



Kubernetes

Data Infrastructure Insights

NetApp
October 17, 2024

This PDF was generated from https://docs.netapp.com/zh-cn/cloudinsights/kubernetes_landing_page.html on October 17, 2024. Always check docs.netapp.com for the latest.

目录

- Kubernetes 1
 - Kubernetes 集群概述 1
 - 在安装或升级NetApp Kubornetes监控操作员之前 2
 - Kubnetes Monitoring Operator安装和配置 6
 - Kubnetes监控操作员配置选项 22
 - Kubernetes 集群详细信息页面 34
 - Kubnetes网络性能监控和映射 39
 - Kubnetes变更分析 46

Kubernetes

Kubernetes 集群概述

Data Infrastructure Insight Kubernetes Explorer是一款功能强大的工具、可用于显示Kubernetes集群的整体运行状况和使用情况、并可用于轻松深入研究调查领域。

单击*信息板> Kubernetes Explorer*将打开Kubernetes集群列表页面。此概述页面包含您环境中的Kubernetes集群表。

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

集群列表

集群列表显示环境中每个集群的以下信息：

- 集群*名称*。单击集群名称将打开 [** 详细信息页面 **](#)。
- *饱和*百分比。整体饱和是CPU、内存或存储饱和的最高值。
- 集群中的*节点*数量。单击此编号将打开节点列表页面。
- 集群中的* Pod *数量。单击此编号将打开Pod列表页面。
- 集群中的*命名空间*数量。单击此数字将打开"命名空间"列表页面。
- 集群中的*工作负载*数量。单击此数字将打开工作负载列表页面。

细化筛选器

筛选时，在开始键入时，系统会根据当前文本显示一个 * 通配符筛选器 *。选择此选项将返回与通配符表达式匹配的所有结果。您也可以使用 NOT 或 AND 创建 * 表达式 *，也可以选择 "无" 选项来筛选字段中的空值。

Filter By

namespace

kube

X

+

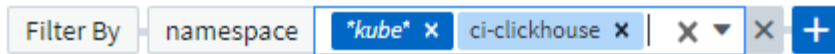
Create wildcard containing "kube"

kube-public

kube-system

None

基于通配符或表达式（例如 NOT ， AND ， "None" 等）在筛选器字段中显示为深蓝色。您直接从列表中选择的項目将以淡蓝色显示。



Kubernetes 筛选器为上下文筛选器，例如，如果您位于特定节点页面上，则 pod_name 筛选器仅会列出与该节点相关的 Pod。此外，如果您对特定命名空间应用筛选器，则 pod_name 筛选器将仅列出该命名空间中节点 *and* 上的 Pod。

请注意，通配符和表达式筛选适用于文本或列表，但不适用于数值，日期或布尔值。

在安装或升级NetApp Kubornetes监控操作员之前

在安装或升级之前、请阅读此信息 ["Kubbernetes监控操作员"](#)。

组件	要求
Kubernetes 版本	Kubbernetes v1.20及更高版本。
Kubbernetes分发	AWS EI生 性Kubelnetes服务(EKS) Azure Kubernetes Service （ AKS ） Google Kubernetes Engine （ GKEE ） Red Hat OpenShift Rancher Kubernetes Engine （ RKE） VMware Tanzu
Linux操作系统	Data Infrastructure Insight不支持运行ARM64架构的节点。网络监控：必须运行Linux内核版本4.18.0或更高版本。不支持光子操作系统。
标签	Data Infrastructure Insight支持监控运行Linux的Kubernetes节点、方法是指定一个Kubernetes节点选择器、用于在这些平台上查找以下Kubernetes标签： Kubernetes v1.20及更高版本：Kubernetes.io/OS = Linux Randcher + Catch.io作为流程编排/Kubernetes平台：Catch.io/OS = Linux
命令	必须提供URL和kubectl命令。；为获得最佳结果、请将这些命令添加到路径中。
连接	已将kubectl命令行界面配置为与目标K8s集群通信、并可通过Internet连接到Data Infrastructure Insight环境。如果您在安装期间使用代理、请按照操作员安装部分中的说明 "配置代理支持" 进行操作。为了获得准确的审核和数据报告、请使用网络时间协议(NTP)或简单网络时间协议(SNTP)同步Agent计算机上的时间。
其他	如果您运行的是OpenShift 4.6或更高版本、则必须遵循 "OpenShift 说明" 此外、还应确保满足这些前提条件。
API令牌	如果要重新部署Operator (即更新或替换Operator)、则无需创建新的API令牌；您可以重复使用先前的令牌。

开始之前需要注意的重要事项

如果您使用运行 [代理](#)、具有 [自定义存储库](#)或正在使用 [OpenShift](#)、请仔细阅读以下各节。

另请阅读相关内容 [权限](#)。

配置代理支持

要安装NetApp Kubernetes监控操作员、您可以在环境中的两个位置使用代理。这些代理系统可以是相同的、也可以是单独的：

- 在执行安装代码段(使用"curl")期间需要代理、以便将执行此代码段的系统连接到Data Infrastructure Insight环境
- 目标Kubernetes集群与Data Infrastructure Insight环境通信所需的代理

如果您对其中一个或这两个环境使用代理、则要安装NetApp Kubernetes操作监控器、必须先确保将代理配置为与Data Infrastructure Insight环境进行良好通信。例如、您需要能够从要安装Operator的服务器/VM访问Data Infrastructure Insight并从Data Infrastructure Insight下载二进制文件。

对于用于安装NetApp Kubernetes操作监控器的代理、在安装操作员之前、请设置_http_proxy/https_proxy_environment变量。对于某些代理环境、您可能还需要设置_no_proxy environment_变量。

要设置变量、请在您的系统上*在*安装NetApp Kubernetes监控操作员之前*执行以下步骤：

1. 为当前用户设置 `https_proxy` 和 / 或 `http_proxy` 环境变量：
 - a. 如果要设置的代理没有身份验证(用户名/密码)、请运行以下命令：

```
export https_proxy=<proxy_server>:<proxy_port>
.. 如果要设置的代理具有身份验证 (用户名/密码)、请运行以下命令：
```

```
export
http_proxy=<proxy_username>:<proxy_password>@<proxy_server>:<proxy_port>
```

要使您的Kubernetes集群所使用的代理与您的Data Infrastructure Insight环境进行通信、请在阅读所有这些说明后安装NetApp Kubernetes监控操作员。

在部署NetApp Kubernetes Monitoring Operator之前、请在operator-config.yaml中配置AgentConfiguration的代理部分。

```

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
  ...
  ...

```

使用自定义或专用**Docker**存储库

默认情况下、NetApp Kubernetes监控操作员将从Data Infrastructure Insight存储库中提取容器映像。如果您将某个Kubornetes集群用作监控目标、并且该集群配置为仅从自定义或私有Docker存储库或容器注册表中提取容器映像、则必须配置对NetApp Kubornetes监控操作员所需容器的访问权限。

从NetApp Monitoring Operator安装磁贴运行"Image Pull Snippet"。此命令将登录到Data Infrastructure Insight存储库、提取操作员的所有映像依赖关系、然后从Data Infrastructure Insight存储库中注销。出现提示时、输入提供的存储库临时密码。此命令可下载操作员使用的所有映像、包括可选功能的映像。请参见以下内容、了解这些图像用于哪些功能。

核心操作员功能和Kubornetes监控

- NetApp监控
- Kube-RBAC-代理
- Kube-state-metrics
- 电报
- distroless root用户

事件日志

- 流畅位
- Kubbernetes-event-exporter

网络性能和映射

- CI-net-observer

根据您的企业策略，将操作员 Docker 映像推送到您的私有 / 本地 / 企业 Docker 存储库。确保存储库中这些映像的映像标记和目录路径与Data Infrastructure Insight存储库中的映像标记和目录路径一致。

在operator-DEPLOYAML中编辑monitor-operator部署、并修改所有映像引用以使用私有Docker存储库。

```
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>
```

编辑operator-config.yaml中的AgentConfiguration以反映新的Docker repo位置。为私有存储库创建新的imagePullSecret,有关更多详细信息, 请参见<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 说明

如果您运行的是OpenShift 4.6或更高版本、则必须在_operator-config.yaml中编辑AgentConfiguration以启用_run特权_设置:

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

OpenShift可以实施更高的安全级别、从而可能阻止对某些Kubernetes组件的访问。

权限

如果您正在监控的集群包含自定义资源、而这些资源没有集群资源 "要查看的聚合", 则需要手动授予操作员对这些资源的访问权限, 才能使用事件日志对其进行监控。

1. 在安装之前或安装之后编辑_operator-additional-permissions. yaml. 编辑资源_ClusterRole
 - <namespace>-additional-permissions

2. 使用动词["GET、"Watch、"list"]为所需的每个组和资源创建一个新规则。参见<https://kubernetes.io/docs/reference/access-authn-authz/rbac/>
3. 将所做的更改应用于集群


Kubernetes Monitoring Operator安装和配置

Data Infrastructure Insight为Kubernetes集合提供了* Kubernetes Monitoring Operator*。导航到*Kubernetes >收集器>+Kubernetes Collector*以部署新操作员。

在安装Kubernetes Monitoring Operator之前

请参见 ["前提条件"](#) 安装或升级Kubornetes Monitoring Operator之前的文档。

安装Kubernetes Monitoring Operator

 **kubernetes**
Kubernetes

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

在KubeNet上安装Kubenetes Monitoring Operator代理的步骤：

1. 输入唯一的集群名称和命名空间。如果您是 [升级](#) 在先前的Kubnetes Operator中、请使用相同的集群名称和命名空间。
2. 输入这些代码后、您可以将Download Command代码录复制到剪贴板。
3. 将此代码片段粘贴到 `bash` 窗口中并执行。此时将下载Operator安装文件。请注意、此代码片段具有唯一的密钥、有效期为24小时。
4. 如果您有自定义或私有存储库、请复制可选的映像提取代码段、将其粘贴到 `_bash_shell` 中并执行该代码段。提取映像后、将其复制到您的私有存储库。请务必保持相同的标记和文件夹结构。更新 `_operator-DEPRAYAML_` 中的路径以及 `_operator-config.yaml_` 中的Docker存储库设置。
5. 如果需要、请查看可用的配置选项、例如代理或专用存储库设置。您可以阅读有关的更多信息 ["配置选项"](#)。
6. 准备好后、请通过复制kubec临时 应用的小程序来部署Operator、然后下载并执行该操作。
7. 安装将自动进行。完成后、单击 `_Next_` 按钮。
8. 安装完成后、单击 `_Next_` 按钮。同时、请务必删除或安全地存储 `_operator-秘密.yaml` 文件。

如果您使用的是代理、请阅读相关内容 [正在配置代理](#)。

如果您有自定义存储库、请阅读有关的信息 [使用自定义/私有Docker存储库](#)。

Kubernetes监控组件

Data Infrastructure Insight Kubernetes监控由四个监控组件组成：

- 集群指标
- 网络性能和映射(可选)
- 事件日志(可选)
- 变更分析(可选)

默认情况下、每个Kubernetes收集器都会启用上述可选组件；如果您确定某个特定收集器不需要某个组件、则可以通过导航到* Kubernetes > Collectors *并从屏幕右侧收集器的"三个点"菜单中选择_Modify Deployment _来禁用此组件。

