



Kubernetes

Cloud Insights

NetApp
April 09, 2024

This PDF was generated from https://docs.netapp.com/zh-cn/cloudinsights/kubernetes_landing_page.html on April 09, 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	7
NetApp Kubnetes监控操作员配置选项	23
Kubernetes 集群详细信息页面	33
Kubnetes网络性能监控和映射	37
Kubnetes变更分析	44

Kubernetes

Kubernetes 集群概述

Cloud Insights Kubernetes 资源管理器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，并可用于轻松深入了解调查领域。

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。

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	

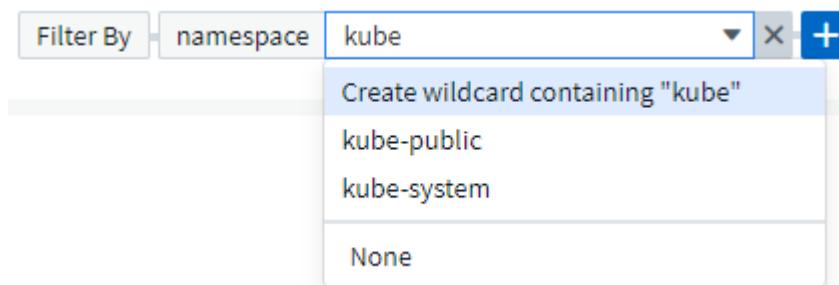
集群列表

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。



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项目将以淡蓝色显示。

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 Kubernetes监控操作员之前

在安装或升级NetApp Kubnetes监控操作员之前、请阅读此信息

前提条件：

- 如果您使用的是自定义或私有Docker存储库、请按照使用自定义或私有Docker存储库一节中的说明进行操作
- Kubernetes 1.20或更高版本支持安装NetApp Kubernetes监控操作员。
- 当Cloud Insights 监控后端存储且Kubernetes与Docker容器运行时结合使用时、Cloud Insights 可以显示NFS和iSCSI的POD到PV到存储映射和指标；其他运行时仅显示NFS。
- 从2022年8月开始、NetApp Kubernetes监控操作员支持Pod安全策略(PSP)。如果您的环境使用PSP、则必须升级到最新的NetApp Kubennetes Monitoring Operator。
- 如果您运行的是OpenShift 4.6或更高版本、则除了确保满足这些前提条件之外、您还必须遵循下面的OpenShift说明。
- 仅在Linux节点上安装监控Cloud Insights 支持监控运行Linux的KubeNet节点、方法是指定一个KubeNet节点选择器、用于在这些平台上查找以下KubeNet标签：

平台	标签
Kubernetes v1.20及更高版本	Kubernetes 。 io/OS = Linux
Rancher + catt.io 作为流程编排 /Kubernetes 平台	catt.io/OS = Linux

- 运行ARM64架构的节点不支持NetApp Kubernetes监控操作员及其依赖项(电报、Kube-state-metrics 、fluentbit等)。
- 必须提供以下命令：Curl、kubectl。可选安装步骤需要使用Docker命令。为获得最佳结果，请将这些命令添加到路径中。请注意、必须至少为kubect配置对以下Kubernetes对象的访问权限：代理、群集角色、群集角色、群集配置、自定义资源定义、部署、名区、角色、rolebinbeds、密钥、服务帐户、和服务。请参见此处、了解具有这些最小的"给予"角色权限的示例.yaml文件。
- 要用于安装NetApp Kubennetes监控操作员的主机必须已配置kubect特 尔、以便与目标K8s集群进行通信、并可通过Internet连接到您的Cloud Insights 环境。
- 如果您在安装期间或操作要监控的K8s集群时使用了代理、请按照配置代理支持一节中的说明进行操作。
- 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。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。
- 如果要重新部署Operator (即更新或替换Operator)、则无需创建_new_API令牌；您可以重新使用先前的令牌。

- 另请注意、如果您最近安装了NetApp Kubernetes Monitoring Operator、并且正在使用可续订的API访问令牌、则即将过期的令牌将自动替换为新的/刷新的API访问令牌。
- 网络监控：
 - 需要Linux内核版本4.18.0及更高版本
 - 不支持光子操作系统。

配置操作员

在较新版本的运算符中，可以在`_AgentConfiguration_`自定义资源中配置最常修改的设置。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。此文件包含一些已注释掉的设置示例。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`。

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](#)。

如果要从先前安装升级、请阅读[升级](#)信息。

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：

- 在执行安装代码片段(使用“curl”)期间需要使用代理将执行该片段的系统连接到Cloud Insights 环境
- 目标Kubernetes集群与Cloud Insights 环境通信所需的代理

如果您对其中一个或两个使用代理、则要安装NetApp Kubornetes操作监控器、必须先确保将代理配置为可以与Cloud Insights环境进行良好的通信。例如、从要安装Operator的服务器/VM中、您需要能够访问Cloud Insights并能够从Cloud Insights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Kubernetes集群与Cloud Insights环境通信所使用的代理、请在阅读完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 no proxy section, enter a comma-separated list of
  # IP addresses and/or resolvable hostnames that should bypass
  # the proxy
  no proxy: <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 Kubrenetes监控操作员将从Cloud Insights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Cloud Insights 存储库、提取操作员的所有映像依赖关系、然后注销Cloud Insights 存储库。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
- Kubernetes-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Cloud Insights 存储库中的映像一致。

