionen".</p> <p>HINWEIS: wenn Sie die Cloud Insights Federal Edition verwenden, können Benutzer mit Einschränkungen hinsichtlich der Verwendung von <i>su</i> keine Docker-Metriken erfassen, da der Zugriff auf den Dockersockel entweder den telegraf-Container als root ausführen muss oder <i>su</i> verwenden muss, um den telegraf-Benutzer zur Docker-Gruppe hinzuzufügen. Die Docker Metric Collection und die Verwendung von <i>su</i> sind standardmäßig aktiviert. Um beides zu deaktivieren, entfernen Sie den Eintrag <i>telegraf.Docker</i> in der Datei <i>AgentConfiguration</i>:</p> <pre> ... Spez.: ... telegraf: ... - Name: docker Run-Modus: - DemonSet Ersetzungen: - SCHLÜSSEL: DOCKER_UNIX_SOCKET_PLACEHOLDER Wert: unix:///run/Docker.Sock </pre>
<p>Ich sehe wiederholte Fehlermeldungen wie die folgenden in meinen Telegraf-Protokollen:</p> <p>E! [Agent] Fehler beim Schreiben in Outputs.http: Post "https://<tenant_url>/Rest/v1/Lake/ingest/influxdb": Kontext-Deadline überschritten (Client. Zeitüberschreitung beim Warten auf Header überschritten)</p>	<p>Bearbeiten Sie den Abschnitt telegraf in <i>AgentConfiguration</i>, und erhöhen Sie <i>outputTimeout</i> auf 10s. Weitere Informationen finden Sie im Abschnitt des Bedieners "Konfigurationsoptionen".</p>
<p>Ich vermisste <i>involvedobject</i> Daten für einige Event Logs.</p>	<p>Stellen Sie sicher, dass Sie die Schritte im befolgt haben "Berechtigungen" Abschnitt oben.</p>
<p>Wieso werden zwei Monitoring Operator Pods ausgeführt, einer mit dem Namen netapp-CI-Monitoring-Operator-<pod> und der andere mit dem Namen Monitoring-Operator-<pod>?</p>	<p>Ab dem 12. Oktober 2023 hat Cloud Insights den Betreiber refaktoriert, um unseren Nutzern besser zu dienen. Damit diese Änderungen vollständig übernommen werden, müssen Sie dies tun Entfernen Sie den alten Bediener Und Installieren Sie den neuen.</p>

Problem:	Versuchen Sie dies:
Meine kubernetes-Ereignisse berichteten unerwartet nicht mehr an Cloud Insights.	<p>Rufen Sie den Namen des POD für den Event-Exporter ab:</p> <pre>`kubectl -n netapp-monitoring get pods`</pre>
grep event-exporter	awk '{print \$1}'
<p>sed 's/event-exporter./event-exporter/'</p> <p>Es sollte entweder „netapp-CI-Event-Exporteur“ oder „Event-Exporteur“ sein. Bearbeiten Sie anschließend den Monitoring-Agent <code>kubectl -n netapp-monitoring edit agent</code>, Und legen Sie den Wert für LOG_FILE so fest, dass der entsprechende POD-Name für den Event-Exporter im vorherigen Schritt angezeigt wird. Genauer gesagt sollte LOG_FILE auf <code>"/var/log/Containers/netapp-CI-Event-exporteur.log"</code> oder <code>"/var/log/Containers/Event-exporteur*.log"</code> gesetzt werden</p> <pre>.... fluent-bit: ... - name: event-exporter-ci substitutions: - key: LOG_FILE values: - /var/log/containers/netapp-ci-event-exporter*.log</pre> <p>Alternativ kann man auch Deinstallieren Und Neu installieren Der Agent.</p>	<p>Ich sehe POD(s), die vom Kubernetes-Monitoring-Operator bereitgestellt werden, aufgrund unzureichender Ressourcen.</p>
Weitere Informationen finden Sie im Kubernetes Monitoring Operator "Konfigurationsoptionen" Um die CPU- und/oder Speichergrenzen je nach Bedarf zu erhöhen.	Durch ein fehlendes Image oder eine ungültige Konfiguration wurden die netapp-CI-kube-State-metrics Pods nicht gestartet oder nicht einsatzbereit gemacht. Jetzt bleibt StatefulSet stecken und Konfigurationsänderungen werden nicht auf die Pods mit den netapp-CI-kube-State-Metriken angewendet.
Das StatefulSet befindet sich in A "Defekt" Bundesland. Nachdem Sie Konfigurationsprobleme behoben haben, springen die netapp-CI-kube-State-metrics-Pods an.	Pods mit netapp-CI-kube-Status-Metriken können nicht gestartet werden, nachdem ein Kubernetes Operator Upgrade ausgeführt wurde. Es wird ErrImagePull geworfen (es konnte nicht das Image entfernt werden).
Versuchen Sie, die Pods manuell zurückzusetzen.	„Event disordered as being older than maxEventAgeSeconds“ Meldungen werden für meinen Kubernetes Cluster unter Log Analysis beobachtet.

Problem:	Versuchen Sie dies:
Ändern Sie den Operator <i>agentkonfiguration</i> , und erhöhen Sie die Erweiterung <i>Event-exporteur-maxEventAgeSeconds</i> (d. h. auf 60s), <i>Event-exporteur-kubeQPS</i> (d. h. auf 100) und <i>Event-exporteur-kubeBurst</i> (d. h. auf 500). Weitere Informationen zu diesen Konfigurationsoptionen finden Sie im " Konfigurationsoptionen " Seite.	Telegraf warnt vor unzureichenden, abschließbaren Speichern oder stürzt ab.

Weitere Informationen finden Sie im "[Unterstützung](#)" Oder auf der "[Data Collector Supportmatrix](#)".

Konfigurationsoptionen Für Kubernetes Monitoring Operator

Der "[Kubernetes Monitoring Operator](#)" Die Konfiguration kann angepasst werden.

In der folgenden Tabelle sind die möglichen Optionen für die *AgentConfiguration*-Datei aufgeführt:

Komponente	Option	Beschreibung
Agent		Konfigurationsoptionen, die allen Komponenten gemeinsam sind, die der Bediener installieren kann. Diese können als "globale" Optionen betrachtet werden.
	DockerRepo	Ein ockerRepo override, um Bilder von Kunden private Docker-Repos im Vergleich zu Cloud Insights Docker Repo zu ziehen. Der Standardwert ist Cloud Insights Docker Repo
	DockerImagePullSecret	Optional: Ein Geheimnis für den Kunden private repo
	ClusterName	Freitextfeld, das einen Cluster über alle Kundencluster eindeutig identifiziert. Diese sollte bei einem Cloud Insights-Mandanten eindeutig sein. Der Standardwert ist das, was der Kunde in die Benutzeroberfläche für das Feld „Cluster Name“ eingibt
	Proxy Format: Proxy: Server: Anschluss: Benutzername: Kennwort: Noproxy: IsTelegrafProxyEnabled: IsAuProxyEnabled: IsFluentbitProxyEnabled: IsCollectorProxyEnabled:	Optional zum Festlegen des Proxys. Dies ist in der Regel der Unternehmensvertreter des Kunden.