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	1.101.0	Modify Deployment

此屏幕将显示每个组件的当前状态、并允许您根据需要为该收集器禁用或启用组件。

kubernetes

Kubernetes

Modify Deployment

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

升级

升级到最新的Kubernetes Monitoring Operator

确定现有Operator是否存在AgentConfiguration (如果您的命名空间不是默认的_netapp-monitoring_、请替换相应的命名空间):

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

如果存在AgentConfiguration:

- **安装** 现有运算符上的最新运算符。
 - 确保您的状态 [提取最新的容器映像](#) 如果使用的是自定义存储库。

如果AgentConfiguration不存在:

- 记下数据基础架构洞察力可识别的集群名称(如果您的命名空间不是默认的NetApp监控、请替换相应的命名空间):

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

* 为现有Operator创建备份 (如果您的命名空间不是默认的NetApp监控、请替换相应的命名空间):

```
kubectl -n netapp-monitoring get agent -o yaml > agent_backup.yaml
```

* <<to-remove-the-kubernetes-monitoring-operator,卸载>> 现有操作员。
* <<installing-the-kubernetes-monitoring-operator,安装>> 最新的运算符。

- 请使用相同的集群名称。
- 下载最新的Operator YAML文件后、在部署之前、将在agent_backup.yaml中找到的所有自定义设置移植到下载的operator-config.yaml。
- 确保您的状态 [提取最新的容器映像](#) 如果使用的是自定义存储库。

停止和启动Kubernetes Monitoring Operator

要停止Kubernetes Monitoring Operator:

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

要启动Kubernetes Monitoring Operator:

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

正在卸载

删除Kubernetes Monitoring Operator

请注意、Kubernetes Monitoring Operator的默认命名空间为"netp-monitoring"。如果您已设置自己的命名空间，请在这些命令和所有后续命令和文件中替换该命名空间。

可以使用以下命令卸载较新版本的监控操作员：

```
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>
```

如果监控操作员部署在自己的专用命名空间中、请删除此命名空间：

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

如果第一个命令返回"未找到资源"、请按照以下说明卸载旧版本的监控操作员。

按顺序执行以下每个命令。根据您当前的安装情况、其中某些命令可能会返回'object not found'消息。可以安全地忽略这些消息。

```
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>
```

如果以前创建了安全上下文约束：

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

关于Kube-state-metrics

NetApp Kubernetes监控操作员会安装自己的Kube-state-metrics、以避免与任何其他实例发生冲突。

有关Kube-State-Metrics的信息、请参见 ["此页面"](#)。

配置/自定义操作员

这些部分包含有关自定义操作员配置、使用代理、使用自定义或私有Docker存储库或使用OpenShift的信息。

配置选项

最常修改的设置可以在AgentConfiguration_自定义资源中进行配置。您可以通过编辑_operator-config.yaml文件来在部署操作员之前编辑此资源。此文件包含注释掉的设置示例。请参见列表 ["可用设置"](#) 对于最新版本的运算符。

您也可以在部署操作员后使用以下命令编辑此资源：

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

要确定您部署的操作员版本是否支持AgentConfiguration、请运行以下命令：

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

如果您看到“Error from server (NotFound)”消息，则必须先升级操作员，然后才能使用AgentConfiguration。

配置代理支持

您可以在环境中的两个位置使用代理来安装Kubernetes Monitoring Operator。这些代理系统可以是相同的、也可以是单独的：

- 在执行安装代码段(使用“cURL”)期间需要代理、以便将执行此代码段的系统连接到Data Infrastructure Insight环境
- 目标Kubernetes集群与Data Infrastructure Insight环境通信所需的代理

如果您对其中一个或这两个环境使用代理、则要安装Kubernetes Operating Monitor、必须首先确保您的代理已配置为能够与Data Infrastructure Insight环境进行良好的通信。如果您有一个代理、并且可以从要安装Operator的服务器/VM访问Data Infrastructure Insight、则您的代理可能已正确配置。

对于用于安装Kubernetes Operating Monitor的代理、在安装Operator之前、请设置_http_proxy/https_proxy_Environment变量。对于某些代理环境、您可能还需要设置_no_proxy environment_变量。

要设置变量，请在*安装Kubernetes Monitoring Operator之前*在系统上执行以下步骤：

1. 为当前用户设置 https_proxy 和 / 或 http_proxy 环境变量：
 - a. 如果要设置的代理没有身份验证(用户名/密码)、请运行以下命令：

```
export https_proxy=<proxy_server>:<proxy_port>
```

.. 如果要设置的代理具有身份验证(用户名/密码)、请运行以下命令：

```
export
http_proxy=<proxy_username>:<proxy_password>@<proxy_server>:<proxy_port>
```

要使Kubennetes集群所使用的代理与Data Infrastructure Insight环境进行通信、请在阅读所有这些说明后安装Kubennetes Monitoring Operator。

在部署Kubernetes Monitoring Operator之前、请在operator-config.yaml中配置AgentConfiguration的代理部分。

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

使用自定义或专用**Docker**存储库

默认情况下、Kubnetes监控操作员将从Data Infrastructure Insight存储库中提取容器映像。如果您将某个Kubornetes集群用作监控目标、并且该集群配置为仅从自定义或私有Docker存储库或容器注册表中提取容器映像、则必须配置对Kubornetes监控操作员所需容器的访问权限。

从NetApp Monitoring Operator安装磁贴运行"Image Pull Snippet"。此命令将登录到Data Infrastructure Insight存储库、提取操作员的所有映像依赖关系、然后从Data Infrastructure Insight存储库中注销。出现提示时、输入提供的存储库临时密码。此命令可下载操作员使用的所有映像、包括可选功能的映像。请参见以下内容、了解这些图像用于哪些功能。

核心操作员功能和Kubornetes监控

- NetApp监控
- CI-KKube-RBAC-代理

- CI-KSM
- CI-(国际通信
- distroless root用户

事件日志

- CI-流畅位
- CI-Kuber-netes-event-exporter

网络性能和映射

- CI-net-observer

根据您的企业策略，将操作员 Docker 映像推送到您的私有 / 本地 / 企业 Docker 存储库。确保存储库中这些映像的映像标记和目录路径与Data Infrastructure Insight存储库中的映像标记和目录路径一致。

在operator-DEPLOYAML中编辑monitor-operator部署、并修改所有映像引用以使用私有Docker存储库。

```
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>
```

编辑operator-config.yaml中的AgentConfiguration以反映新的Docker repo位置。为私有存储库创建新的imagePullSecret,有关更多详细信息，请参见<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 说明

如果您运行的是OpenShift 4.6或更高版本、则必须在_operator-config.yaml中编辑AgentConfiguration以启用_run特权_设置：

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

OpenShift可以实施更高的安全级别、从而可能阻止对某些Kubernetes组件的访问。

关于安全的注意事项

要删除Kubernetes Monitoring Operator在集群范围内查看机密的权限、请在安装之前从_operator-setup.yaml文件中删除以下资源：

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

如果是升级、请同时从集群中删除资源：

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

如果启用了"变更分析"、请修改_AgentConfiguration_或_operator-config.yaml_以取消注释change-management部分、并在change-management部分下包括_kindsToIgnoreFamWatch: "secnes"_.记下此行中单引号和双引号的存在和位置。

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

验证 Kubernetes 校验和

Data Infrastructure Insight代理安装程序会执行完整性检查、但某些用户可能希望在安装或应用下载的项目之前执行自己的验证。要执行仅下载操作（与默认的下载和安装操作相反），这些用户可以编辑从 UI 获取的代理安装命令并删除尾随的 "install" 选项。

请按照以下步骤操作：

1. 按照说明复制 Agent 安装程序代码片段。
2. 请将代码片段粘贴到文本编辑器中，而不是将其粘贴到命令窗口中。

3. 从命令中删除后缀"-install"。
4. 从文本编辑器复制整个命令。
5. 现在，将其粘贴到命令窗口（在工作目录中）并运行。

◦ Download and install （下载并安装）（默认）：

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

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

仅下载命令会将所有必需项目从Data Infrastructure Insight下载到工作目录。这些项目包括但不限于：

- 安装脚本
- 环境文件
- YAML 文件
- 签名校验和文件（SHA256.signed）
- 用于签名验证的 PEM 文件（netapp_cert.pem）

安装脚本，环境文件和 YAML 文件可以通过目视检查进行验证。

可以通过确认 PEM 文件的指纹为以下内容来验证 PEM 文件：

```
1A918038E8E127BB5C87A202DF173B97A05B4996  
更具体地说，
```

```
openssl x509 -fingerprint -sha1 -noout -inform pem -in netapp_cert.pem  
可以使用 PEM 文件验证签名校验和文件：
```

```
openssl smime -verify -in sha256.signed -CAfile netapp_cert.pem -purpose  
any  
在对所有项目进行满意的验证后，可以通过运行以下命令启动代理安装：
```

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

容差和污物

netapp-ci-tentlaf-ds_、netapp-ci-fluent-bit-ds_和_netapp-ci-net-oboder-L4-DS DemonSets必须在集群中的每个节点上计划一个POD、以便正确收集所有节点上的数据。操作器已配置为允许某些众所周知的*污染*。如果您在节点上配置了任何自定义污染、从而阻止Pod在每个节点上运行、则可以为这些污染创建一个*容错* "在_AgentConfiguration_中"。如果已将自定义污染应用于集群中的所有节点、则还必须向操作员部署添加必要的容错值、以便可以计划和执行操作员POD。

详细了解Kubbernetes "损害和公差"。

返回到 "NetApp Kubernetes监控操作员安装*页面"

故障排除

在设置Kubnetes Monitoring Operator时遇到问题时、请尝试以下操作：

问题：	请尝试以下操作：
我未看到 Kubernetes 永久性卷与相应后端存储设备之间的超链接 / 连接。我的 Kubernetes 永久性卷使用存储服务器的主机名进行配置。	按照以下步骤卸载现有的 Telegraf 代理，然后重新安装最新的 Telegraf 代理。您必须使用Telegraf 2.0或更高版本、并且Data Infrastructure Insight必须主动监控Kubernetes集群存储。

<p>问题：</p> <p>我在日志中看到如下消息：</p> <pre>E0901 15: 21: 39.962145 1 refinder.go: 178] K8s.io/Kube-state-metrics/Internal、store/Builder : 352: 无法列出* 。v1.MutatingWebhankConfiguration: 服务器找不到 请求的资源 E0901 15: 21: 43.168161 1反射器.GO: 178] K8s.io/Kube-state-metrics/Internal / store/Builder : 352: 无法列出* v1.Lease: 服务器找不到请求的资 源(get leased.co其中.k8s.io) 等</pre>	<p>请尝试以下操作：</p> <p>如果您运行的是Kube-state-metrics版本2.0.0或更高版本、而Kubernetes版本低于1.20、则可能会出现这些消息。</p> <p>要获取Kubnetes版本、请执行以下操作：</p> <p><i>kubect</i>版本</p> <p>要获取Kube-state-metrics版本、请执行以下操作：</p> <pre>kubectl get Deploy /kube-state-metrics -o jsonpath='{..image }'</pre> <p>为了防止出现这些消息、用户可以修改其Kube-state-metrics部署以禁用以下租约：</p> <pre>mutatingwebbankconfigurations validatingwebbankconfiguration volumeAttachments资源</pre> <p>更具体地说、他们可以使用以下命令行界面参数：</p> <p>资源=验证签名请求、配置映射、cronjobs、守护程序、部署、端点、水平脚本自动扩展程序、安装、作业、限制范围、空间、网络策略、节点、持久卷、持久性预算、Pod、复制集、复制控制器、资源均衡、机密、服务、状态集、存储空间</p> <p>默认资源列表为：</p> <p>"验证签名请求、配置映射、cronjobs、守护程序、部署、端点、水平podAutocalers、安装、作业、租用、限制范围、mutatingwebhankconfigurations、名目、网络策略、节点、持久性卷、复制卷、podrisbudation髡器、Pod、资源集、状态控制器、存储、密钥、服务、持久性 验证webfokconfigurations,卷附件"</p>
---	---