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 link here: https://docs.netapp.com/us-
  # en/cloudinsights/task_config_telegraf_agent_k8s.html#using-a-custom-or-
  # private-docker-repository
  dockerRepo: your.docker.repo/long/path/to/test
  # Optional: A docker image pull secret that maybe needed for your
  # private docker registry
  dockerImagePullSecret: docker-secret-name

```

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

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"

Kubernetes Monitoring Operator安装和配置

Cloud Insights 为Kubernetes收集提供了* NetApp Kubernetes监控操作员*(NKMO) 。添加数据收集器时、只需选择"Kubernetes "图块即可。



如果您使用的是Cloud Insights联邦版、则安装和配置说明可能与此页面上的说明不同。按照Cloud Insights中的说明安装NetApp Kubernetes监控操作员。

Choose a Data Collector to Monitor

可从Cloud Insights Docker注册表下载Kubnetes Operator和数据收集器。安装后、操作员将管理部署在Kubernetes集群节点中的任何与操作员兼容的收集器以获取数据、包括管理这些收集器的生命周期。在此链之后、将从收集器中获取数据并将其发送到Cloud Insights。

安装NetApp Kubernetes监控操作员之前



阅读 "[安装或升级前](#)" 安装或升级NetApp Kubornetes监控操作员之前的必备文档。

安装NetApp Kubernetes监控操作员

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 

Need Help?

Installation Instructions

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
clustername	netapp-monitoring

2 Download the operator YAML files

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

[Copy Download Command Snippet](#)

[!\[\]\(eb1074bfd91059c9cff57cf6b5c22a5b_img.jpg\) 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 **Kubernetes** 上安装 **NetApp Kubernetes** 监控操作员代理的步骤：

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。

了解更多信息 [正在配置代理](#)。

了解更多信息 [使用自定义/私有Docker存储库](#)。

在安装NetApp Kubnetes Monitoring Operator时、默认情况下会启用Kubnetes EMS日志收集。要在安装后禁用

此收集、请单击Kubernetes集群详细信息页面顶部的“修改部署”按钮、然后取消选择“日志收集”。

Cluster Information

Kubernetes Cluster: k3s-2nodes Log Collection: Enabled - Online

Deployment Options

Log Collection

Need Help?

Cancel **Complete Modification**

此屏幕还会显示当前日志收集状态。以下是可能的状态：

- 已禁用
- enabled
- Enabled (已启用)—正在进行安装
- Enabled (已启用)—脱机
- Enabled (已启用)-联机
- 错误- API密钥权限不足

升级

升级到最新的NetApp Kubernetes监控操作员

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Cloud Insights 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-netapp-kubernetes-monitoring-operator, 卸载>>  
现有操作员。  
* <<installing-the-netapp-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**NetApp Kubernetes**监控操作员

要停止NetApp Kubernetes监控操作员、请执行以下操作：

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

要启动NetApp Kubernetes监控操作员、请执行以下操作：

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

正在卸载

删除**NetApp Kubernetes**监控操作员

请注意、NetApp Kubernetes监控操作员的默认命名空间为"netapp-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监控操作员会自动安装Kube-state-metrics；无需用户交互。

Kube-state-metrics 计数器

使用以下链接访问这些Kube状态指标计数器的信息：

1. "[ConfigMap 指标](#)"
2. "[DaemonSet 指标](#)"
3. "[部署指标](#)"
4. "[传入指标](#)"
5. "[命名空间指标](#)"
6. "[节点指标](#)"
7. "[持久性卷指标](#)"
8. "[持久性卷声明指标](#)"
9. "[POD 指标](#)"

10. "ReplicaSet 指标"

11. "机密指标"

12. "服务指标"

13. "StatusSet 指标"

== Configuring the Operator

在较新版本的运算符中，可以在`_AgentConfiguration_`自定义资源中配置最常修改的设置。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。此文件包含一些已注释掉的设置示例。请参见列表 [xref:{relative_path}telegraf_agent_k8s_config_options.html](#) ["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

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：

- 在执行安装代码片段(使用"curl")期间需要使用代理将执行该片段的系统连接到Cloud Insights 环境
- 目标Kubernetes集群与Cloud Insights 环境通信所需的代理

如果您对其中一项或两项操作使用代理、则要安装NetApp Kubernetes操作监控器、必须先确保您的代理已配置为可以与Cloud Insights 环境进行良好的通信。如果您有一个代理、并且可以从要安装此操作员的服务器/VM访问Cloud Insights 、则您的代理可能已正确配置。

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Kubernetes集群与Cloud Insights环境通信所使用的代理、请在阅读完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 no proxy section, enter a comma-separated list of
    # IP addresses and/or resolvable hostnames that should bypass
    # the proxy
    no proxy: <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 Kubrenetes监控操作员将从Cloud Insights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Cloud Insights存储库、提取操作员的所有映像依赖关系、然后注销Cloud Insights存储库。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Kubernetes监控

- NetApp监控
- CI-Kube-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 Cloud Insights 存储库中的映像一致。

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: https://docs.netapp.com/us-
  en/cloudinsights/task_config_telegraf_agent_k8s.html#using-a-custom-or-
  private-docker-repository
  dockerRepo: your.docker.repo/long/path/to/test
  # Optional: A docker image pull secret that maybe needed for your
  private docker registry
  dockerImagePullSecret: docker-secret-name
```

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NetApp Kubernetes监控操作员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_kindsToIgnoreFromWatch: "secrets"_{。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