Komponente	Option	Beschreibung
telegraf		Konfigurationsoptionen, mit denen die telegraf-Installation des Bedieners angepasst werden kann
	Erfassungsintervall	Messgrößen-Erfassungsintervall, in Sekunden (max. = 60 s)
	DsCpuLimit	CPU-Limit für telegraf ds
	DsMemLimit	Speicherlimit für telegraf ds
	DsCpuRequest	CPU-Anforderung für telegraf ds
	DsMemRequest	Speicheranforderung für telegraf ds
	RsCpuLimit	CPU-Limit für telegraf rs
	RsMemLimit	Speichergrenze für telegraf rs
	RsCpuRequest	CPU-Anforderung für telegraf rs
	RsMemRequest	Speicheranforderung für telegraf rs
	RunPrivileged	Führen Sie den telegraf Container im privilegierten Modus aus. Setzen Sie diese Einstellung auf TRUE, wenn SELinux auf Ihren K8s-Knoten aktiviert ist
	Stapelgröße	Siehe "Telegraf-Konfigurationsdokumentation"
	BufferLimit	Siehe "Telegraf-Konfigurationsdokumentation"
	Rundintervall	Siehe "Telegraf-Konfigurationsdokumentation"
	SammlungJitter	Siehe "Telegraf-Konfigurationsdokumentation"
	Präzision	Siehe "Telegraf-Konfigurationsdokumentation"
	Flushintervall	Siehe "Telegraf-Konfigurationsdokumentation"
	FlushJitter	Siehe "Telegraf-Konfigurationsdokumentation"
	AusgabeTimeout	Siehe "Telegraf-Konfigurationsdokumentation"
	DsToleranzen	telegraf-ds zusätzliche Toleranzen.
	RsToleranzen	telegraf-rs zusätzliche Toleranzen.
	SkipProcessorsAfterAggregatoren	Siehe "Telegraf-Konfigurationsdokumentation"
	Ungeschützt	Siehe das "Bekanntes Problem mit Telegraf" . Durch die Einstellung „ <i>Unprotected</i> “ wird der Kubernetes Monitoring Operator angewiesen, Telegraf mit dem auszuführen <code>--unprotected</code> Flagge.
status-Kennzahlen von kube		Konfigurationsoptionen, mit denen die installation von kube-Statusmetriken des Operators angepasst werden kann
	CpuLimit	CPU-Limit für die bereitstellung von kube-State-Metriken
	MemLimit	MEM-Limit für die implementierung von kube-State-Metriken

Komponente	Option	Beschreibung
	CpuRequest	CPU-Anforderung für die Bereitstellung von kube-Statusmetriken
	MemRequest	MEM-Anforderung für die Bereitstellung von kube-Statuskennzahlen
	Ressourcen	Eine kommasetrennte Liste der zu erfassenden Ressourcen. Beispiel: Cronjobs,demonsets,Bereitstellungen,ingress,Jobs,Namespaces,Nodes,persistent Volumeclaims, persistent Volumes,Pods,Replikasets,resourcequotas,Services,statefulsets
	Toleranzen	zusätzliche Toleranzen für kube-State-Metriken.
	Etiketten	Eine kommasetrennte Liste von Ressourcen, die kube-State-metrics erfassen sollte Beispiel: Cronjobs=[*],demonsets=[*],Deployments=[*],ingresses=[*],Jobs=[*],Namespaces=[*],Nodes=[*], Persistent volumeclaims=[*],persistent Volumes=[*],Pods=[*],replikasets=[*],resourcequotas=[*],Services=[*],statefulsets=[*]
Protokolle		Konfigurationsoptionen, mit denen die Protokollsammlung und die Installation des Bedieners angepasst werden können
	Wieder FromHead	Wahr/falsch, sollte fließendes Bit das Protokoll vom Kopf lesen
	Zeitüberschreitung	Timeout in Sekunden
	DnsMode	TCP/UDP, Modus für DNS
	Fluent-Bit-Tolerationen	Fluent-Bit-ds zusätzliche Toleranzen.
	Ereignis-Exporteur-Tolerationen	Ereignis-Exporteur zusätzliche Toleranzen.
	Event-Exporteur-maxEventAgeSeconds	Ereignis-Exporteur max. Ereignisalter. Siehe https://github.com/jkroepke/resmoio-kubernetes-event-exporter
Workload-Zuordnung		Konfigurationsoptionen, mit denen die Erfassung der Workload-Zuordnung und die Installation des Operators angepasst werden können
	CpuLimit	CPU-Limit für Netto-Observer ds
	MemLimit	MEM-Grenze für Netto-Beobachter ds
	CpuRequest	CPU-Anforderung für Netto-Observer-ds
	MemRequest	MEM-Anforderung für Netto-Beobachter ds

Komponente	Option	Beschreibung
	MetricAggregationInterval	Intervall für die metrische Aggregation in Sekunden
	BpfPollInterval	BPF-Abfrageintervall in Sekunden
	EnableDNSLookup	True/false, DNS-Suche aktivieren
	I4-Tolerationen	NET-Observer-I4-ds zusätzliche Toleranzen.
	RunPrivileged	True/false - Setzen Sie runPrivileged auf true, wenn SELinux auf Ihren Kubernetes-Knoten aktiviert ist.
Änderungsmanagement		Konfigurationsoptionen für das Kubernetes Change Management und die Analyse
	CpuLimit	CPU-Limit für Change-Observer-watch-rs
	MemLimit	MEM Limit für Change-Observer-Watch-rs
	CpuRequest	CPU-Anforderung für Change-Observer-watch-rs
	MemRequest	MEM-Anforderung für Change-Observer-Watch-rs
	Ausfallerklärungsintervall in Minuten	Intervall in Minuten, nach dem eine nicht erfolgreiche Bereitstellung eines Workloads als fehlgeschlagen markiert wird
	EinsatzAggrInterval in Sekunden	Häufigkeit, mit der Ereignisse zur laufenden Workload-Bereitstellung gesendet werden
	Nicht-WorkloadAggrInterval in Sekunden	Häufigkeit der Kombination und des Sendeens von nicht-Workload-Implementierungen
	TermsToAkt	Ein Satz von regulären Ausdrücken, die in Env-Namen und Datenkarten verwendet werden, deren Wert bearbeitet wird Beispielbegriffe: „pwd“, „Passwort“, „Token“, „apikey“, „API-key“, „jwt“
	Zusätzlich KindsToWatch	Eine kommasetrennte Liste mit weiteren Arten, die von den vom Sammler überwachten Standardtypen überwacht werden sollen
	KindsToIgnoreFromWatch	Eine kommasetrennte Liste von Arten, die ignoriert werden sollen, wenn sie von den vom Sammler überwachten Standardtypen überwacht werden
	LogRecordAggrInterval in Sekunden	Häufigkeit, mit der Protokolldatensätze vom Collector an CI gesendet werden
	Überwachen von Toleranzen	Change-Observer-watch-ds zusätzliche Toleranzen. Nur abgekürztes Einzelzeilenformat. Beispiel: '{key: Taint1, Operator: Existiert, Effekt: NoSchedule},{key: Taint2, Operator: Existiert, Effekt: NoExecute}'