问题：	请尝试以下操作：
<p>我看到来自Telegraf的错误消息如下所示、但Telegraf确实启动并运行：</p> <p>10月11日14: 23: 41 IP-172-31-39-47 systemd[1]: 启动插件驱动型服务器代理、以便向InfluxDB报告指标。</p> <p>Oct 11 14 : 23 : 41 IP-172-31-39-47 cailaf[1827] : time="2021-10-11T14 : 23 : 41Z" level=error msg="failed to create cache directory" 。 /etc/tedlaf/.cache/snowlkip、错误: mkdir /etc/tedlaf/.ca</p> <p>CHE: 权限被拒绝。已忽略\n" func="gosnowscap. (*DEPLOADER).Errorf" file="log.go:120"</p> <p>10月11日14: 23: 41 IP-172-31-39-47 TELABE[1827] : time="2021-10-11T14: 23: 41Z" level = error msg="failed to open.已忽略。打开/etc/德拉夫/.cache/snowspace/ocsp_response_cache.json : no s此 选项文件或目录\n" func="gosnowscap. (*DEPORTLogger).Errorf" file="log.go:120"</p> <p>10月11日14: 23: 41 IP-172-31-39-47、特拉夫[1827] : 2021-10-11T14: 23: 41Z ! 启动 Telegraf 1.19.3</p>	<p>这是一个已知的问题描述。请参见 "此 GitHub 文章" 有关详细信息：只要 Telegraf 启动并运行，用户就可以忽略这些错误消息。</p>
<p>在Kubelnetes上、我的Telegraf Pod报告以下错误： "处理mountstats信息时出错：无法打开mountstats文件：/hostfs/proc/1/mountstats、错误：打 开/hostfs/proc/1/mountstats：权限被拒绝"</p>	<p>如果启用并强制实施SELinux、则可能会阻止Telegraf Pod访问Kubelnetes节点上的/proc/1/mountstats文件。要克服此限制、请编辑代理配置并启用run特权 设置。有关详细信息、请参见 "OpenShift 说明"。</p>
<p>在Kubelnetes上、我的Telegraf ReporticaSet Pod报告以下错误：</p> <p>[inputs.prometheus]插件错误：无法加载密钥对/etc/Kubernetes/pki/etcD/server.crt ： /etc/Kubernetes/pki/etcD/server.key：打 开/etc/Kubernetes/pki/etcD/server.crt：无此文件或目录</p>	<p>Telegraf ReplicaSet Pod 应在指定为主节点或 etcd 节点上运行。如果 ReplicaSet Pod 未在其中一个节点上运行，您将收到这些错误。检查您的主 /etcd 节点是否具有此类节点的影响。如果是，请将必要的容错添加到 Telegraf ReplicaSet ，即 Teleaf-RS 中。</p> <p>例如、编辑"System..."</p> <p>kubect-rs edit rs德拉夫-rs</p> <p>...并将适当的容差添加到规范中。然后，重新启动 ReplicaSet Pod 。</p>

问题：	请尝试以下操作：
我使用的是PSP/PSA环境。这是否会影响我的监控操作员？	<p>如果您的Kubernetes集群运行的是Pod安全策略(PSP)或Pod安全准入(PSA)、则必须升级到最新的Kubernetes Monitoring Operator。按照以下步骤升级到支持PSP/PSA的当前Operator：</p> <ol style="list-style-type: none"> 1. 卸载 上一个监控操作员： <pre>kubect delete agent agent-monitor-NetApp -n NetApp-monitoring kubect-delete ns ns-monitoring kubec联系 删除crd agents.monitoring.netapp.com kubect-delete集群角色agent-manager-Role agent-proxy-Role agent-metrics-reader kubeca delete cluster rolebingagent-manager-rolebingagent-proxy-rolebingagent-cluster-admin-rolebing</pre> <ol style="list-style-type: none"> 2. 安装 监控运算符的最新版本。
我在尝试部署操作员时遇到问题、并且我正在使用PSP/PSA。	<ol style="list-style-type: none"> 1.使用以下命令编辑代理： <pre>kubect -n <name-space>编辑代理</pre> <ol style="list-style-type: none"> 2.将"security-policy-enenabled "标记为"false"。这将禁用Pod安全策略和Pod安全准入、并允许操作员进行部署。使用以下命令进行确认： <pre>kubecol get PSP (应显示Pod安全策略已删除) kubect get all -n <namespace></pre>
grep -i psp (应显示未找到任何内容)	出现"ImagePullBackoff"错误
如果您具有自定义或专用Docker存储库、但尚未将Kubernetes Monitoring Operator配置为正确识别它、则可能会出现这些错误。 阅读更多内容 关于为自定义/私有repo.	我正在部署监控操作员问题描述 、而当前文档对我的解决没有帮助。
捕获或记下以下命令的输出、然后联系技术支持团队。	Operator命名空间中的Net-Observer (Workload Map) Pod位于CrashLoopBackOff中
<pre>kubectl -n netapp-monitoring get all kubectl -n netapp-monitoring describe all kubectl -n netapp-monitoring logs <monitoring-operator-pod> --all -containers=true kubectl -n netapp-monitoring logs <telegraf-pod> --all -containers=true</pre>	

问题：	请尝试以下操作：
<p>这些Pod对应于用于网络可观察性的工作负载映射数据收集器。请尝试以下操作：</p> <ul style="list-style-type: none"> •检查其中一个Pod的日志以确认最低内核版本。例如： <pre>{ "ci租户id": "Your -en租 户id", "c收集 器集群" : "Your -K8s-cluster-name", "문 凭": "prod", "level" : "error", "msg": "验证失败。原因：内核版本3.10.0 低于最低内核版本4.18.0", "time": "2022-11-09T08 : 23: 08Z" } —</pre> •Net-observer Pod要求Linux内核版本至少为4.18.0。使用命令"uname -r "检查内核版本、并确保它们>= 4.18.0 	Pod正在Operator命名空间中运行(默认值：netapo-monitoring)、但在查询中、UI中不会显示工作负载映射或KubeNet指标的任何数据
检查K8S集群节点上的时间设置。为了准确地进行审核和数据报告、强烈建议使用网络时间协议(NTP)或简单网络时间协议(SNTP)同步Agent计算机上的时间。	Operator命名空间中的某些Net-observer Pod处于Pending状态
<p>Net-observer是一个DemonSet、在K8s集群的每个节点上运行一个POD。</p> <ul style="list-style-type: none"> •记下处于“待定”状态的POD，并检查它是否遇到了CPU或内存的资源问题描述。确保节点中具有所需的内存和CPU。 	<p>安装Kubenetes Monitoring Operator后、我的日志中立即显示以下内容：</p> <p>[inputs.prometheus]插件错误：向\tcp.tcp.svc.cluster-local http://kube-state-metrics:8080/metrics发出请求时出错：get \tcp.tcp.svc.cluster-local:8080/metrics http://kube-state-metrics: 拨号<namespace><namespace>: LOOKUP Kupe-state-metrics.tcp.svc.cluster-local <namespace>: 无此主机</p>
通常、只有在安装了新操作员且_craaf-RS_ POD在_KSM_ POD启动之前启动时、才会显示此消息。所有Pod运行后、这些消息应停止。	我没有看到为集群中的Kubnetes CronJobs收集任何指标。
验证您的Kubbernetes版本(即 kubectl version)。如果是v1.20.x或更低版本、则这是预期的限制。随Kubernetes Monitoring Operator部署的Kube-state-metrics版本仅支持v1.cronjob. 对于Kubernetes 1.2.x及更低版本、cronJob资源位于v1beta.cronJob。因此、Kube-state-metrics找不到cronJob资源。	安装操作员后、该特拉夫DS Pod进入CrashLoopBackOff、并且POD日志指示"su : authentication failure"(su: 身份验证失败)。
<p>编辑_AgentConfiguration_中的"特拉夫"部分、并将_dockerMetricCollectionEnabled"设置为false。有关详细信息，请参阅操作员的"配置选项"。注意：如果您使用的是Data Infrastructure Insight Federal Edition、则对_su_的使用有限制的用户将无法收集Docker指标、因为访问Docker套接字需要以root用户身份运行该特拉夫容器或使用_su_将该特拉夫用户添加到Docker组。默认情况下、Docker指标收集和_su_的使用处于启用状态；要同时禁用这两者、请删除_AgentConfiguration_文件中的_tenderaf.Docker条目：... spec: ... tenderaf: ... -名称 : Docker run-mode: - DemonSet 替换项: -关键字: Docker_UNIS_sdoc_s占位符 值: UNIX: ///run/Docker。sk.....</p>	<p>我在Telegraf日志中看到重复出现以下错误消息：</p> <p>E! [agent]写入至Outputs.http: POST "\https : //lace/rest/v1/lace/ingest/影响xdb"时出错：超过上下文截止时间(<tenant_url>。等待标头时超时)</p>

问题：	请尝试以下操作：
编辑_AgentConfiguration_中的"特拉夫"部分、并将_outputTimeout_增加到10秒。有关详细信息、请参见操作员的 "配置选项" 。	我缺少一些事件日志的_volvedobject_数据。
确保已按照中的步骤进行操作 "权限" 第节。	为什么我看到两个监控操作员Pod正在运行、一个名为NetApp-CI-monitoring operator-Pod <pod>、另一个名为monitoring operator-Pod? <pod>
自2023年10月12日起、Data Infrastructure Insight对运营者进行了重构、以更好地为用户服务；要完全采用这些变更 删除旧运算符 、您必须和 安装新的 。	我的Kubbernetes事件意外停止向Data Infrastructure Insight报告。
检索事件导出器Pod的名称：	grep event-exporter
<pre>`kubectl -n netapp-monitoring get pods</pre>	
awk '{print \$1}'	<p>sed 's/event-exporter./event-exporter/' 此名称应为"NetApp-CI-event-exporter "或"event-exporter。接下来、编辑监控代理 kubectl -n netapp-monitoring edit agent，然后设置log_file的值，以反映在上一步中找到的相应事件导出器POD名称。更具体地说、log_file应设置为"/var/log/containers/NetApp-CI-event-exporter .log"或"/var/log/containers/event-exporter 。log"</p> <pre>.... fluent-bit: ... - name: event-exporter-ci substitutions: - key: LOG_FILE values: - /var/log/containers/netapp-ci-event-exporter.log</pre> <p>或者、也可以这样做 卸载 和 重新安装 代理。</p>
我发现Kubenetes监控操作员部署的POD因资源不足而崩溃。	请参见Kubbernetes监控操作员 "配置选项" 根据需要增加CPU和/或内存限制。
缺少映像或配置无效会导致NetApp-CI-Kube-state-metrics Pod无法启动或准备就绪。现在、StatefulSet停止运行、并且配置更改未应用于NetApp-CI-KUE-STATE-MERics Pod。	StatefulSet位于中 "已损坏" 状态。修复任何配置问题后、退回NetApp-CI-Kube-state-metrics Pod。
运行Kubelnetes Operator升级后、netapo-CI-Kube-state-metrics Pod无法启动、引发ErrImagePull (无法提取映像)。	尝试手动重置Pod。