Cloud Insights 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所需的项目从 Cloud Insights 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
```

故障排除

如果在设置NetApp Kubernetes监控操作员时遇到问题、请尝试执行以下操作：

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 Cloud Insights 必须主动监控您的 Kubernetes 集群存储。
我在日志中看到如下消息： 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	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。 要获取 Kubnetes 版本、请执行以下操作： <i>kubectl</i> 版本 要获取 Kube-state-metrics 版本、请执行以下操作： <i>kubectl get Deploy /kube-state-metrics -o jsonpath='{..image}'</i> 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： <i>mutatingwebbankconfigurations</i> <i>validatingwebbankconfiguration</i> <i>volumeAttachments</i> 资源 更具体地说、他们可以使用以下命令行界面参数： 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 默认资源列表为： "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 罢 疑、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：</p> <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> <pre>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。 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/snowlkp、错误：mkdir /etc/tedlaf/.ca CHE：权限被拒绝。已忽略\n" func="gosnowsclap. (*DEPLOADER).Errorf" file="log.go:120" 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/snowspache/ocsp_response_cache.json : no s此 选项 文件或目录\n" func="gosnowsclap. (*DEPORTLogger).Errorf" file="log.go:120" 10月11日14: 23: 41 IP-172-31-39-47、特拉夫[1827] : 2021-10-11T14: 23: 41Z I! 启动 Telegraf 1.19.3</pre>	<p>请尝试以下操作：</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：https://docs.netapp.com/us-en/cloudinsights/task_config_telegraf_agent_k8s.html#openshift-instructions。</p>
<p>在Kubelnetes上、我的Telegraf ReporticaSet Pod报告以下错误：</p> <pre>[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</pre>	<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> <pre>kubect-rs edit rs德拉夫-rs</pre> <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： 我使用的是PSP/PSA环境。这是否会影响我的监控操作员？	请尝试以下操作： 如果您的Kubernetes集群运行时已设置Pod安全策略(PSP)或Pod安全准入(PSA)、则必须升级到最新的NetApp Kubernetes监控操作员。按照以下步骤升级到支持PSP/PSA的最新的新一轮驱动程序： 1. 卸载 上一个监控操作员： kubect delete agent agent-monitor-NetApp -n NetApp-monitoring kubect delete ns ns-monitoring kubect delete crd agents.monitoring.netapp.com kubect delete 集群角色agent-manager-Role agent-proxy-Role agent-metrics-reader kubect delete cluster rolebinding agent-manager-rolebinding agent-proxy-rolebinding agent-cluster-admin-rolebinding 2. 安装 监控运算符的最新版本。
我在尝试部署NKMOO时遇到问题、并且正在使用PSP/PSA。	1. 使用以下命令编辑代理： kubect -n <name-space> 编辑代理 2. 将"security-policy-enabled"标记为"false"。此操作将禁用Pod安全策略和Pod安全准入、并允许NKMO。使用以下命令进行确认： kubecol get PSP (应显示Pod安全策略已删除) kubect get all -n <namespace>
grep -i psp (应显示未找到任何内容) 如果您拥有自定义或专用Docker存储库、但尚未将NetApp Kubernetes监控操作员配置为正确识别该存储库、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。	出现"ImagePullBackoff"错误 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。 <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>	在KMO命名空间中、Net-Observer (Workload Map) Pod位于CrashLoopBackOff中

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"、"员号": "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> <ul style="list-style-type: none"> • 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KMO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数据或Kubernetes指标数据
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。	在新工单命名空间中、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NetApp Kubernetes监控操作员后、我的日志中立即显示以下内容：</p> <pre>[inputs.prometheus]插件错误：向发出HTTP请求时出错 <a href="http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics">http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: 获取 <a href="http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics">http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: 拨号<namespace> : LOOKUP Kube状态指标.tcp.svc.cluster-local: 无此主机</pre>
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Kubernetes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NetApp Kubernetes监控操作员部署的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>问题：</p> <p>编辑_AgentConfiguration_中的"特拉夫"部分、并 将_dockerMetricCollectionEnabled"设置为false。有关 详细信息、请参见操作员的 "配置选项"。</p> <p>注意： 如果您使用的是Cloud Insights联邦版、则 对_su_的使用有限制的用户将无法收集Docker指标、 因为要访问Docker套接字、需要以root用户身份运行预 制数据容器或使用_su_将预制数据用户添加到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：</p> <p>...</p> <p>规格：</p> <p>...</p> <p>电报：</p> <p>...</p> <ul style="list-style-type: none"> -名称： Docker 运行模式： <ul style="list-style-type: none"> - DemonSet 可进行的其他操作： <ul style="list-style-type: none"> -关键字： Docker _UNIS_sdoc_PLATORY 值： UNIX： //run/Docker。 sk <p>...</p> <p>...</p>	<p>请尝试以下操作：</p> <p>我在Telegraf日志中看到重复出现以下错误消息：

 E! [agent]写入Outputs.http： POST时出错 "<a href="https://&lt;tenant_url&gt;/rest/v1/lake/ingest/influxdb":" class="bare">https://&lt;tenant_url&gt;/rest/v1/lake/ingest/influxdb": 超过上下文期限(等待标头时 超出客户端超时)</p>
<p>编辑_AgentConfiguration_中的"特拉夫"部分、并 将_dockerMetricCollectionEnabled"设置为false。有关 详细信息、请参见操作员的 "配置选项"。</p>	<p>我缺少一些事件日志的_volvedobject_数据。</p>
<p>确保已按照中的步骤进行操作 "权限" 第节。</p>	<p>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></p>
<p>自2023年10月12日起、Cloud Insights已对运营商进行 重构、以更好地为用户服务；要全面采用这些更改、您 必须执行 删除旧运算符 和 安装新的。</p>	<p>我的Kubernetes事件意外停止向Cloud Insights报告。</p>
<p>检索事件导出器Pod的名称：</p> <pre>`kubectl -n netapp-monitoring get pods`</pre>	<p>grep event-exporter</p>

问题:	请尝试以下操作:
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 <code>kubectl -n netapp-monitoring edit agent</code>，然后设置<code>log_file</code>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、<code>log_file</code>应设置为"/var/log/containers/NetApp-CI-event-exporter.log"或"/var/log/containers/event-exporter 。log"</p> <p>....</p> <p>fluent-bit:</p> <p>...</p> <pre>- name: event-exporter-ci substitutions: - key: LOG_FILE values: - /var/log/containers/netapp-ci-event-exporter.log ...</pre> <p>....</p> <p>或者、也可以这样做 卸载 和 重新安装 代理。</p>
我发现NetApp Kubenetes监控操作员部署的POD因资源不足而崩溃。	请参见NetApp Kubenetes监控操作员 " 配置选项 " 根据需要增加CPU和/或内存限制。

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](#)"。

NetApp Kubernetes监控操作员配置选项

- 。 "[NetApp Kubernetes监控操作员](#)" 可以自定义安装和配置。

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Cloud Insights Docker repo相比、使用dockerRepo 覆盖从客户的专用Docker reposo中提取映像。默认值为Cloud Insights 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Cloud Insights 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
	secionInterval	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
	dockerMountPoint	对dockerMountPoint路径的覆盖。这适用于在K8s平台(如云代工厂)上安装非标准Docker
	文档UnixSocket	对dockerUnixSocket路径的覆盖。这适用于在K8s平台(如云代工厂)上安装非标准Docker。
	配置路径	对克里oSockPath路径的覆盖。这适用于在K8s平台(如云代工厂)上安装非标准Docker。
	run特权	在特权模式下运行该特拉夫容器。如果在K8s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 "

组件	选项	说明
	输出超时	请参见 " Telegraf配置文档 "
	已启用dockerMetricCollection	收集Docker指标。默认情况下、此值设置为true、系统将为基于Docker的内部部署K8s收集Docker指标。要禁用Docker指标收集、请将此选项设置为false。
	ds.公差	Telegraf-DS的额外耐受性。
	Rs公差	Telegraf-RS额外的耐受性。
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KIBEstate 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。
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
	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

组件	选项	说明
	L4-公差	Net-obler-L4-DS附加容错。
	run特权	true或false—如果在Kubernetes节点上启用了SELinux，则将run特权设置为true。
变更管理		Kubernetes变更管理和分析的配置选项
	cpuLimit	change-obit 目标-Watch RS的CPU限制
	memLimit	change-obit 目标-Watch RS的MEM限制
	cpuQuest	对change-obit 目标-Watch -rs的CPU请求
	MemQuest	MEM请求change-obit 目标-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-obit 目标-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

下面是一个AgentConfiguration文件示例。