Beispieldatei für AgentConfiguration

Unten finden Sie eine *AgentConfiguration*-Beispieldatei.

```

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 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#agent

    # # 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,persistentvolumeclaims,persistentvolumes,pods,replicasets,resourcequotas,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=[*],namespaces=[*],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: ''

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

# # 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.
# enabledDNSLookup: '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",
"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 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: ''

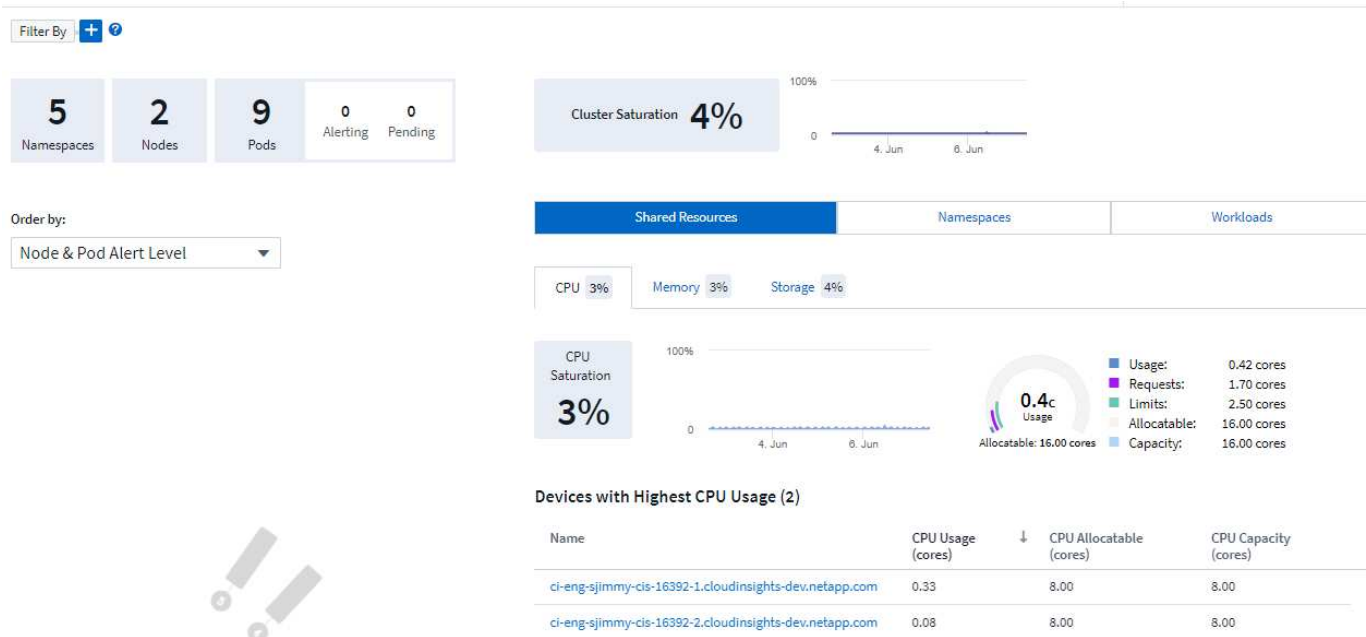
# # Frequency with which log records are sent to CI from the collector
# logRecordAggrIntervalSeconds: '20'

# # 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: ''

```

Detailseite Zu Kubernetes Cluster

Auf der Kubernetes-Cluster-Detailseite wird eine detaillierte Übersicht über das Kubernetes-Cluster angezeigt.



Namespace, Node und Pod-Anzahl

Die Zählungen oben auf der Seite zeigen Ihnen die Gesamtzahl der Namespaces, Nodes und Pods im Cluster sowie die Anzahl der Pods, die derzeit Warnungen und ausstehend sind.

Shared Ressourcen und Sättigung

Oben rechts auf der Detailseite ist Ihre Cluster-Sättigung als aktueller Prozentsatz sowie ein Diagramm, das den letzten Trend im Laufe der Zeit zeigt. Cluster-Sättigung ist der höchste CPU-, Arbeitsspeicher- oder Storage-Sättigung bei jedem Zeitpunkt.

Im Folgenden wird die Seite standardmäßig **Nutzung von freigegebenen Ressourcen** mit Registerkarten für CPU, Speicher und Speicher angezeigt. Auf jeder Registerkarte werden der Sättigungspunkt und der Trend über die Zeit mit zusätzlichen Nutzungsdetails angezeigt. Für den Storage ist der angezeigte Wert der größere Backend- und Filesystem-Sättigung, die unabhängig voneinander berechnet wird.

Die Geräte mit der höchsten Nutzung werden in einer Tabelle unten angezeigt. Klicken Sie auf einen beliebigen Link, um diese Geräte zu durchsuchen.

Namespaces

Auf der Registerkarte Namespaces wird eine Liste aller Namespaces in der Kubernetes-Umgebung angezeigt. Die CPU- und Arbeitsspeicherauslastung sowie die Anzahl der Workloads in jedem Namespace werden angezeigt. Klicken Sie auf die Namenslinks, um die einzelnen Namespaces zu erkunden.

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

Workloads

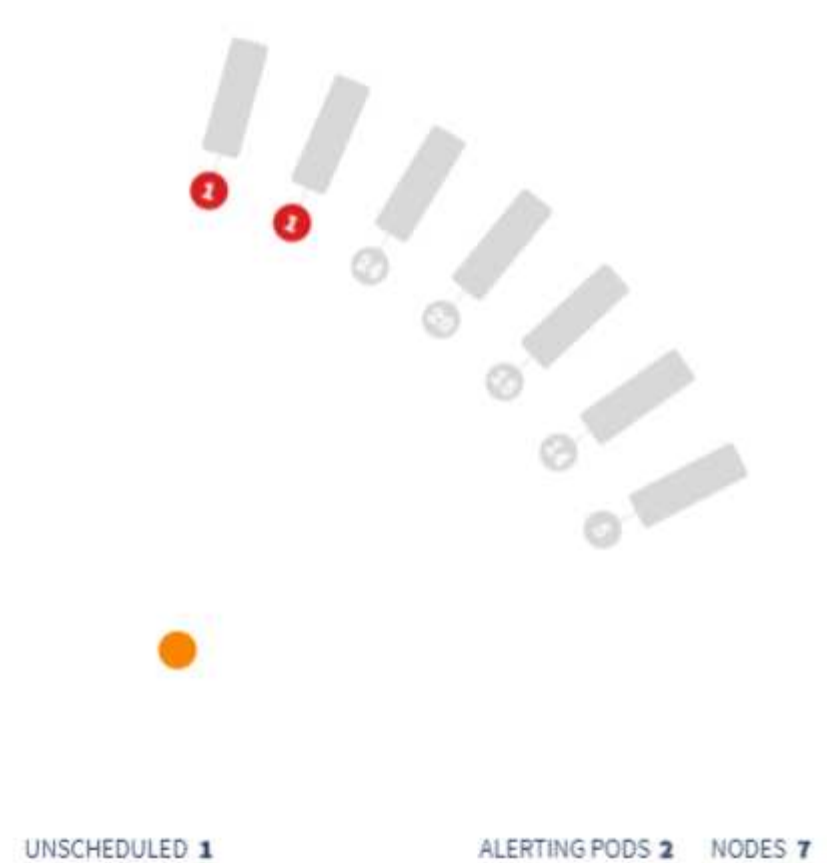
Auf der Registerkarte Workloads wird zudem eine Liste der Workloads in den einzelnen Namespace angezeigt. Auch hier wird die CPU- und Arbeitsspeicherauslastung angezeigt. Wenn Sie auf den Namespace-Links klicken, ist jeder Link bohrt.

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

Das Cluster „Wheel“



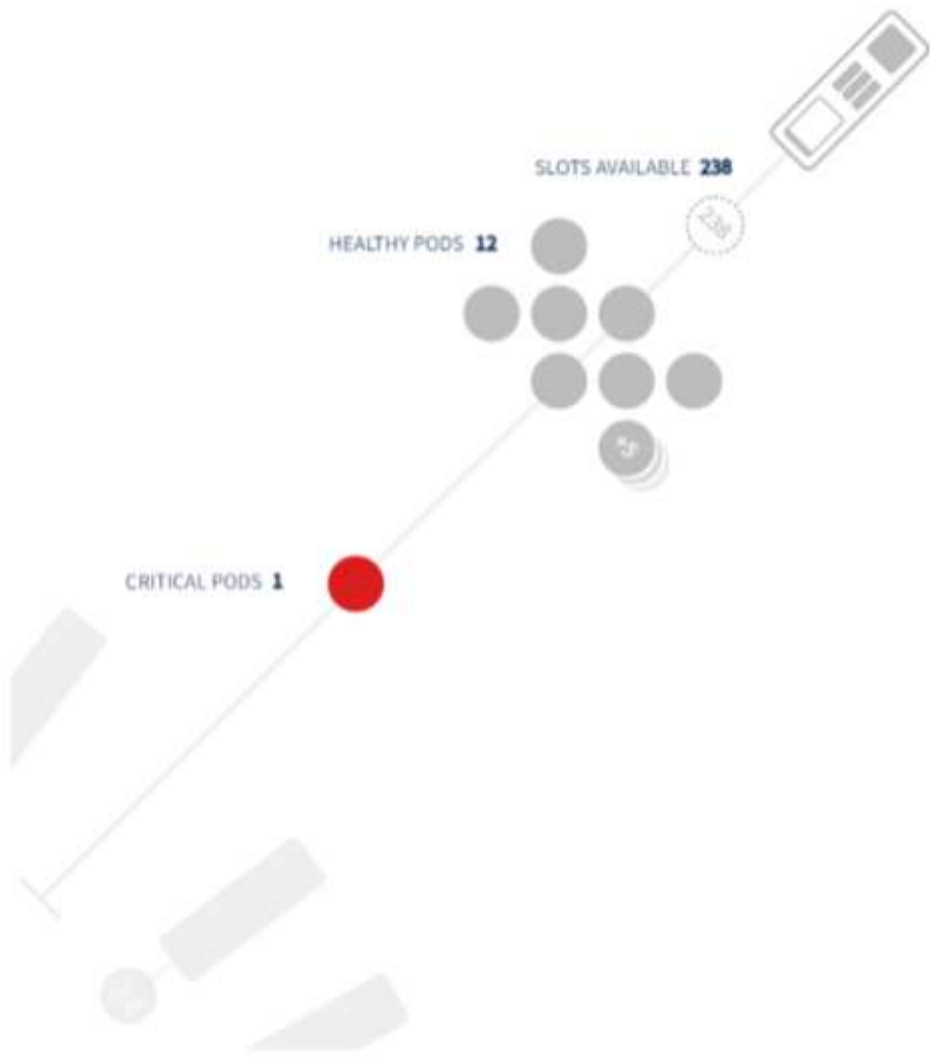
Im Abschnitt „Cluster „Wheel“ finden Sie auf einen Blick den Zustand der Nodes und des POD. Weitere Informationen hierzu finden Sie unter. Wenn Ihr Cluster mehr Nodes enthält, als in diesem Bereich der Seite angezeigt werden kann, können Sie das Rad mit den verfügbaren Schaltflächen drehen.