问题：	请尝试以下操作：
在日志分析下、我的Kubernetes集群显示"Event Discarded as older then maxEventAgeSonds"消息。	修改Operator <i>agentconfiguration_</i> 并将 <i>_event-exporter maxEventAgeSonds</i> (例如、60秒)、 <i>event-exporter kubeQPS</i> (例如、100)和 <i>_event-exporter kubeBurst_</i> (例如、500)增加到。有关这些配置选项的更多详细信息、请参见 "配置选项" 页面。
Telegraf会发出警告、指出可锁定内存不足或崩溃。	尝试增加底层操作系统/节点中Telegraf可锁定内存的限制。如果不能增加限制、请修改nLMO代理配置并将 <i>_Unproted_</i> 设置为 <i>_true</i> 。这将指示Telegraf不尝试预留锁定的内存页。由于解密的机密可能会交换到磁盘、因此这可能会带来安全风险、但它允许在无法预留锁定内存的环境中执行。有关 <i>_UnprotECE_</i> 配置选项的更多详细信息、请参阅 "配置选项" 页面。
我看到Telegraf发出的警告消息如下所示： <i>W ! [Inputs.diskio]无法收集"vdc"的磁盘名称：读取/dev/vdc时出错：没有此文件或目录</i>	对于Kubnetes监控操作员、这些警告消息不会产生负面影响、可以放心地忽略。 或者、也可以编辑AgentConfiguration中的"tendraf"部分、并将 <i>_runDs</i> 专用_ 设置为true。有关详细信息，请参阅 "操作员配置选项" 。

可以从找到追加信息 ["支持"](#) 页面或中的 ["数据收集器支持列表"](#)。

Kubnetes监控操作员配置选项

。 ["Kubbernetes监控操作员"](#) 配置可以自定义。

下表列出了 *_AgentConfiguration_* 文件的可能选项：

组件	选项	说明
代理		操作员可以安装的所有组件通用的配置选项。这些选项可视为"全局"选项。
	文档报告员报告	与Data Infrastructure Insight Docker repo相比、使用dockerRepo覆盖从客户的私有Docker reposo中提取映像。默认值为Data Infrastructure Insight Docker repo
	dockerImagePullSecret	可选：客户专用repo的机密
	clusterName	自由文本字段、用于在所有客户集群中唯一标识集群。这在Data Infrastructure Insight租户中应该是唯一的。默认值是客户在UI中为"Cluster Name"字段输入的内容

组件	选项	说明
	代理 格式。 代理： 服务器： 端口： 用户名： 密码： NoProxy： 已启用isTelegrafProxy： 已启用isAuProxy： isFluentbitProxyEnabled： 已启用isCollectorProxy：	可选设置代理。这通常是客户的公司代理。
电报		配置选项、可自定义操作员的安装
	seccionInterval	指标收集间隔(以秒为单位)(最大= 60秒)
	dsCpuLimit	用于数据终端的CPU限制
	dsMemLimit	用于数据的存储器限制
	dsCpuRequest	为数据发送的CPU请求
	dsMemRequest	为数据发送的内存请求
	rsCpuLimit	用于RS的CPU限制
	rsMemLimit	用于Rs的存储器限制
	rsCpuRequest	对RS的CPU请求
	rsMemRequest	对RS的存储器请求
	run特权	在特权模式下运行特拉特夫DemonSet的_table-mountstats-poller_容器。如果在Kubnetes节点上启用了SELinux、请将此选项设置为true。
	批大小	请参见 "Telegraf配置文档"
	缓冲区限制	请参见 "Telegraf配置文档"
	RoundInterval	请参见 "Telegraf配置文档"
	Jitter	请参见 "Telegraf配置文档"
	精确度	请参见 "Telegraf配置文档"
	FlushInterval	请参见 "Telegraf配置文档"
	Flush抖动	请参见 "Telegraf配置文档"
	输出超时	请参见 "Telegraf配置文档"
	ds.公差	Telegraf-DS的额外耐受性。
	Rs公差	Telegraf-RS额外的耐受性。

组件	选项	说明
	skipProcessorsAfterRegat ors	请参见 "Telegraf配置文档"
	未受保护	请参见此内容 "已知电报问题描述" 。设置_UNprotECE_将指示Kubelnetes监控操作员使用运行Telegraf --unprotected 标志。
Kube-state-metrics		可自定义操作员的KUbe状态指标安装的配置选项
	cpuLimit	Kube-state-metrics部署的CPU限制
	memLimit	Kube-state-metrics部署的MEM限制
	cpuQuest	KU状态指标部署的CPU请求
	MemQuest	MEM请求部署KIBEstimate metrics
	资源	要捕获的资源的逗号分隔列表。示例：cronjobs、守护程序集、部署、服务器、作业、名称空间、节点、persistentvolumeclaims、persistentvolumes、Pod、复制集、资源集合、服务、状态集
	公差	Kube-state-metrics的其他容错性。
	labels	应捕获的Kube-state-metrics资源的逗号分隔列表 + 示例：cronjobs=[*]、守护程序sets=[*]、部署=[*]、ingresses=[*]、Jobs=[*]、名称空间=[*]、节点=[*]、persistentvolumeclaims=[*]，persistentvolumes=[*]，Pod=[*]，复制集=[*]，资源公平otas=[*]，services=[*]，状态集=[*] +
日志		可自定义操作员日志收集和安装的配置选项
	readFromHead	true或false、则应以流畅位从head读取日志
	超时	超时(以秒为单位)
	dnsMode	TCP/UDP、DNS的模式
	流畅的位容差	Fluent-Bit-DS的额外容差。
	事件-导出器-容错	事件导出器附加容错。
	Event- exporter—maxEventAgeS econds	事件导出器最大事件期限。 请参见 https://github.com/jkroepke/resmoio-kubernetes-event-exporter
	run特权	如果Fluent Bit无法启动并尝试打开/创建其数据库、请将run特权 设置为true。
	运行特权	将runDs专用 设置为true可在特权模式下运行特拉特拉夫DemonSet的特拉特拉特容器。
工作负载映射		可自定义Operator的工作负载映射收集和安装的配置选项。
	cpuLimit	Net observer DS的CPU限制
	memLimit	净观察者DS的MEM限制

组件	选项	说明
	cpuQuest	Net observer DS的CPU请求
	MemQuest	MEM请求提供Net observer DS
	metricRegationInterval	指标聚合间隔(以秒为单位)
	bpfPolollInterval	BPF轮询间隔(以秒为单位)
	启用DNSLook.e.	是非题、启用DNS查找
	L4-公差	Net-obler-L4-DS附加容错。
	run特权	true或false—如果在Kubbernetes节点上启用了SELinux、则将run特权 设置为true。
变更管理		Kubnetes变更管理和分析的配置选项
	cpuLimit	change-ob맛 달 꺼-Watch RS的CPU限制
	memLimit	change-ob맛 달 꺼-Watch RS的MEM限制
	cpuQuest	对change-ob맛 달 꺼-手表-rs的CPU请求
	MemQuest	MEM请求change-ob맛 달 꺼-Watch RS
	failureMins	工作负载部署失败后将标记为失败的间隔(以分钟为单位)
	部署聚合IntervalSeconds	发送正在进行的工作负载部署事件的频率
	nonWorkloadAggrIntervalSeconds	合并和发送非工作负载部署的频率
	TermsToRedact	在env名称和数据映射中使用的一组正则表达式、其值将被编辑 示例术语: "pwd"、"password"、"令牌"、"APIkey"、"API-key"、"jwt"
	其他KindsToWatch	收集器监控的一组默认类型中要监控的其他类型的逗号分隔列表
	kindsToIgnoreFromWatch	收集器监控的一组默认类型中要忽略的监视类型的逗号分隔列表
	LogRecordAggrIntervalSeconds	从收集器向CI发送日志记录的频率
	手表耐受性	change-ob맛 달 꺼-Watch—DS的额外容差。仅限简写单行格式。 示例: " {key: tint1、operator: exists、effect : NoSchedule} 、 {key: tint2、operator: exists 、effect: NoExecute} "

AgentConfiguration文件示例

以下是_AgentConfiguration_文件示例。

```
apiVersion: monitoring.netapp.com/v1alpha1
kind: AgentConfiguration
```

```

metadata:
  name: netapp-ci-monitoring-configuration
  namespace: "netapp-monitoring"
  labels:
    installed-by: nkmo-netapp-monitoring

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

    # # Proxy settings. The proxy that the operator should use to send
    # # metrics to Data Infrastructure Insights.
    # # Please see documentation here: https://docs.netapp.com/us-en/cloudinsights/task\_config\_telegraf\_agent\_k8s.html#configuring-proxy-support
    # proxy:
    #   server:
    #   port:
    #   noproxy:
    #   username:
    #   password:
    #   isTelegrafProxyEnabled:
    #   isFluentbitProxyEnabled:
    #   isCollectorsProxyEnabled:

    # # [Required Field] By default, the operator uses the CI repository.
    # # To use a private repository, change this field to your repository
    # # name.
    # # Please see documentation here: https://docs.netapp.com/us-en/cloudinsights/task\_config\_telegraf\_agent\_k8s.html#using-a-custom-or-private-docker-repository
    dockerRepo: 'docker.c01.cloudinsights.netapp.com'
    # # [Required Field] The name of the imagePullSecret for dockerRepo.
    # # If you are using a private repository, change this field from
    # # 'netapp-ci-docker' to the name of your secret.
    dockerImagePullSecret: 'netapp-ci-docker'

```

```

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

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

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

# # telegraf-ds CPU/Mem limits and requests.
# # See https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

```

```

# dsCpuLimit: '750m'
# dsMemLimit: '800Mi'
# dsCpuRequest: '100m'
# dsMemRequest: '500Mi'

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

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

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

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

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

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

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

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