```

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

```

```

spec:
  # # You can modify the following fields to configure the operator.
  # # Optional settings are commented out and include default values for
reference
  # # To update them, uncomment the line, change the value, and apply
the updated AgentConfiguration.

  agent:
    # # [Required Field] A uniquely identifiable user-friendly
clustername.
    # # clusterName must be unique across all clusters in your Cloud
Insights environment.
    clusterName: "CLUSTERNAME_PLACEHOLDER"

    # # Proxy settings. The proxy that the operator should use to send
metrics to Cloud Insights.
    # # Please see documentation here: https://docs.netapp.com/us-
en/cloudinsights/task_config_telegraf_agent_k8s.html#configuring-proxy-
support

    # proxy:
    #   server:
    #     port:
    #     noProxy:
    #     username:
    #     password:
    #     isTelegrafProxyEnabled:
    #     isFluentbitProxyEnabled:
    #     isCollectorsProxyEnabled:

    # # [Required Field] By default, the operator uses the CI repository.
    # # To use a private repository, change this field to your repository
name.
    # # Please see documentation here: https://docs.netapp.com/us-
en/cloudinsights/task_config_telegraf_agent_k8s.html#using-a-custom-or-
private-docker-repository
    dockerRepo: 'DOCKER_REPO_PLACEHOLDER'
    # # [Required Field] The name of the imagePullSecret for dockerRepo.
    # # If you are using a private repository, change this field from
'docker' to the name of your secret.
{{ if not (contains .Values.config.cloudType "aws") }}# {{ end -}}
    dockerImagePullSecret: 'docker'

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

```

```

rotated. This must be less than the total ApiKey duration.

# tokenRotationThresholdDays: '{{  

.Values.telegraf_installer.kubernetes.rs.shim_token_rotation_threshold_days  

}}}'  

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_collec-
tion }'

    # # 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,persistent-
volumeclaims,persistentvolumes,pods,replicasets,resourcequotas,services,s-
tatefulsets'

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

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

    # labels:
'cronjobs=*,daemonsets=*,deployments=*,ingresses=*,jobs=*,namesp-
aces=*,nodes=*,persistentvolumeclaims=*,persistentvolumes=*,pods=*
*,replicasets=*,resourcequotas=*,services=*,statefulsets=*''

    # # kube-state-metrics additional tolerations. Use the following
abbreviated single line format only.

    # # No tolerations are applied by default
    # # Example: '{key: taint1, operator: Exists, effect:
NoSchedule},{key: taint2, operator: Exists, effect: NoExecute}'
```

```

# tolerations: ''

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

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

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


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

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

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

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

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

```

```

NoSchedule}, {key: taint2, operator: Exists, effect: NoExecute}'

# tolerations: ''

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

# # Note: In OpenShift environments, this is set to true
automatically.

# runPrivileged: 'false'

# change-management:
# # change-observer-watch-rs CPU/Mem limits and requests.
# # See https://kubernetes.io/docs/concepts/configuration/manage-
resources-containers/
# cpuLimit: '500m'
# memLimit: '500Mi'
# cpuRequest: '100m'
# memRequest: '500Mi'

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

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

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

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

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

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

# # Frequency with which log records are sent to CI from the collector

```

```

# logRecordAggrIntervalSeconds: '20'

# # change-observer-watch-ds additional tolerations. Use the following
# abbreviated single line format only.

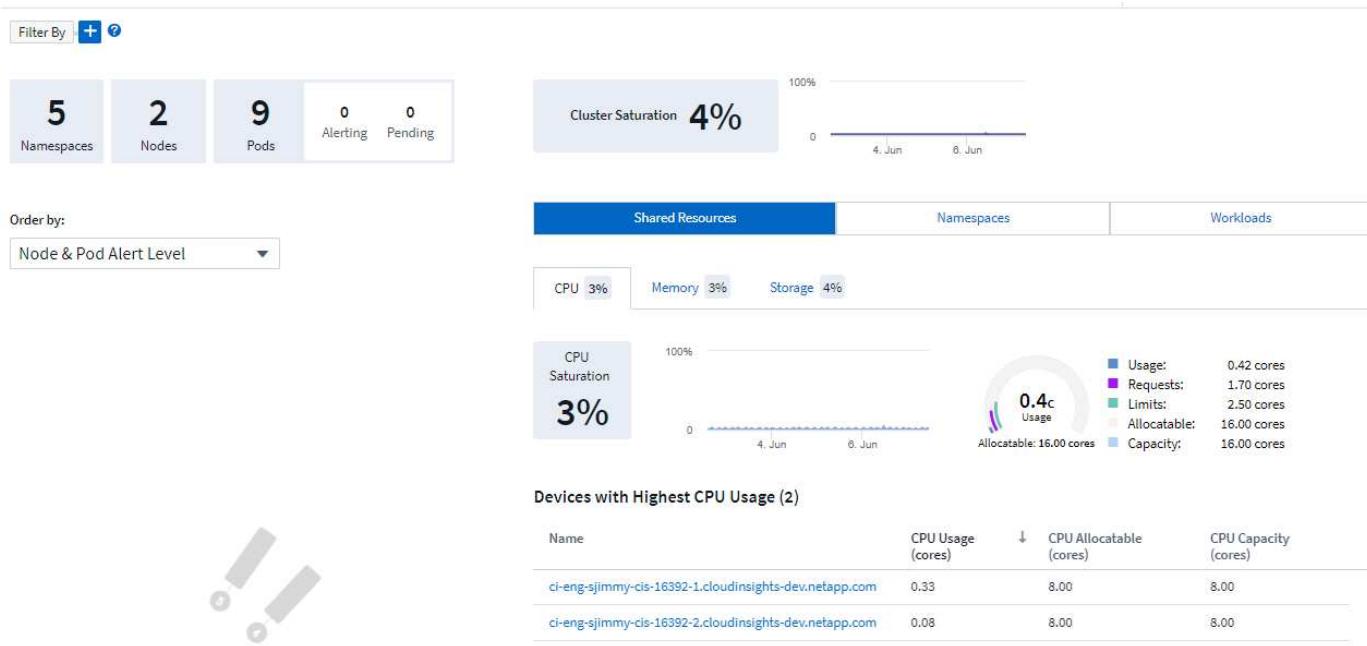
# # Inspect change-observer-watch-ds to view tolerations which are
# always present.