AlarmPods oder Nodes werden rot angezeigt. Die Bereiche „Warnung“ werden orange angezeigt. PODs, die nicht geplant sind (d.h. unangebracht), werden in der unteren Ecke des Cluster „Wheel“ angezeigt.

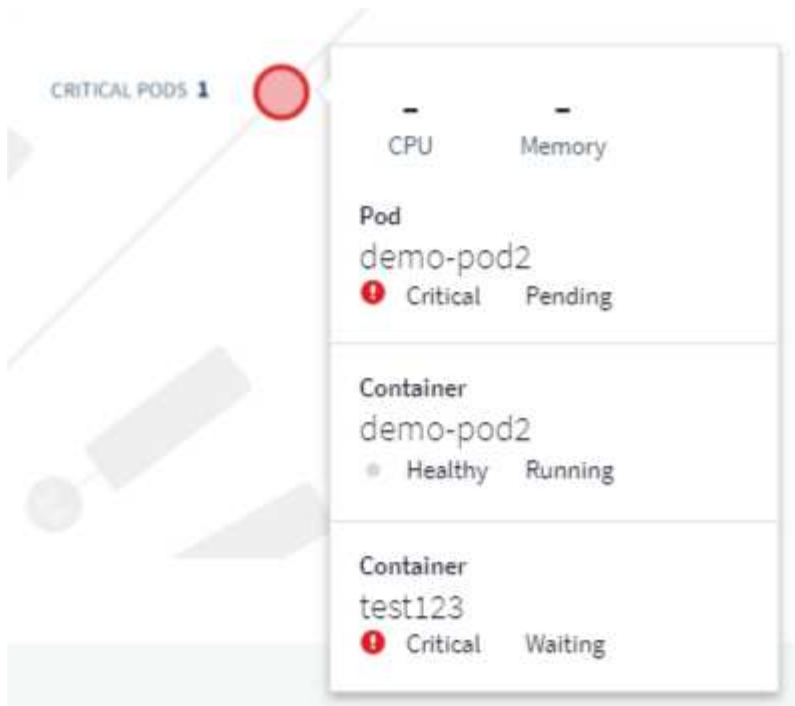
Wenn Sie sich über einen Pod (Kreis) oder Knoten (Balken) bewegen, wird die Ansicht des Knotens erweitert.



Wenn Sie in der Ansicht auf den Pod oder Node klicken, wird die Ansicht „erweiterter Node“ vergrößert.



Von hier aus können Sie mit dem Mauszeiger auf ein Element zeigen, um Details zu diesem Element anzuzeigen. Beispiel: Wenn Sie den Mauszeiger über den kritischen POD in diesem Beispiel halten, werden Details zu diesem POD angezeigt.



Sie können Filesystem-, Speicher- und CPU-Informationen anzeigen, indem Sie den Mauszeiger über die Knoten-Elemente bewegen.



Ein Hinweis zu den Messgeräten

Die Speicher- und CPU-Anzeigen zeigen drei Farben, da sie *used* in Bezug auf *zuteilbare Kapazität* und *Gesamtkapazität* zeigen.

Performance-Monitoring und -Zuordnung des Kubernetes-Netzwerks


Die Kubernetes Network Performance Monitoring and Map Funktion vereinfacht die Fehlerbehebung durch die Zuordnung von Abhängigkeiten zwischen Services (auch Workloads genannt). Sie bietet Echtzeiteinblick in Latenzen und Anomalien bei der Netzwerk-Performance. So können Performance-Probleme erkannt werden, bevor sie sich auf die Benutzer auswirken.

Diese Funktion hilft Unternehmen, durch Analyse und Prüfung des Kubernetes-Traffic-Flows die Gesamtkosten zu senken.

Die wichtigsten Funktionen • die Workload-Map präsentiert Kubernetes-Workload-Abhängigkeiten und -Abläufe und hebt Netzwerk- und Performance-Probleme hervor. • Monitoring des Netzwerkverkehrs zwischen Kubernetes-Pods, Workloads und Nodes; Ermittlung der Quelle von Traffic- und Latenzproblemen • Senkung der Gesamtkosten durch Analyse des Ingress-, Egress-, Regions- und zonenübergreifenden Netzwerk-Traffics.

Voraussetzungen

Bevor Sie die Kubernetes-Netzwerk-Performance-Überwachung und -Zuordnung verwenden können, müssen Sie den konfiguriert haben ["NetApp Kubernetes Monitoring Operator"](#) Um diese Option zu aktivieren. Aktivieren Sie während der Bereitstellung des Operators das Kontrollkästchen „Netzwerkleistung und Zuordnung“, um es zu aktivieren. Sie können diese Option auch aktivieren, indem Sie zu einer Kubernetes-Landing Page navigieren und „Implementierung ändern“ auswählen.

 **kubernetes**
Kubernetes

Configure Data Acquisition

Review Kubernetes cluster information and choose additional data to collect.

Cluster Information

Kubernetes Cluster	Network Performance and Map	Events Log
stream8	Disabled	Disabled

Deployment Options

☒ Network Performance and Map

☒ Events Log

Complete Setup

[Need Help?](#)