```

```

metrics.
    # managedK8sSystemMetricCollectionEnabled: 'false'

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

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

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

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

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

# # Comma-separated list of metrics to enable.
# # See metric-allowlist in https://github.com/kubernetes/kube-state-
metrics/blob/main/docs/cli-arguments.md
# metrics:
'kube_cronjob_created,kube_cronjob_status_active,kube_cronjob_labels,kube_daemonset_created,kube_daemonset_status_current_number_scheduled,kube_daemonset_status_desired_number_scheduled,kube_daemonset_status_number_available,kube_daemonset_status_number_misscheduled,kube_daemonset_status_number_ready,kube_daemonset_status_number_unavailable,kube_daemonset_status_observed_generation,kube_daemonset_status_updated_number_scheduled,kube_daemonset_metadata_generation,kube_daemonset_labels,kube_deployment_status_replicas,kube_deployment_status_replicas_available,kube_deployment_status_replicas_unavailable,kube_deployment_status_replicas_updated,kube_deployment_status_observed_generation,kube_deployment_spec_replicas,kube_deployment_spec_paused,kube_deployment_spec_strategy_rollingupdate_max_unavailable,kube_d

```

deployment_spec_strategy_rollingupdate_max_surge,kube_deployment_metadata_generation,kube_deployment_labels,kube_deployment_created,kube_job_created,kube_job_owner,kube_job_status_active,kube_job_status_succeeded,kube_job_status_failed,kube_job_labels,kube_job_status_start_time,kube_job_status_completion_time,kube_namespace_created,kube_namespace_labels,kube_namespace_status_phase,kube_node_info,kube_node_labels,kube_node_role,kube_node_spec_unschedulable,kube_node_created,kube_persistentvolume_capacity_bytes,kube_persistentvolume_status_phase,kube_persistentvolume_labels,kube_persistentvolume_info,kube_persistentvolume_claim_ref,kube_persistentvolumeclaim_access_mode,kube_persistentvolumeclaim_info,kube_persistentvolumeclaim_labels,kube_persistentvolumeclaim_resource_requests_storage_bytes,kube_persistentvolumeclaim_status_phase,kube_pod_info,kube_pod_start_time,kube_pod_completion_time,kube_pod_owner,kube_pod_labels,kube_pod_status_phase,kube_pod_status_ready,kube_pod_status_scheduled,kube_pod_container_info,kube_pod_container_status_waiting,kube_pod_container_status_waiting_reason,kube_pod_container_status_running,kube_pod_container_state_started,kube_pod_container_status_terminated,kube_pod_container_status_terminated_reason,kube_pod_container_status_last_terminated_reason,kube_pod_container_status_ready,kube_pod_container_status_restarts_total,kube_pod_overhead_cpu_cores,kube_pod_overhead_memory_bytes,kube_pod_created,kube_pod_deletion_timestamp,kube_pod_init_container_info,kube_pod_init_container_status_waiting,kube_pod_init_container_status_waiting_reason,kube_pod_init_container_status_running,kube_pod_init_container_status_terminated,kube_pod_init_container_status_terminated_reason,kube_pod_init_container_status_last_terminated_reason,kube_pod_init_container_status_ready,kube_pod_init_container_status_restarts_total,kube_pod_status_scheduled_time,kube_pod_status_unschedulable,kube_pod_spec_volumes_persistentvolumeclaims_readonly,kube_pod_container_resource_requests_cpu_cores,kube_pod_container_resource_requests_memory_bytes,kube_pod_container_resource_requests_storage_bytes,kube_pod_container_resource_requests_ephemeral_storage_bytes,kube_pod_container_resource_limits_cpu_cores,kube_pod_container_resource_limits_memory_bytes,kube_pod_container_resource_limits_storage_bytes,kube_pod_init_container_resource_limits_cpu_cores,kube_pod_init_container_resource_limits_memory_bytes,kube_pod_init_container_resource_limits_storage_bytes,kube_pod_init_container_resource_limits_ephemeral_storage_bytes,kube_pod_init_container_resource_requests_cpu_cores,kube_pod_init_container_resource_requests_memory_bytes,kube_pod_init_container_resource_requests_storage_bytes,kube_pod_init_container_resource_requests_ephemeral_storage_bytes,kube_replicaset_status_replicas,kube_replicaset_status_ready_replicas,kube_replicaset_status_observed_generation,kube_replicaset_spec_replicas,kube_replicaset_metadata_generation,kube_replicaset_labels,kube_replicaset_created,kube_replicaset_owner,kube_resourcequota,kube_resourcequota_created,kube_service_info,kube_service_labels,kube_service_created,kube_service_spec_type,kube_statefulset_status_replicas,kube_statefulset_status_replicas_current,kube_statefulset_status_replicas_ready,kube_statefulset_status_replicas_updated,kube_statefulset_status_observed_generation,


```
kube_statefulset_replicas,kube_statefulset_metadata_generation,kube_statefulset_created,kube_statefulset_labels,kube_statefulset_status_current_revision,kube_statefulset_status_update_revision,kube_node_status_capacity,kube_node_status_allocatable,kube_node_status_condition,kube_pod_container_resource_requests,kube_pod_container_resource_limits,kube_pod_init_container_resource_limits,kube_pod_init_container_resource_requests'
```

```
# # Comma-separated list of Kubernetes label keys that will be used in the resources' labels metric.
```

```
# # See metric-labels-allowlist in https://github.com/kubernetes/kube-state-metrics/blob/main/docs/cli-arguments.md
```

```
# labels:
```

```
'cronjobs=[*],daemonsets=[*],deployments=[*],ingresses=[*],jobs=[*],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: ''
```

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

```
# shards: '2'
```

```
# # Settings for the Events Log feature.
```

```
# logs:
```

```
# # Set runPrivileged to true if Fluent Bit fails to start, trying to open/create its database.
```

```
# runPrivileged: 'false'
```

```
# # If Fluent Bit should read new files from the head, not tail.
```

```
# # See Read_from_Head in
```

```
https://docs.fluentbit.io/manual/pipeline/inputs/tail
```

```
# readFromHead: "true"
```

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

```
# dnsMode: "UDP"
```

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

```
# fluentBitDNSResolver: "LEGACY"
```

```
# # Logs additional tolerations. Use the following abbreviated single
```

```

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

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

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

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

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

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

```

```

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

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

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

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

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

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

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

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

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

# # A set of regular expressions used in env names and data maps whose
value will be redacted
# termsToRedact: '"pwd", "password", "token", "apikey", "api-key",
"api_key", "jwt", "accesskey", "access_key", "access-key", "ca-file",

```

```

"key-file", "cert", "cafile", "keyfile", "tls", "crt", "salt",
".dockerconfigjson", "auth", "secret"

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

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

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

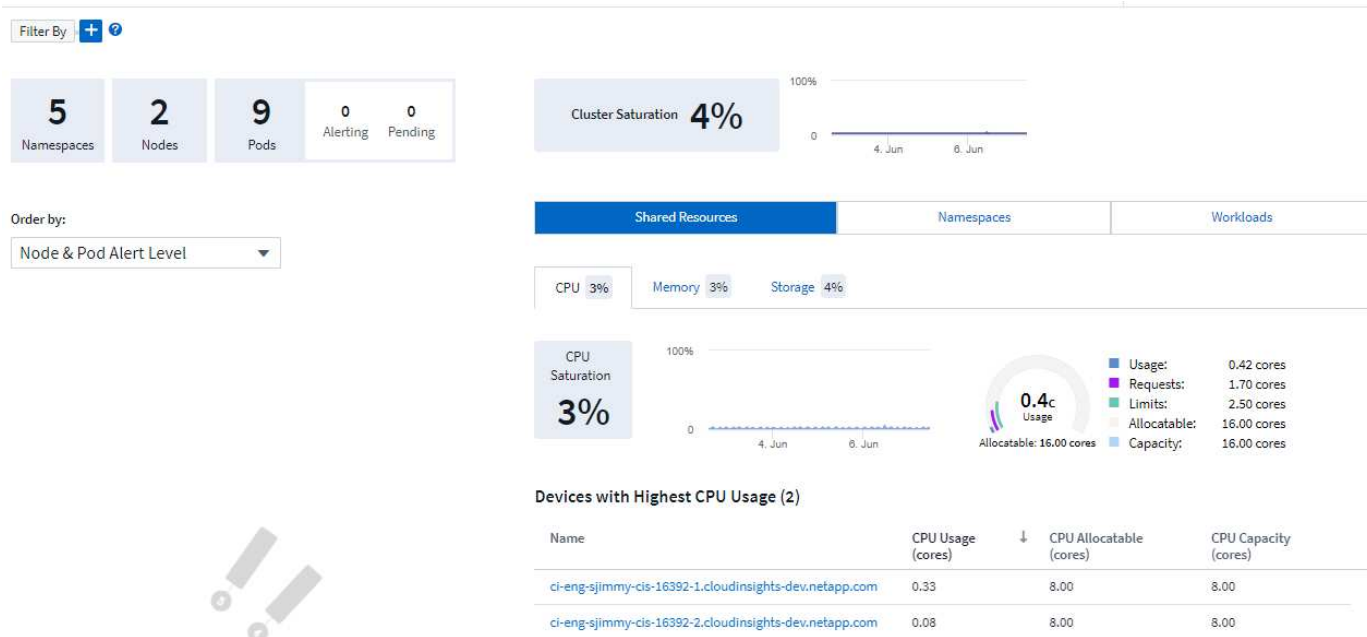
# # Frequency with which log records are sent to CI from the collector
# logRecordAggrIntervalSeconds: '20'

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

```

Kubernetes 集群详细信息页面

Kubernetes 集群详细信息页面显示了 Kubernetes 集群的详细概述。



命名空间、节点和Pod计数

页面顶部的计数显示了集群中的命名空间、节点和Pod总数、以及当前正在发出警报且处于待定状态的弹出消息数。

共享资源和饱和

详细信息页面右上角是您的集群饱和当前百分比以及显示一段时间内的最新趋势的图形。集群饱和是指每个时间点的最高CPU、内存或存储饱和。

在此下方、页面默认显示*共享资源*使用情况、并提供CPU、内存和存储选项卡。每个选项卡都会显示一段时间内的饱和百分比和趋势、以及其他使用情况详细信息。对于存储、显示的值是后端和文件系统饱和的较大值、而这两个值是单独计算的。

使用率最高的设备显示在底部的表中。单击任何链接以浏览这些设备。

命名空间

命名空间选项卡显示Kubernetes环境中所有命名空间的列表、其中显示了CPU和内存使用情况以及每个命名空间中的工作负载计数。单击名称链接以浏览每个命名空间。

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

工作负载

同样、工作负载选项卡会显示每个命名空间中的工作负载列表、再次显示CPU和内存使用情况。单击命名空间链接可深入了解每个。

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

集群 " 车轮 "