# # Example: '{key: taint1, operator: Exists, effect:
NoSchedule}, {key: taint2, operator: Exists, effect: NoExecute}'

# watch-tolerations: ''----
```

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 "



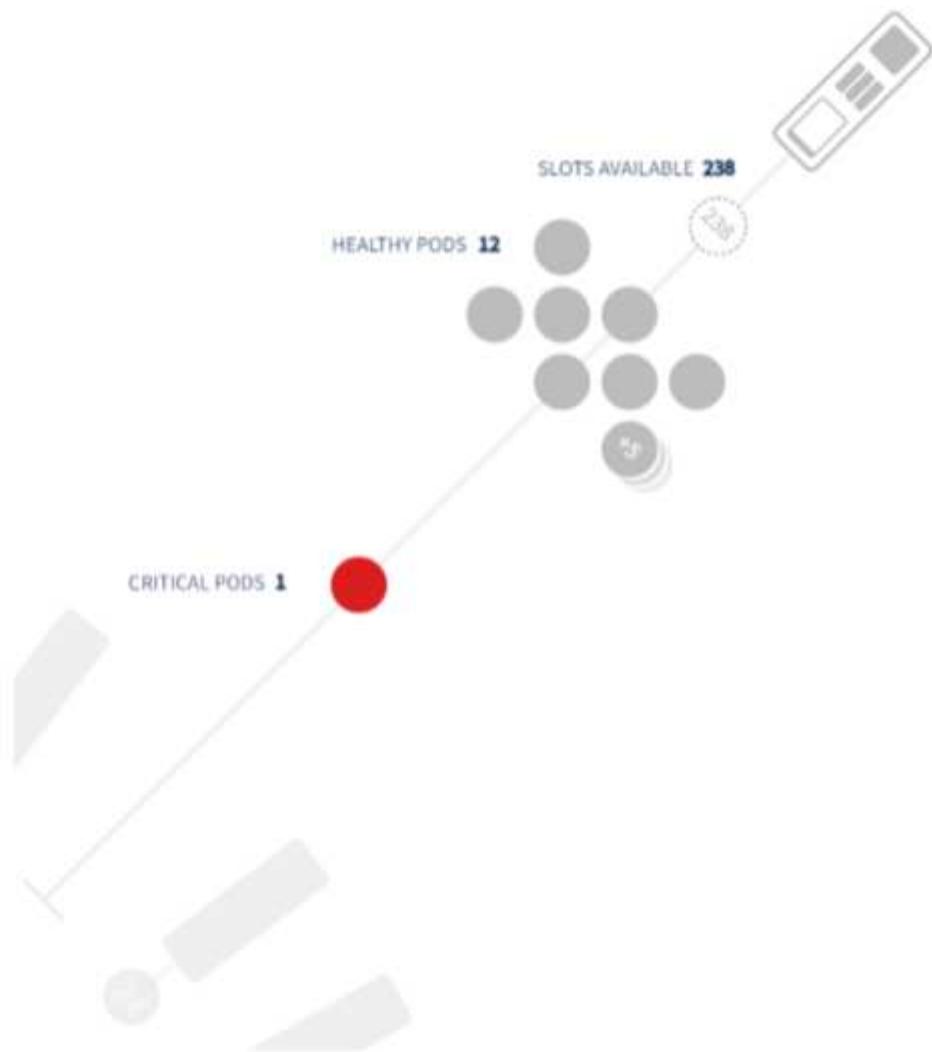
集群 "车轮" 部分简要介绍了节点和 POD 运行状况，您可以深入了解这些信息。如果集群包含的节点数超过页面此区域中显示的节点数，您可以使用可用按钮转动车轮。

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。

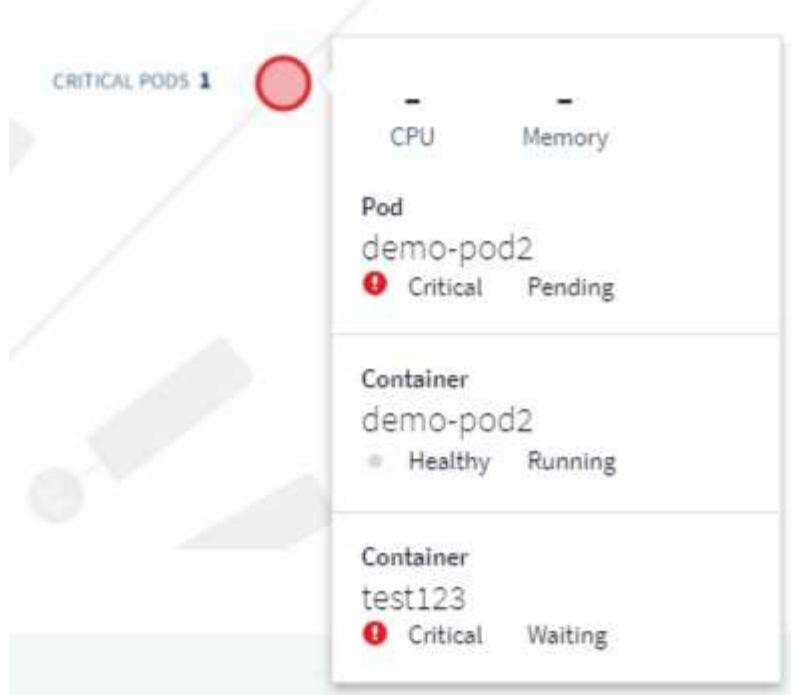
将鼠标悬停在 Pod（圆形）或 Node（条形）上可扩展节点视图。



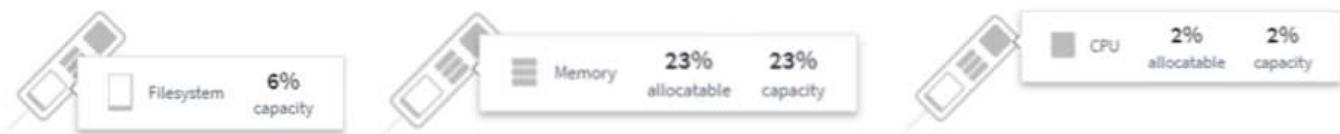
单击该视图中的 Pod 或节点将放大到展开的节点视图。



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、并重点显示了网络和性能问题。
- 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Kubbernetes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Kubbernetes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

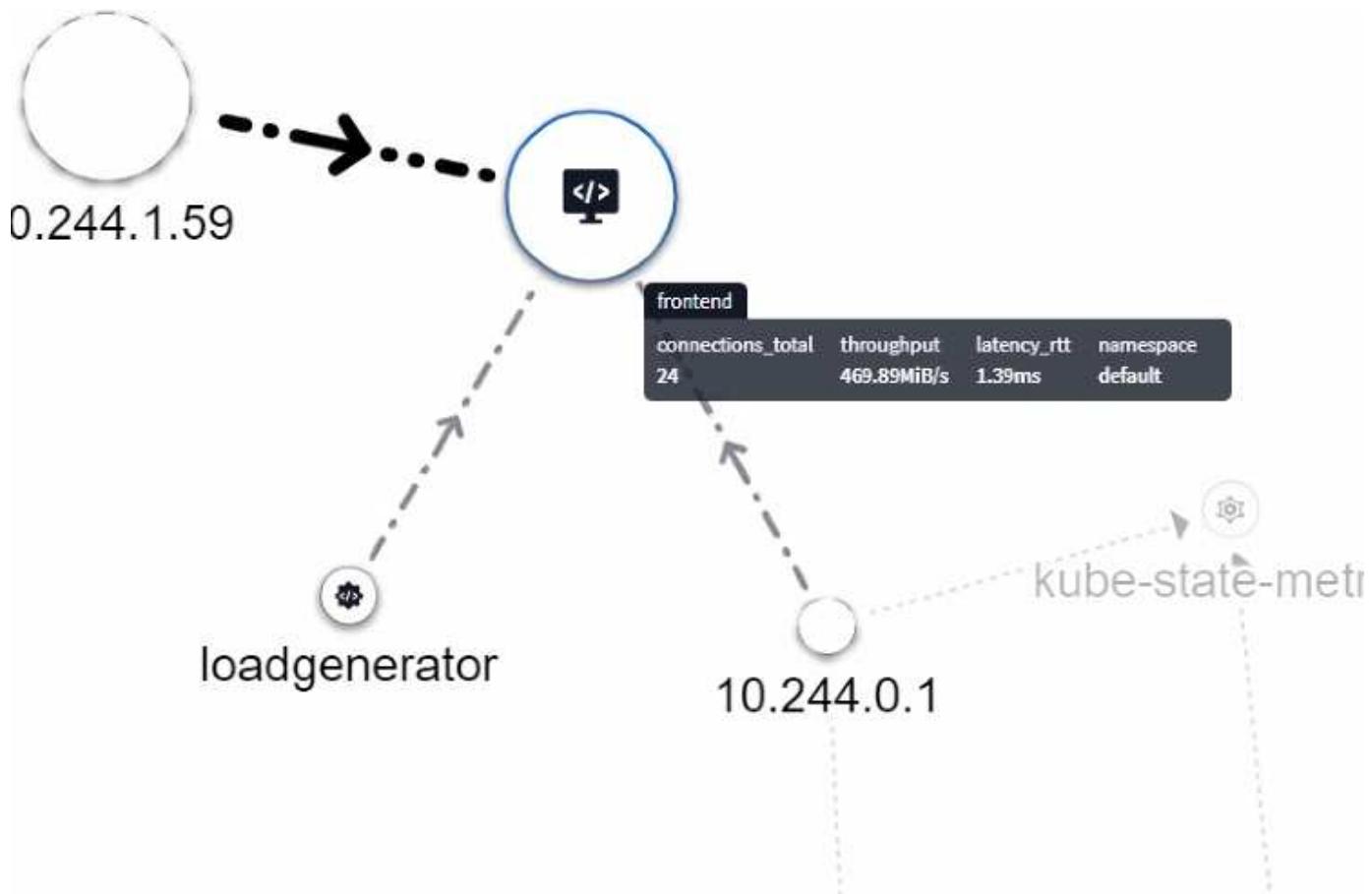
Workload Map使用“**监控器**”派生信息。Cloud Insights提供了许多默认的Kubernetes监控器(请注意、默认情况下、这些监控器可能为_Pauses_。您可以_Resume_(即启用)所需的监控器)、也可以为Kubornetes对象创建自定义监控器、工作负载映射也将使用这些监控器。

您可以在以下任意对象类型上创建Cloud Insights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：

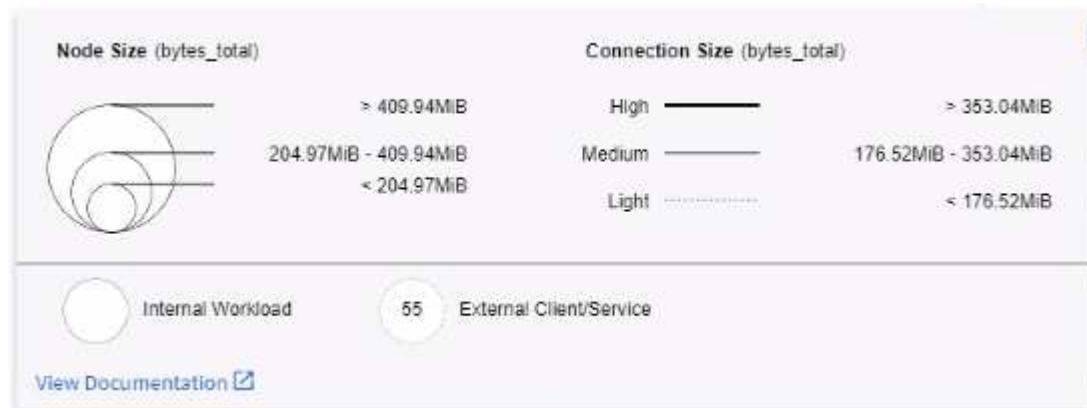


圆圈内的图标表示不同的服务类型。请注意、只有在底层对象具有时、图标才可见 [labels](#)。



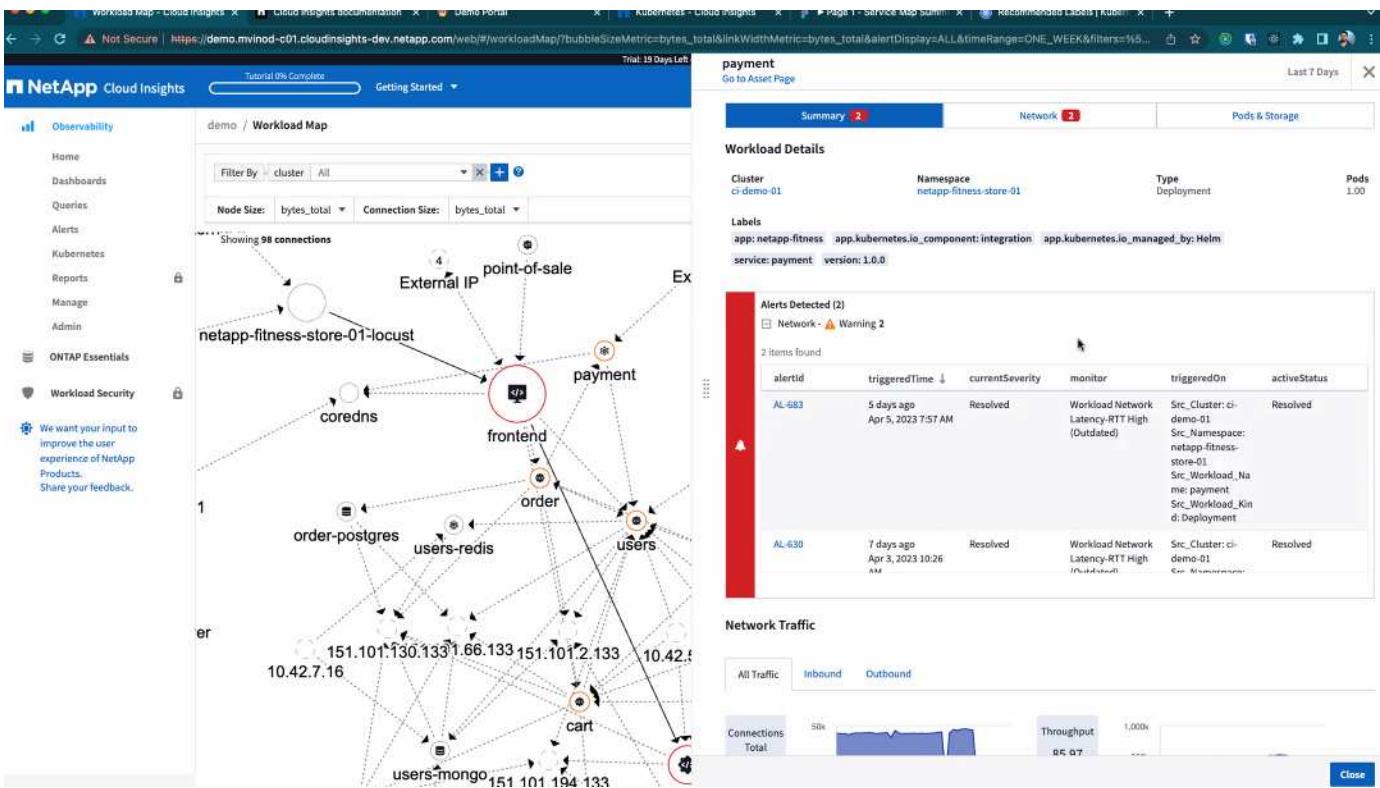