Monitore

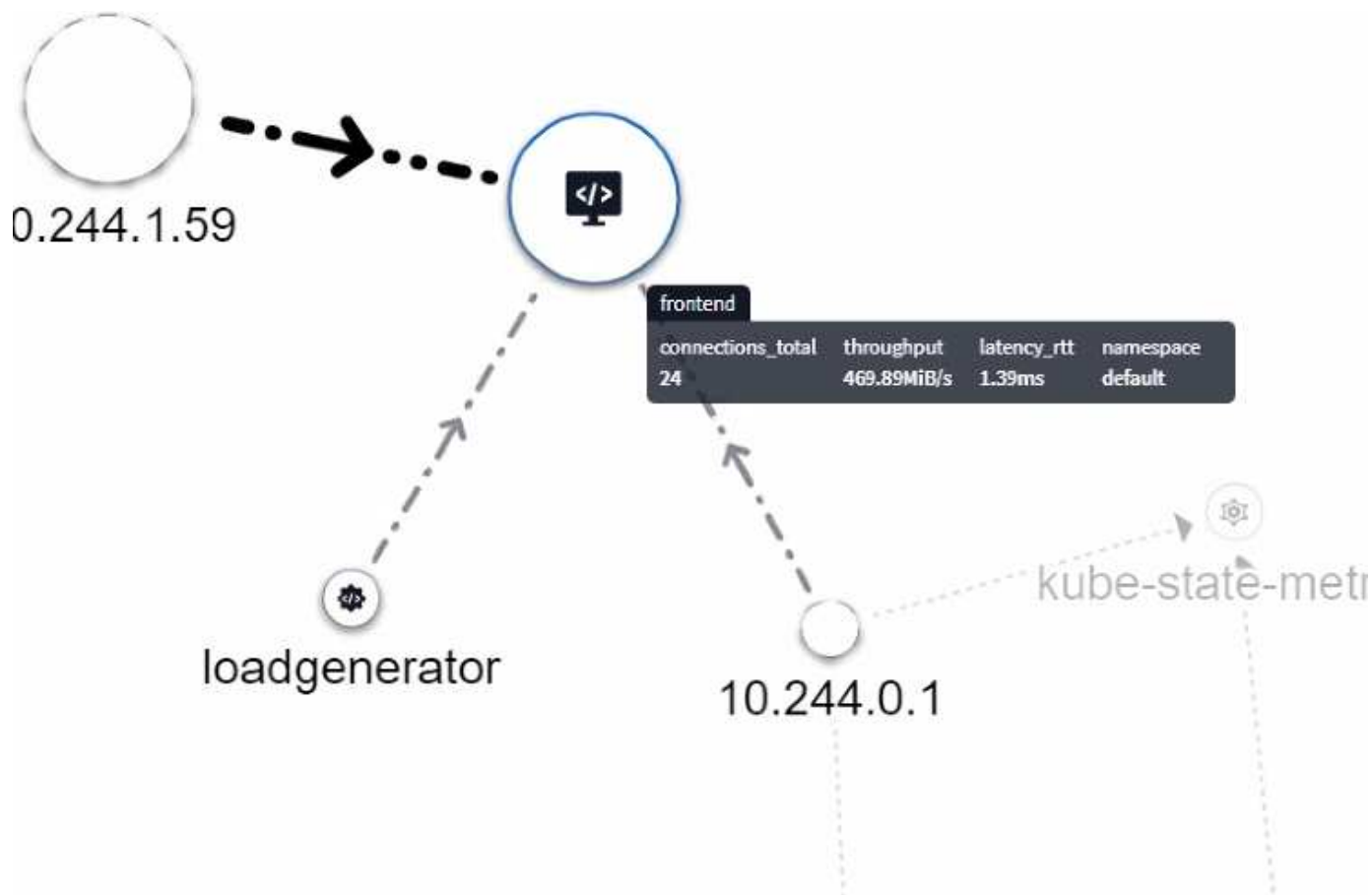
Die Workload Map verwendet "Monitore" Um Informationen abzuleiten. Cloud Insights bietet eine Reihe von Kubernetes-Standardmonitoren an (beachten Sie, dass diese standardmäßig „Paused“ sein können. Sie können die gewünschten Monitore *Resume* (d. h. aktivieren) oder benutzerdefinierte Monitore für kubernetes-Objekte erstellen, die auch von der Workload Map verwendet werden.

Sie können Cloud Insights-Metrik-Warnmeldungen für jeden der unten aufgeführten Objekttypen erstellen. Stellen Sie sicher, dass die Daten nach dem Standardobjekttyp gruppiert sind.

- kubernetes.Workload
- kubernetes.demonset
- kubernetes.deployment
- kubernetes.cronjob
- kubernetes.Job
- kubernetes.Replicaset
- kubernetes.statefulset
- kubernetes.POD
- kubernetes.network_traffic_l4

Die Karte

Die Karte zeigt Services/Workloads und deren Beziehungen zueinander an. Pfeile zeigen die Verkehrsrichtung. Wenn Sie den Mauszeiger über einen Workload halten, werden zusammenfassende Informationen zu diesem Workload angezeigt, wie im folgenden Beispiel zu sehen ist:

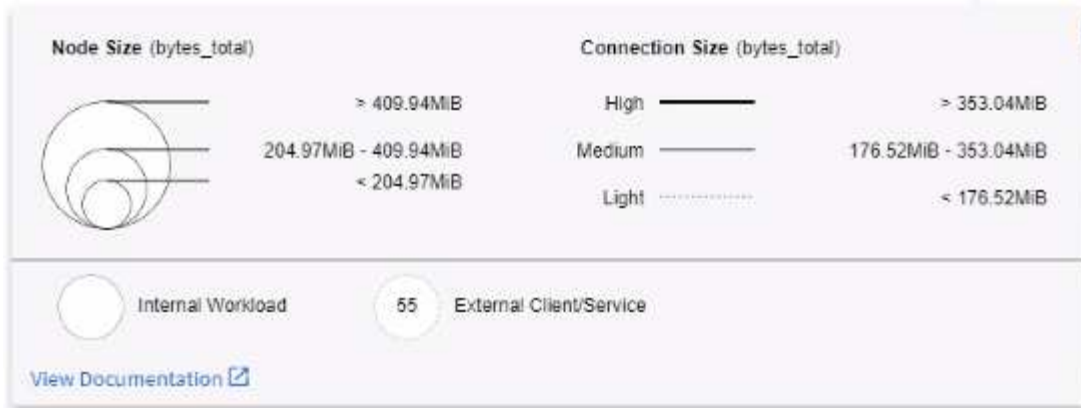


Symbole innerhalb der Kreise stellen verschiedene Dienstypen dar. Beachten Sie, dass Symbole nur sichtbar sind, wenn die zugrunde liegenden Objekte vorhanden sind [Etiketten](#).



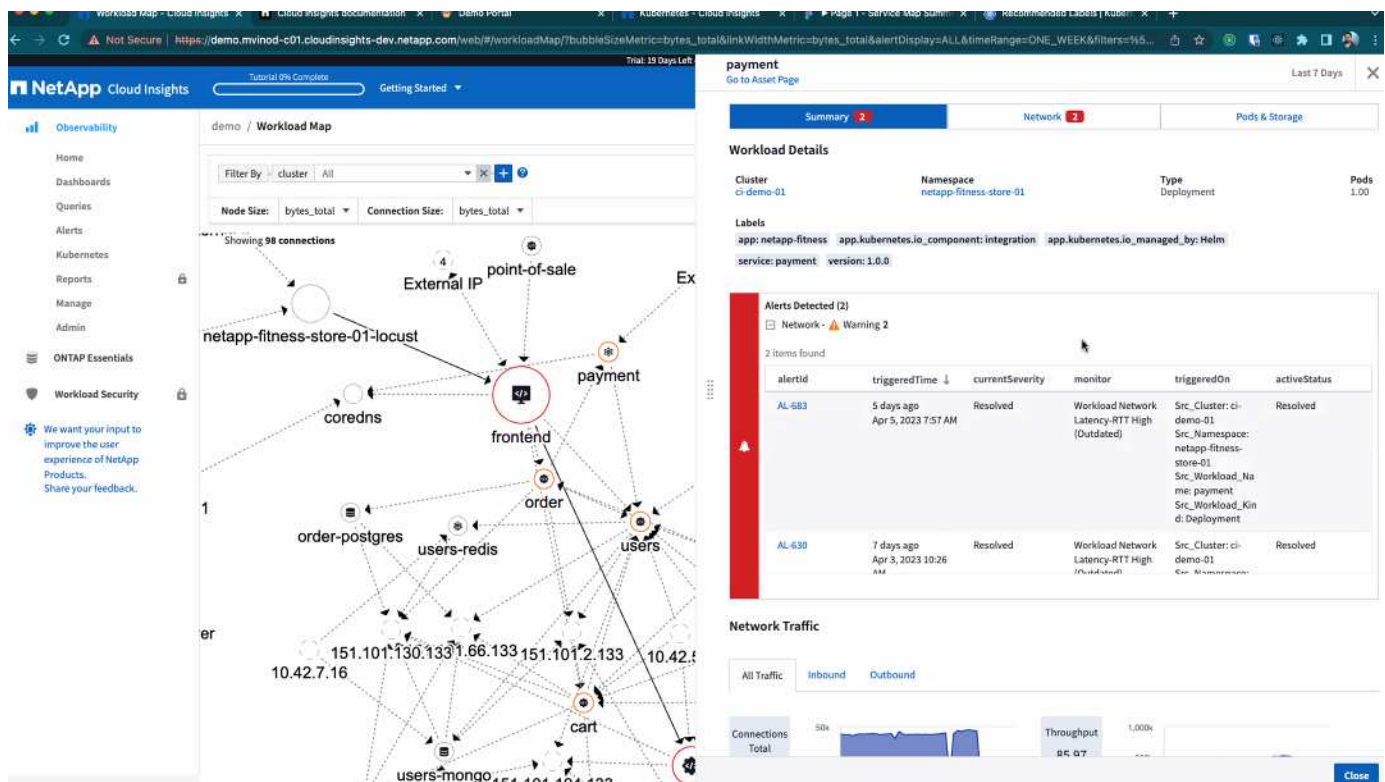
Die Größe jedes Kreises gibt die Knotengröße an. Beachten Sie, dass diese Größen relativ sind. Ihr Browser-Zoom-Level oder die Bildschirmgröße kann sich auf die tatsächlichen Kreisgrößen auswirken. Auf die gleiche Weise gibt Ihnen der Linienstil einen schnellen Überblick über die Verbindungsgröße; fett leuchtete Linien sind stark frequentlicht, während die gestrichelten Linien weniger Verkehr aufweisen.