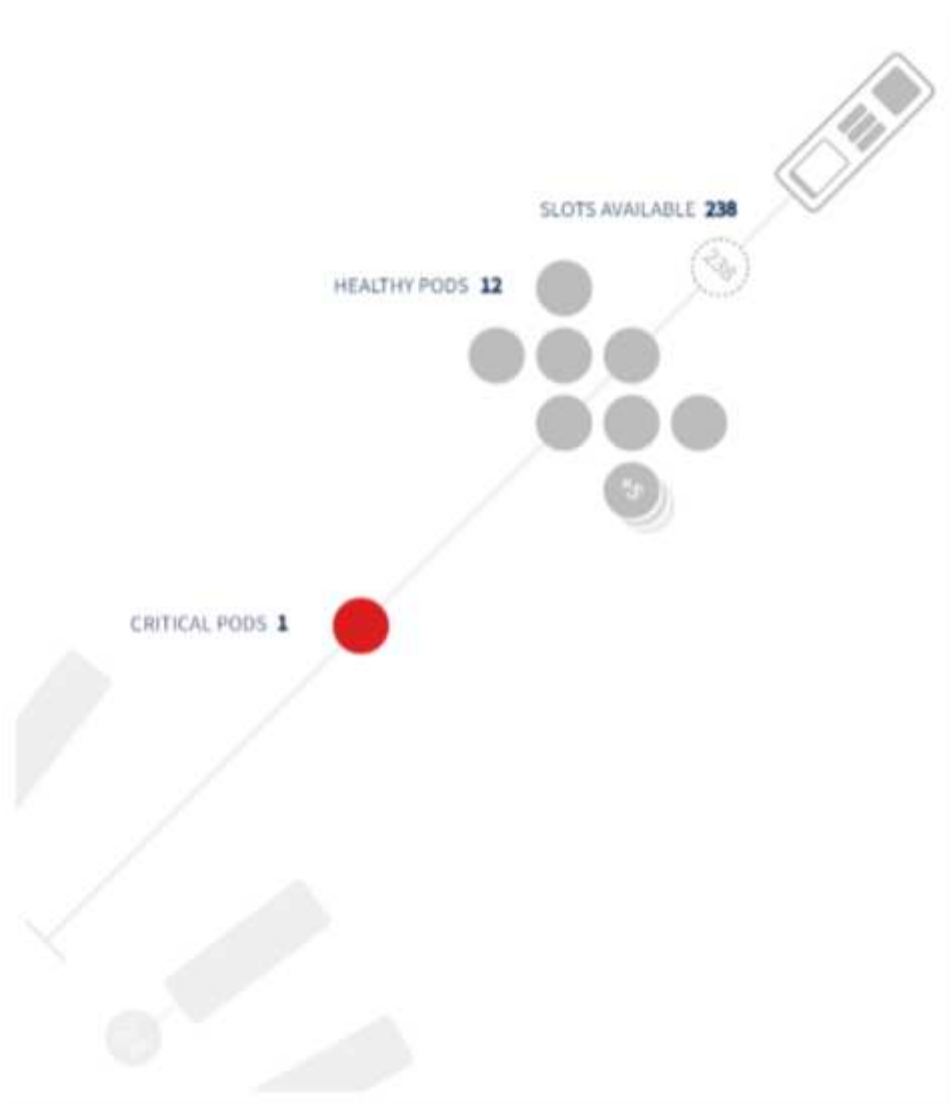
集群 " 车轮 " 部分简要介绍了节点和 POD 运行状况，您可以深入了解这些信息。如果集群包含的节点数超过页面此区域中显示的节点数，您可以使用可用按钮转动车轮。

警报 Pod 或节点以红色显示。" 警告 " 区域显示为橙色。未计划（即未连接）的 Pod 将显示在集群 " 车轮 " 的下角。

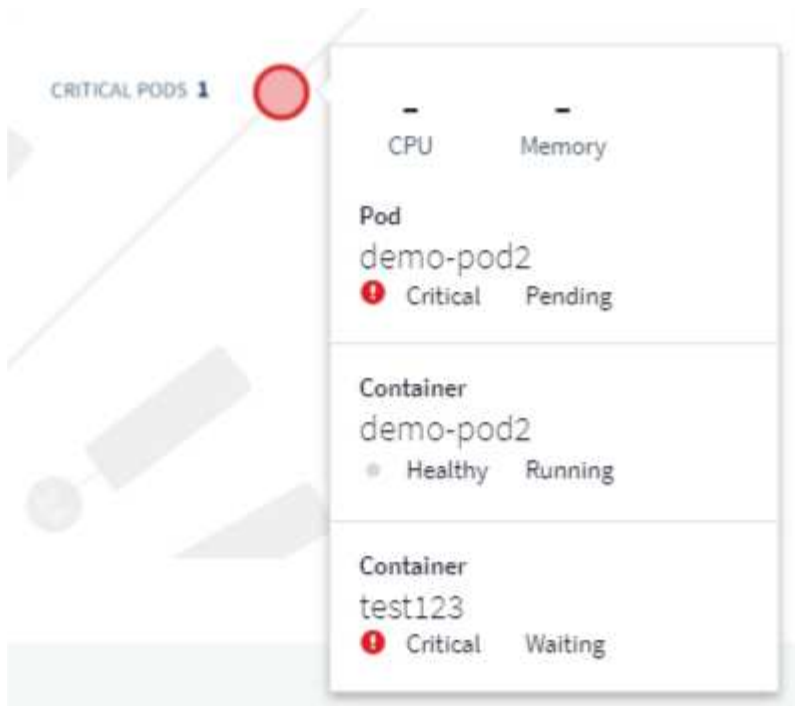
将鼠标悬停在 Pod （圆形）或 Node （条形）上可扩展节点视图。



单击该视图中的 Pod 或节点将放大到展开的节点视图。



从此处，您可以将鼠标悬停在某个元素上以显示有关该元素的详细信息。例如，将鼠标悬停在此示例中的关键 POD 上可显示有关该 POD 的详细信息。



您可以将鼠标悬停在 Node 元素上方来查看文件系统，内存和 CPU 信息。



有关仪表的说明

内存和 CPU 量表显示三种颜色，因为它们显示的 *used* 关系到 *_allocatable capacity* 和 *total capacity*。

Kubernetes网络性能监控和映射


Kubernetes网络性能监控和映射功能通过映射服务(也称为工作负载)之间的依赖关系来简化故障排除、并提供对网络性能等待时间和异常的实时可见性、以便在性能问题影响用户之前发现这些问题。

此功能可通过分析和审核Kubernetes流量来帮助企业降低整体成本。

主要功能：•工作负载图显示了Kubernetes工作负载的依赖关系和流、并重点显示了网络 and 性能问题。•监控Kubernetes Pod、工作负载和节点之间的网络流量；确定流量来源和延迟问题。•通过分析传入、传出、跨区域和跨区域网络流量来降低整体成本。

前提条件

在使用Kubbbernetes网络性能监控和映射之前、必须先配置 "[NetApp Kubernetes监控操作员](#)" 以启用此选项。在操作员部署期间、选中"Network Performance and Map"(网络性能和映射)复选框以启用。您也可以通过导航到Kubbbernetes登录页面并选择"修改部署"来启用此选项。

 **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

Need Help?

☒ Network Performance and Map

☒ Events Log

Complete Setup

监控器

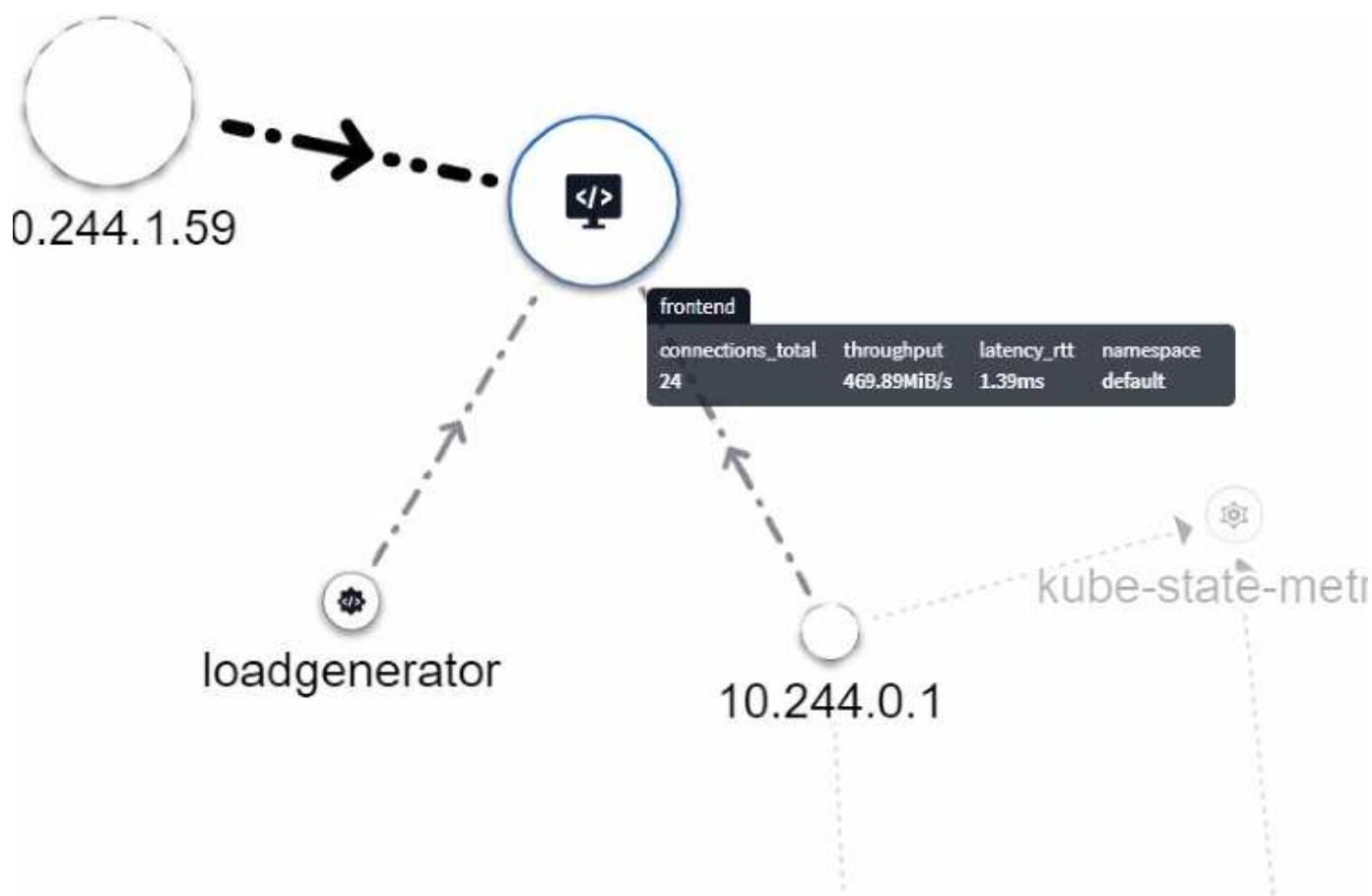
Workload Map可使用"监控器"来派生信息。Data Infrastructure Insight提供了许多默认的Kubernetes监控器(请注意、默认情况下、这些监控器可能为_Paused_。您可以_Resume_(即启用)所需的监控器)、也可以为Kubernetes对象创建自定义监控器、工作负载映射也将使用这些监控器。

您可以为以下任何对象类型创建Data Infrastructure Insight指标警报。确保数据按默认对象类型分组。

- Kubernetes.Workload
- Kubernetes.daemonset
- kubernetes.deployment
- Kubernetes.cronJob
- Kubernetes.job
- Kubernetes.replicaset
- Kubernetes.statefulset
- Kubernetes.Pod
- kubernetes.network_traffic_l4