每个圆圈的大小表示节点大小。请注意、这些大小是相对的、您的浏览器缩放级别或屏幕大小可能会影响实际的圆圈大小。同样、交通线路样式也可让您一目了然地查看连接大小；粗实线表示高流量、浅虚线表示低流量。

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Cloud Insights 功能一样、您可以轻松地设置筛选器、以关注所需的特定对象或工作负载属性。

AQA / Workload Map

Filter By: cluster All scope_cluster All

Node Size: bytes_total Connection Size: bytes_total

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

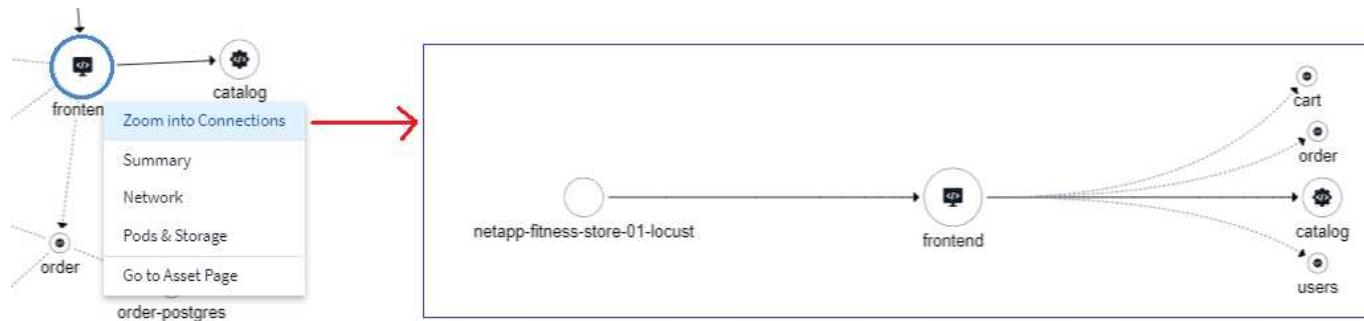
如果希望映射标识显示的工作负载类型(即圆圈图标)、则需要使用工作负载标签。标签派生如下：

- 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：
 - Cloud Insights 将尝试解析DNS解析名称以提取服务类型。

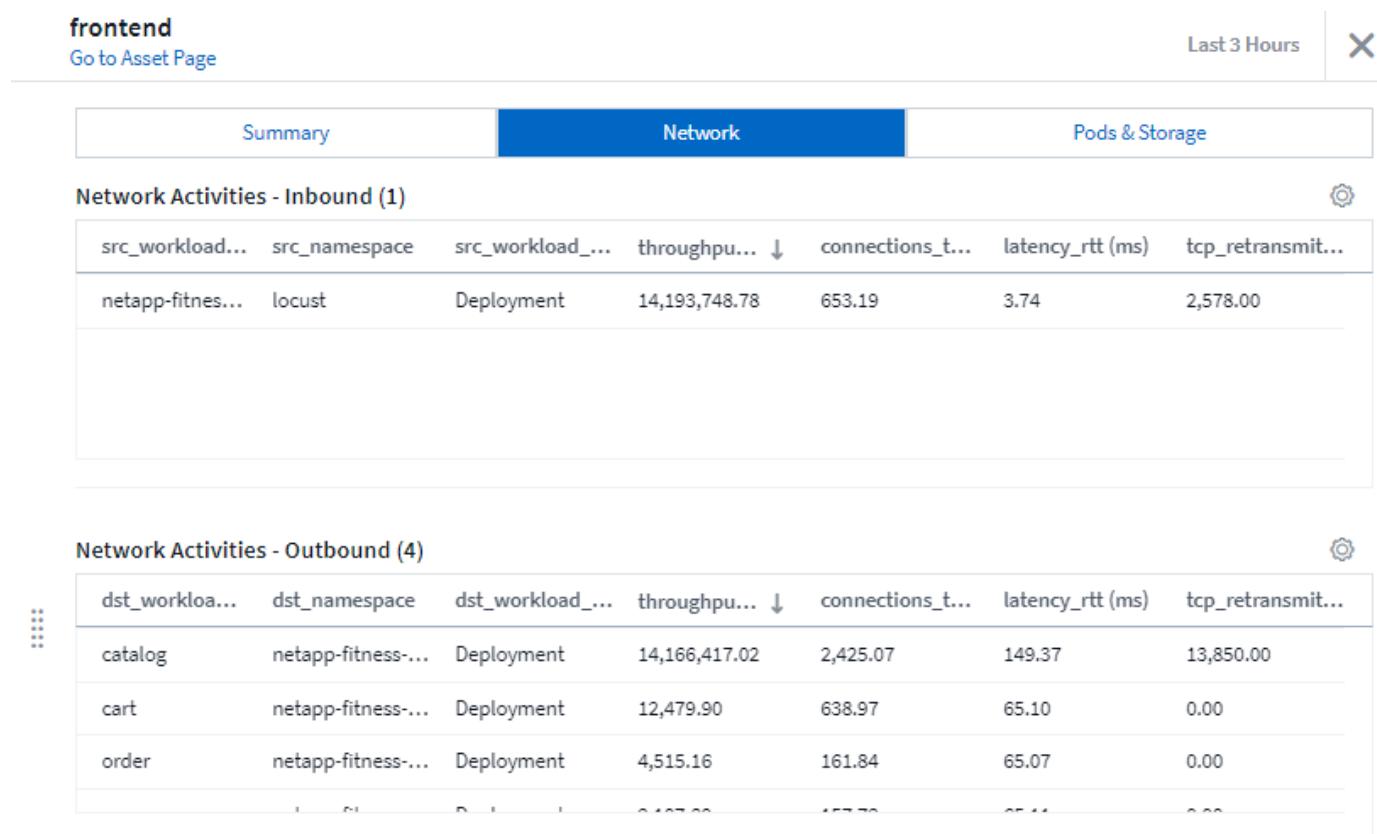
例如、如果DNS解析名称为_s3.eu-north-1.amazonaws.com、则解析后的名称将获取_S3_作为服务类型。

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。



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	2	0
Pods: Current / Desired	Up-to-date	Unavailable

Namespace netapp-fitness-store-01	Type Deployment	Date Created Apr 11, 2023 11:34 AM
Labels -		

**Highest CPU Demand by Pod**

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

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

Highest Memory Demand by Pod

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

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

Pods (2)