Zahlen innerhalb der Kreise sind die Anzahl der externen Verbindungen, die derzeit vom Dienst verarbeitet werden.



Workload-Details und -Alarme

Farbige Kreise weisen auf eine Warnung auf Warn- oder kritische Ebene für die Arbeitslast hin. Bewegen Sie den Mauszeiger über den Kreis, um eine Zusammenfassung des Problems zu erhalten, oder klicken Sie auf den Kreis, um ein Slideout-Fenster mit mehr Details zu öffnen.



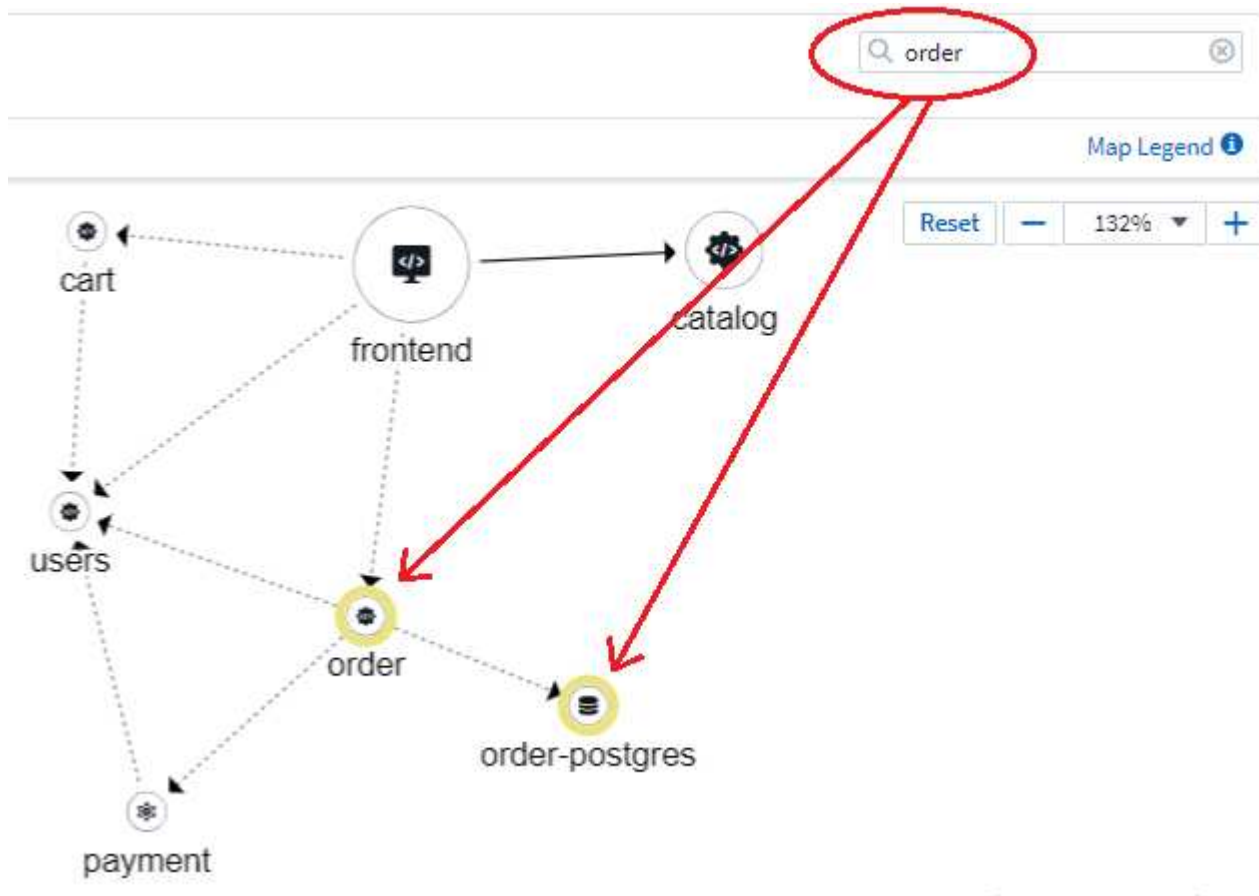
Suchen und Filtern

Wie bei anderen Cloud Insights-Funktionen können Sie auch bei den Filtern einfach den Fokus auf die gewünschten Objekte oder Workload-Attribute legen.

Filter By: cluster All X scope_cluster All X + ?

Node Size: bytes_total Connection Size: bytes_total

Ebenso wird durch Eingabe einer Zeichenfolge im Feld *Find* die übereinstimmenden Workloads hervorgehoben.



Workload-Etiketten

Workload-Bezeichnungen sind erforderlich, wenn die Zuordnung die angezeigten Workload-Typen (d. h. die Kreissymbole) identifizieren soll. Die Bezeichnungen werden wie folgt abgeleitet:

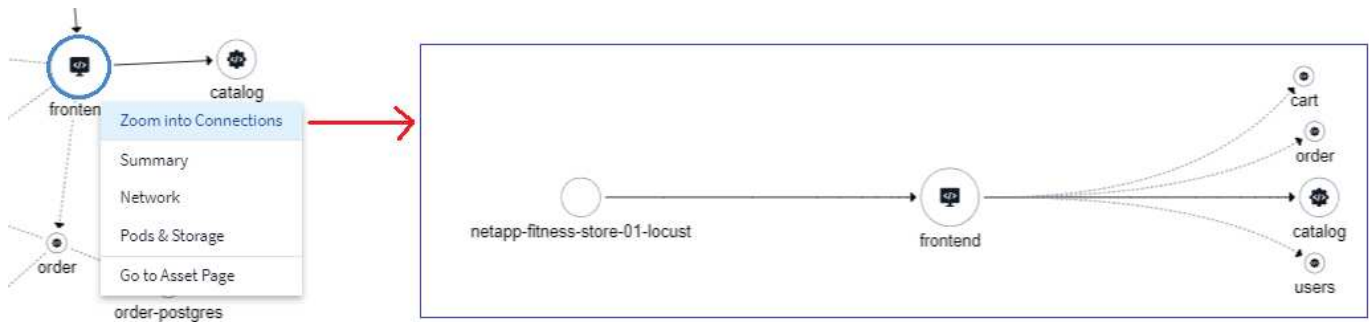
- Name des Dienstes/der Anwendung, der allgemein ausgeführt wird
- Wenn es sich bei der Quelle um einen Pod handelt:
 - Die Bezeichnung leitet sich vom Workload-Etikett des Pods ab
 - Erwartetes Label für den Workload: App.kubernetes.io/component
 - Bezeichnung Name Referenz: <https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels/>
 - Empfohlene Etiketten:
 - Frontend

- Back-End
 - Datenbank
 - Cache
 - Warteschlange
 - kafka
- Wenn sich die Quelle außerhalb des kubernetes-Clusters befindet:
 - Cloud Insights versucht, den DNS-aufgelösten Namen zu analysieren, um den Dienstyp zu extrahieren.