地图

该映射显示了服务/工作负载及其相互关系。箭头显示交通方向。将鼠标悬停在某个工作负载上可显示该工作负载的摘要信息、如以下示例所示：

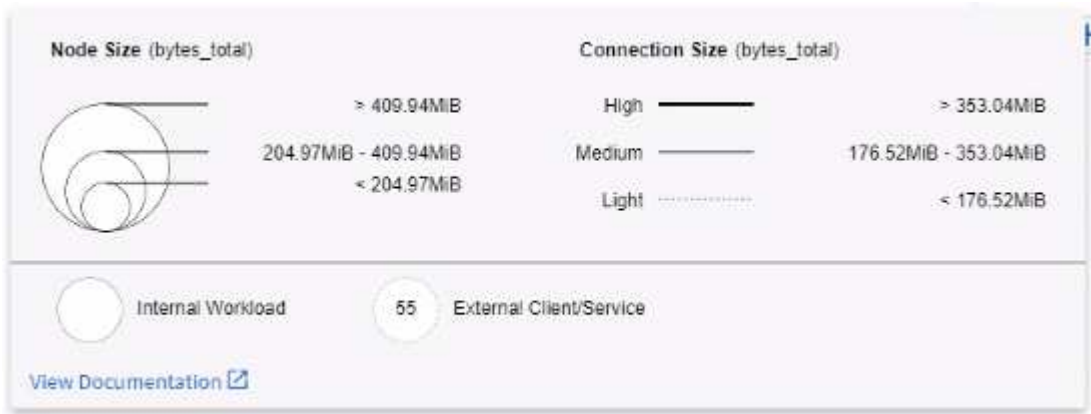


圆圈内的图标表示不同的服务类型。请注意、只有在底层对象具有时、图标才可见 [labels](#)。



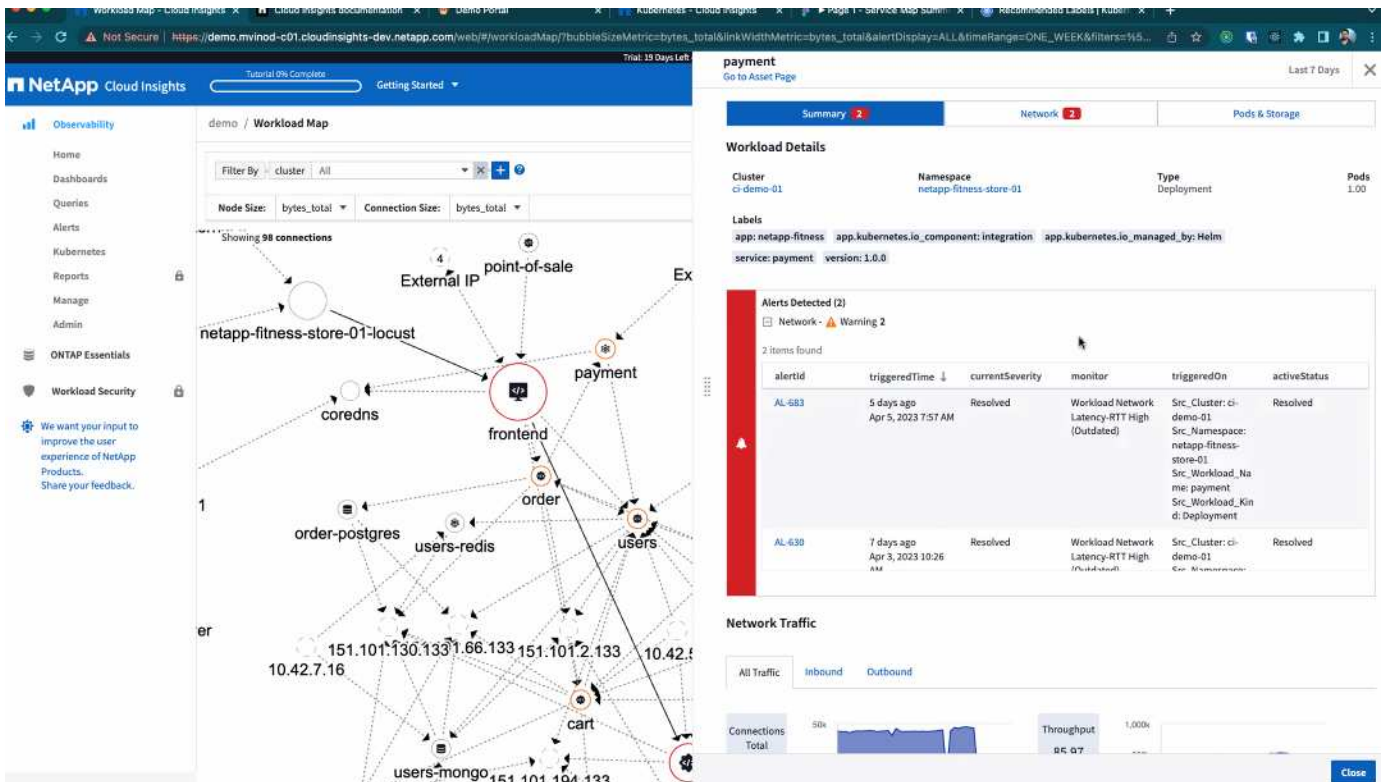
每个圆圈的大小表示节点大小。请注意、这些大小是相对的、您的浏览器缩放级别或屏幕大小可能会影响实际的圆圈大小。同样、交通线路样式也可让您一目了然地查看连接大小；粗实线表示高流量、浅虚线表示低流量。

圆圈内的数字表示服务当前正在处理的外部连接数。



工作负载详细信息和警报

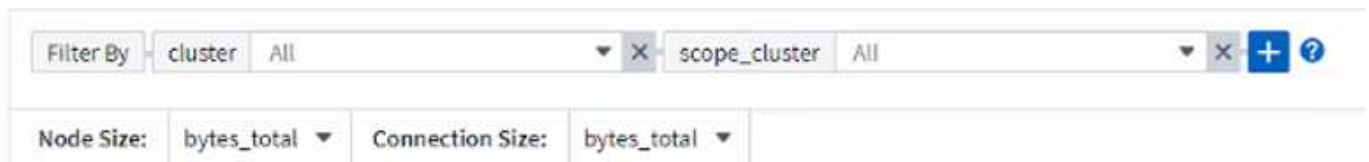
以颜色显示的圆圈表示工作负载的警告或严重级别警报。将鼠标悬停在圆圈上可查看问题描述 摘要、或者单击圆圈可打开包含更多详细信息的滑出面板。



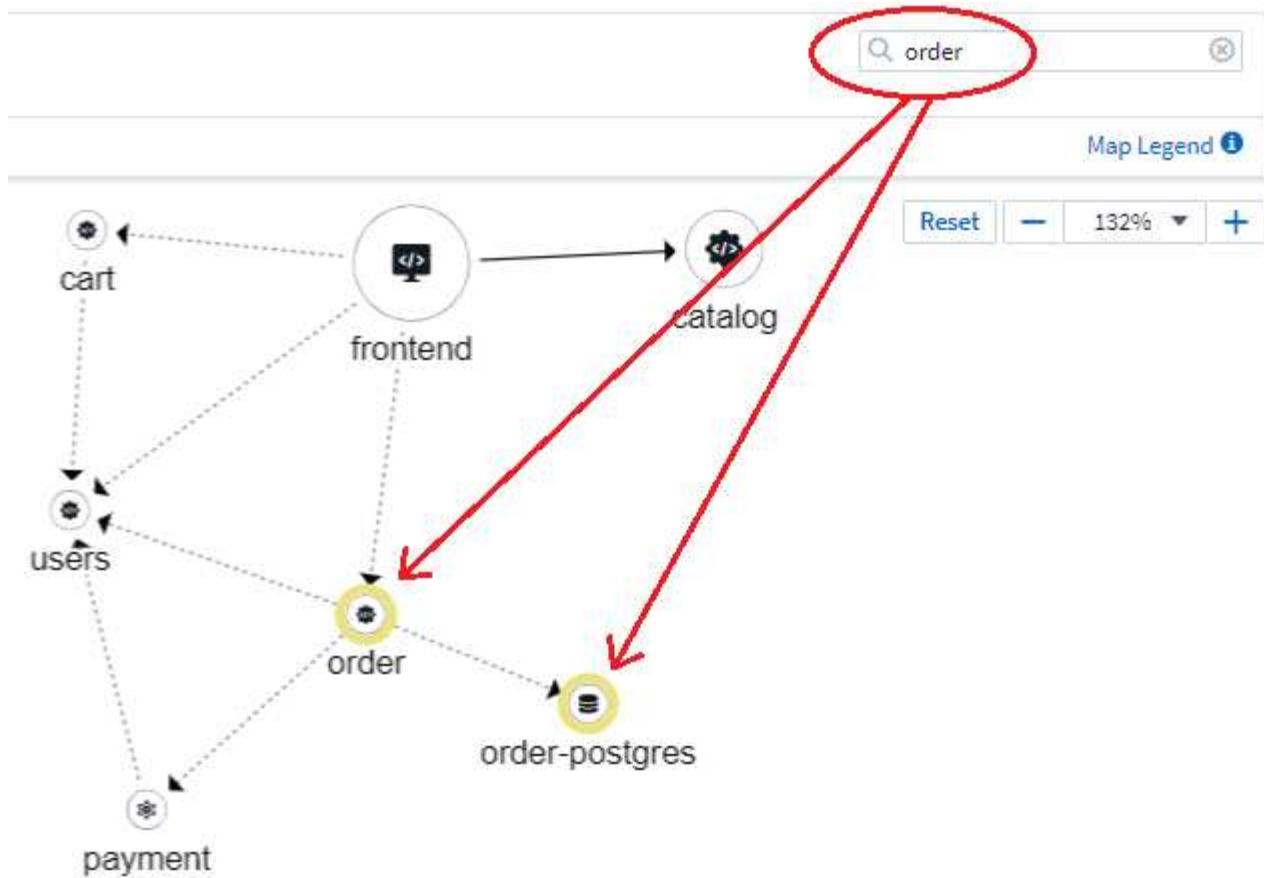
查找和筛选

与其他Data Infrastructure Insight功能一样、您可以轻松设置筛选器、以关注所需的特定对象或工作负载属性。

AQA / Workload Map



同样、在_find_字段中键入字符串将突出显示匹配的工作负载。



工作负载标签

如果希望映射标识显示的工作负载类型(即圆圈图标)、则需要使用工作负载标签。标签派生如下:

- 以通用术语运行的服务/应用程序的名称
- 如果源为POD:
 - 标签源自POD的工作负载标签
 - 工作负载上的预期标签: `app.Kubernetes.io/component`
 - 标签名称引用: <https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels/>
 - 建议标签:
 - 前端
 - 后端
 - 数据库
 - 缓存
 - 队列
 - Kafka
- 如果源位于Kubnetes集群外部:
 - Data Infrastructure Insight将尝试解析DNS解析名称以提取服务类型。

例如、如果DNS解析名称为_s3.eu-north-1.amazonaws.com、则解析后的名称将获取_S3_作为服务类型。

深入剖析

右键单击工作负载可提供更多选项供您进一步了解。例如、您可以从此处放大查看该工作负载的连接。



或者、您也可以打开详细信息分出面板、直接查看_Summary_、_Network_或_Pod & Storage_选项卡。

frontend
[Go to Asset Page](#)