Pod Name ↑	Status	Healthy Containers	cpu_usage_nanocores (mc)	memory_rss_bytes (GiB)
frontend-7fcccd9f8f-284kb	Healthy Running	1 of 1	133	0.09
frontend-7fcccd9f8f-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。

请记住以下几点：

- 在多租户环境中、由于配置不当的更改、可能会发生中断。在动态环境中、Cloud Insights可能无法正确跟踪所有更改。
- 变更分析提供了一个窗格、用于查看和关联工作负载的运行状况和配置更改。这可能有助于对动态环境进行故障排除。

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

Filter By: Kubernetes Cluster: ci-demo-01 Namespace: netapp-fitness-store-01 Workload Name: frontend

Alerts: 4 (1) Deployments: 1 (0)

Kind: Deployment Health: Unhealthy Pods: 2/2 Storage: 0 Labels: 5

Timeline Reset Zoom Bucket: 20 seconds

Compare to: ?

Workloads

- order: 18 Changes and 0 Alerts
- catalog: 4 Changes and 18 Alerts
- point-of-sale: 0 Changes and 4 Alerts

Kubernetes Resources

- Namespace (8): 0 Changes and 8 Alerts

Changes Last updated 12/14/2023 1:29:55 PM

Type	Summary	Start Time	Duration	Triggered On : name	Status
AL-279510	Workload CPU Throttling	12/14/2023 10:35:00 AM		Deployment: catalog	Active
Deploy	5 attributes changed	12/14/2023 10:30:01 AM	5 minutes	Deployment: catalog	Complete
AL-279476	Workload Network Latency-RTT High	12/14/2023 10:04:00 AM	27 minutes	Deployment: frontend	Resolved