Beispiel: Mit einem DNS-aufgelösten Namen von *s3.eu-north-1.amazonaws.com* wird der aufgelöste Name analysiert, um *s3* als Dienstyp zu erhalten.

So Geht Es Richtig

Mit einem Rechtsklick auf einen Workload erhalten Sie zusätzliche Optionen, um weitere Informationen zu erhalten. Von hier aus können Sie beispielsweise die Ansicht vergrößern, um die Verbindungen für diesen Workload anzuzeigen.



Alternativ können Sie das Detailslideout-Panel öffnen, um die Registerkarte *Summary*, *Network* oder *Pod & Storage* direkt anzuzeigen.



Summary	Network	Pods & Storage
---------	---------	----------------

Network Activities - Inbound (1)

src_workload...	src_namespace	src_workload_...	throughpu... ↓	connections_t...	latency_rtt (ms)	tcp_retransmit...
netapp-fitness...	locust	Deployment	14,193,748.78	653.19	3.74	2,578.00

Network Activities - Outbound (4)

dst_workloa...	dst_namespace	dst_workload_...	throughpu... ↓	connections_t...	latency_rtt (ms)	tcp_retransmit...
catalog	netapp-fitness-...	Deployment	14,166,417.02	2,425.07	149.37	13,850.00
cart	netapp-fitness-...	Deployment	12,479.90	638.97	65.10	0.00
order	netapp-fitness-...	Deployment	4,515.16	161.84	65.07	0.00

Durch Auswahl von *Gehe zu Anlagenseite* wird die detaillierte Zielseite für die Anlage für den Workload geöffnet.

Filter By + ?

2/2

Pods: Current / Desired

2

Up-to-date

0

Unavailable

Namespace
netapp-fitness-store-01Type
DeploymentDate Created
Apr 11, 2023 11:34 AM

Labels

-

260mc

CPU



Highest CPU Demand by Pod

132.76m frontend-7...9f8f-284kb

127.55m frontend-7...9f8f-gd8mk

0.17GiB

Memory



Highest Memory Demand by Pod

0.09 GiB frontend-7...9f8f-284kb

0.09 GiB frontend-7...9f8f-gd8mk

0.00GiB

Total PVC Capacity claimed

Pods (2)

Pod Name ↑	Status	Healthy Containers	cpu_usage_nanocores (mc)	memory_rss_bytes (GiB)
frontend-7fccd9f8f-284kb	● Healthy Running	1 of 1	133	0.09
frontend-7fccd9f8f-gd8mk	● Healthy Running	1 of 1	128	0.09

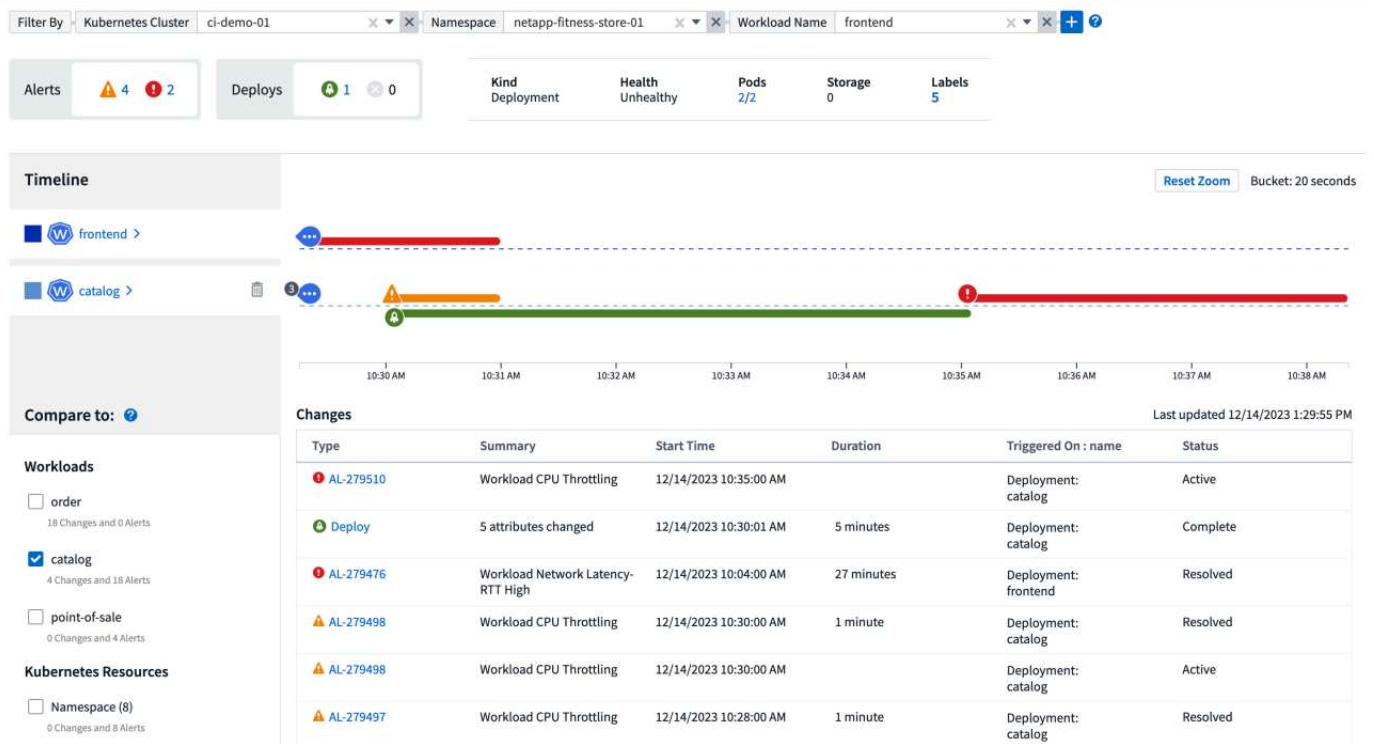
Kubernetes Change Analytics

Kubernetes Change Analytics bietet Ihnen einen All-in-One-Überblick über die letzten Änderungen an Ihrer K8s-Umgebung. Warnmeldungen und Bereitstellungsstatus stehen Ihnen jederzeit zur Verfügung. Mit Change Analytics lassen sich jede Implementierungs- und Konfigurationsänderung nachverfolgen und mit dem Zustand und der Performance von Kubernetes-Services, Infrastruktur und Clustern korrelieren.

Wie hilft die Änderungsanalyse?

- In mandantenfähigen Kubernetes-Umgebungen können Ausfälle aufgrund falsch konfigurierter Änderungen auftreten. Change Analytics unterstützt dies durch die Bereitstellung eines zentralen Fensters zur Ansicht und Korrelation des Systemzustands von Workloads und Konfigurationsänderungen. Dies kann bei der Fehlerbehebung in dynamischen Kubernetes-Umgebungen helfen.

Um Kubernetes Change Analytics anzuzeigen, navigieren Sie zu **Kubernetes > Change Analysis**.