Last 3 Hours

✕

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

最后、选择_Go to Asset Page_将打开工作负载的详细资产登录页面。

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

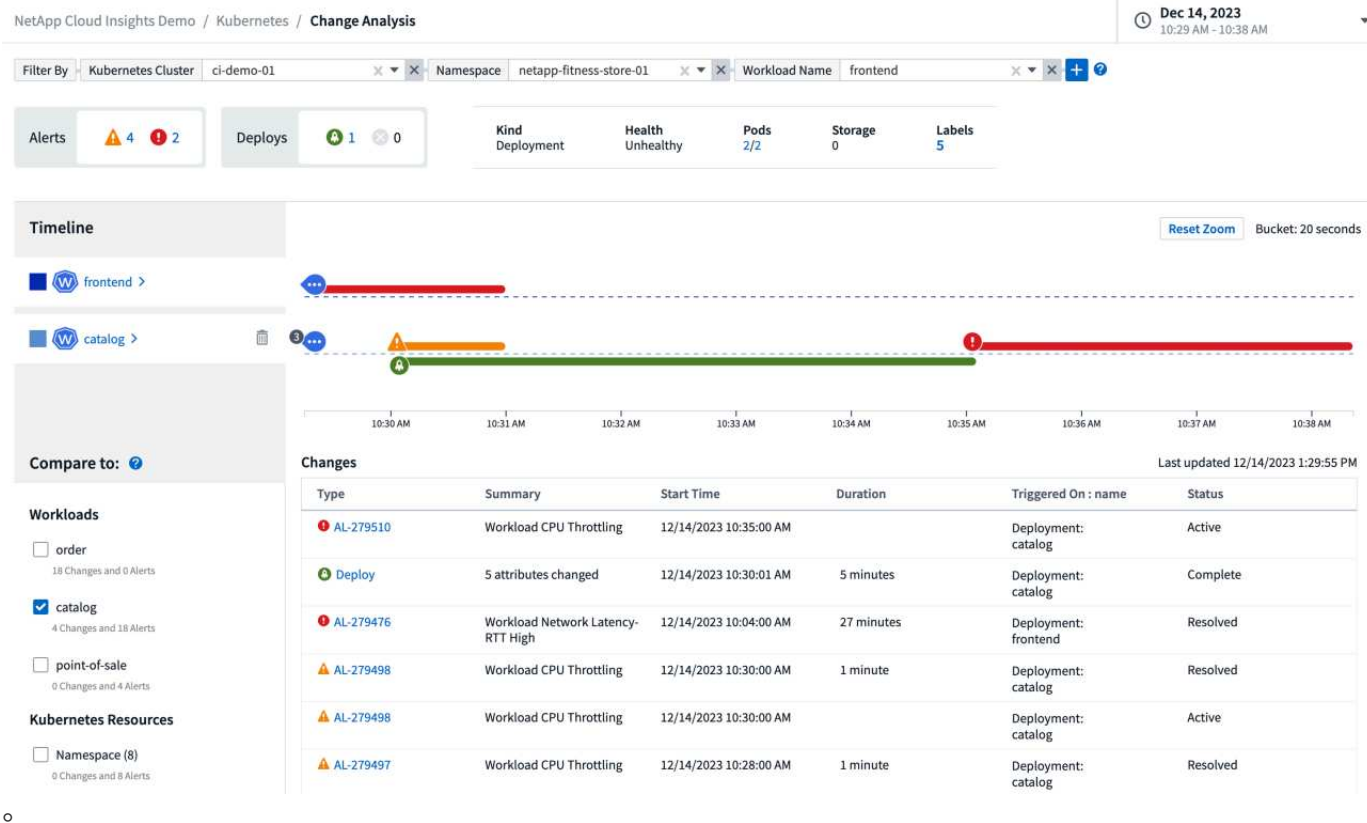
Kubernetes变更分析

Kubernetes变更分析为您提供了一个一体化视图、用于查看K8s环境的最新变更。警报和部署状态触手可及。借助变更分析、您可以跟踪每个部署和配置变更、并将其与K8s服务、基础架构和集群的运行状况和性能相关联。

变更分析有何帮助？

- 在多租户Kubernetes环境中、可能会由于配置不当的更改而发生中断。变更分析通过提供一个窗格来查看和关联工作负载的运行状况和配置更改、从而有助于实现这一点。这可能有助于对动态Kubernetes环境进行故障排除。

要查看Kubernetes变更分析、请导航到* Kubernetes >变更分析*。

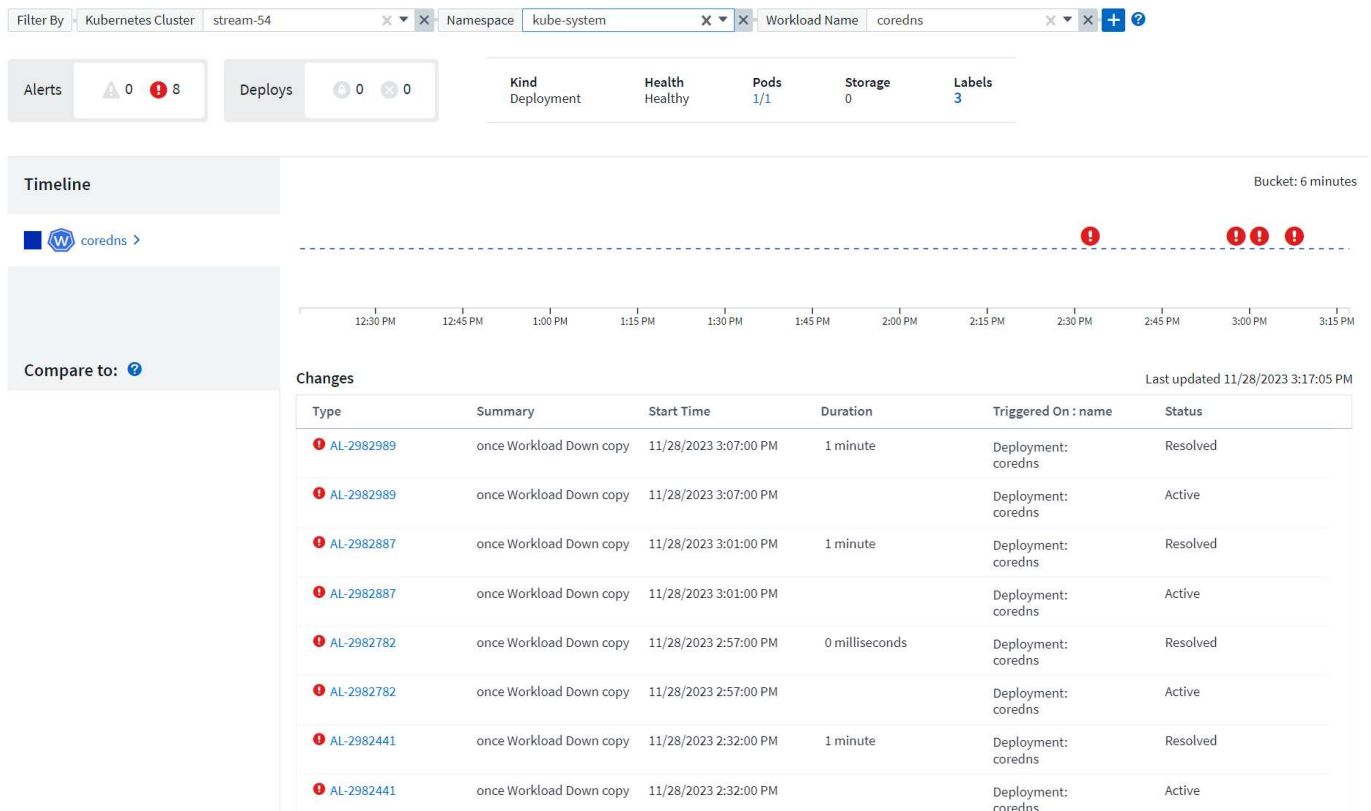


此页面会根据当前选定的Data Infrastructure Insight时间范围自动刷新。较小的时间范围意味着屏幕刷新频率更高。

筛选

与Data Infrastructure Insight的所有功能一样、筛选更改列表也非常直观：在页面顶部、输入或选择Kubernetes集群、命名空间或工作负载的值、或者通过选择 {+} 按钮添加您自己的筛选器。

筛选到特定集群、命名空间和工作负载(以及您设置的任何其他筛选器)时、系统将显示该集群上该命名空间中该工作负载的部署和警报时间表。通过单击并拖动图形来进一步放大、以关注更具体的时间范围。



快速状态

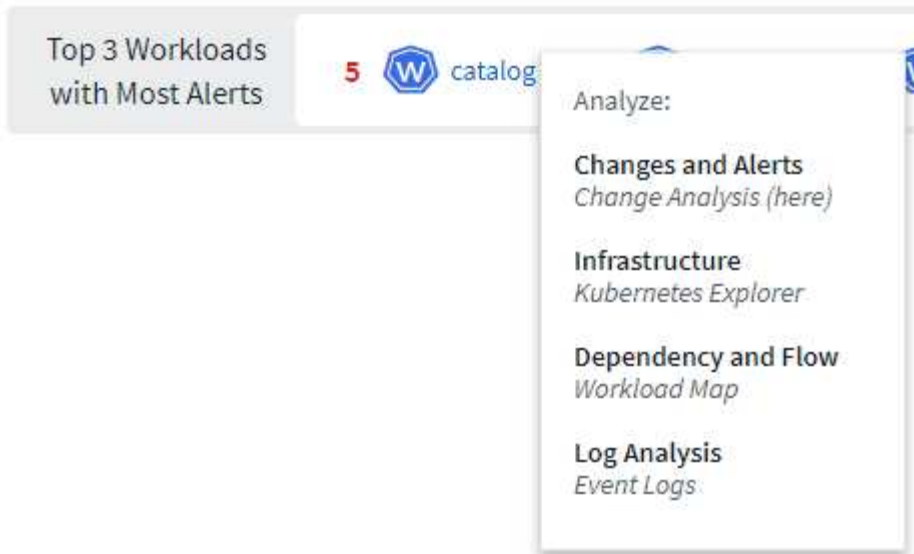
筛选区域下方有许多高级别的指示器。左侧是警报数量(警告和严重)。此数字包括_Active_警报和_已解决_警报。要仅查看_Active_alerts_、请为"Status"设置筛选器、然后选择"Active"。



此处还会显示部署状态。同样、默认值是显示_started_、_complete_和_failed部署的计数。要仅查看_failed部署、请为"Status"设置筛选器、然后选择"Filed"。



下一个是警报最多的前3个工作负载。每个工作负载旁边的红色数字表示与该工作负载相关的警报数量。单击工作负载链接可浏览基础架构(Kubernetes Explorer)、依赖关系(工作负载映射)或日志分析(事件日志)。



详细信息面板

在列表中选择一项更改将打开一个面板、以更详细地描述更改。例如、选择失败的部署将显示部署摘要、开始和结束时间、持续时间以及部署的触发位置、并提供用于浏览这些资源的链接。此外、它还会显示失败原因、任何相关更改以及任何关联事件。

✖ 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

同样、选择警报可提供有关警报的详细信息、包括触发警报的监控器以及显示警报可视时间线的图表。

版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。