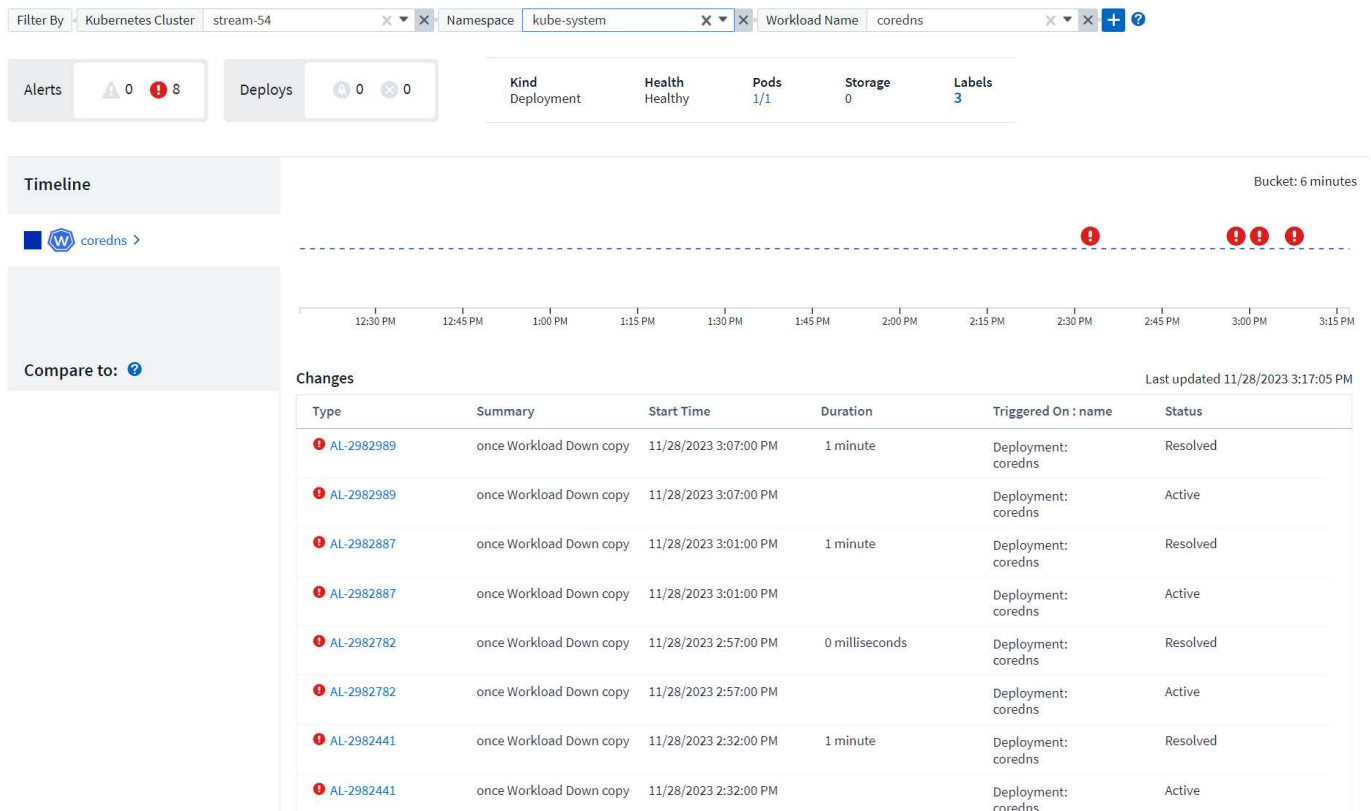


Die Seite wird basierend auf dem aktuell ausgewählten Cloud Insights-Zeitbereich automatisch aktualisiert. Kleinere Zeitbereiche bedeuten eine häufigere Bildschirmerneuerung.

Filtern

Wie bei allen Funktionen von Cloud Insights ist auch die Filterung der Änderungsliste intuitiv: Ganz oben auf der Seite können Sie Werte für Ihren Kubernetes-Cluster, Namespace oder Workload eingeben oder auswählen oder Ihre eigenen Filter hinzufügen, indem Sie auf die Schaltfläche {+} klicken.

Wenn Sie nach unten zu einem bestimmten Cluster, Namespace und Workload filtern (zusammen mit allen anderen Filtern, die Sie festlegen), wird Ihnen ein Zeitplan für die Implementierungen und Warnungen für diesen Workload in diesem Namespace auf dem Cluster angezeigt. Vergrößern Sie die Ansicht weiter, indem Sie auf das Diagramm klicken und es ziehen, um einen bestimmten Zeitraum zu fokussieren.



Schnellstatus

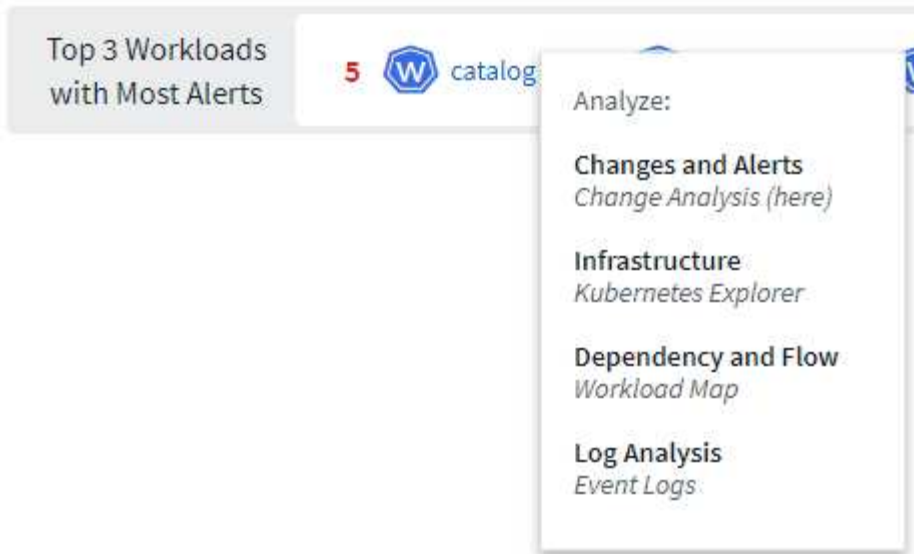
Unterhalb des Filterbereichs befinden sich eine Reihe von High-Level-Indikatoren. Auf der linken Seite ist die Anzahl der Warnungen (Warnung und kritisch). Diese Nummer enthält sowohl *Active* als auch *Resolved* Warnungen. Um nur *Active*-Warnungen anzuzeigen, setzen Sie einen Filter für „Status“ und wählen Sie „aktiv“.



Hier wird auch der Bereitstellungsstatus angezeigt. Auch hier wird standardmäßig die Anzahl der Bereitstellungen *started*, *complete* und *failed* angezeigt. Um nur *failed*-Bereitstellungen anzuzeigen, setzen Sie einen Filter für „Status“ und wählen Sie „failed“ aus.



Als Nächstes kommen die 3 wichtigsten Workloads mit den meisten Warnmeldungen zum Einsatz. Die Zahl in rot neben jedem Workload gibt die Anzahl der Warnmeldungen in Bezug auf diesen Workload an. Klicken Sie auf den Workload-Link, um ihn in Ihre Infrastruktur (Kubernetes Explorer), Abhängigkeiten (Workload Map) oder Protokollanalyse (Event Logs) zu untersuchen.



Detailfenster

Durch Auswahl einer Änderung in der Liste wird ein Fenster geöffnet, in dem die Änderung näher beschrieben wird. Wenn Sie beispielsweise eine fehlgeschlagene Bereitstellung auswählen, wird eine Zusammenfassung der Bereitstellung mit Start- und Endzeiten, Dauer und dem Auslösungsort der Bereitstellung sowie Links zur Untersuchung dieser Ressourcen angezeigt. Außerdem werden der Grund für den Fehler, alle zugehörigen Änderungen und alle zugehörigen Ereignisse angezeigt.

✖ 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

Durch die Auswahl einer Warnmeldung erhalten Sie ebenfalls Details zur Warnmeldung, einschließlich des Monitors, der die Warnmeldung ausgelöst hat, sowie ein Diagramm mit einer visuellen Zeitleiste für die Warnmeldung.

Copyright-Informationen

Copyright © 2024 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFTE SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGEND EINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.