AL-279498	Workload CPU Throttling	12/14/2023 10:30:00 AM	1 minute	Deployment: catalog	Resolved
AL-279498	Workload CPU Throttling	12/14/2023 10:30:00 AM		Deployment: catalog	Active
AL-279497	Workload CPU Throttling	12/14/2023 10:28:00 AM	1 minute	Deployment: catalog	Resolved

页面将根据当前选定的Cloud Insights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Cloud Insights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。

Filter By: Kubernetes Cluster: stream-54 Namespace: kube-system Workload Name: coredns

Kind	Deployment	Health	Pods	Storage	Labels
	Deployment	Healthy	1/1	0	3

Timeline Bucket: 6 minutes

Compare to: ?

Changes Last updated 11/28/2023 3:17:05 PM

Type	Summary	Start Time	Duration	Triggered On : name	Status
AL-2982989	once Workload Down copy	11/28/2023 3:07:00 PM	1 minute	Deployment: coredns	Resolved
AL-2982989	once Workload Down copy	11/28/2023 3:07:00 PM		Deployment: coredns	Active
AL-2982887	once Workload Down copy	11/28/2023 3:01:00 PM	1 minute	Deployment: coredns	Resolved
AL-2982887	once Workload Down copy	11/28/2023 3:01:00 PM		Deployment: coredns	Active
AL-2982782	once Workload Down copy	11/28/2023 2:57:00 PM	0 milliseconds	Deployment: coredns	Resolved
AL-2982782	once Workload Down copy	11/28/2023 2:57:00 PM		Deployment: coredns	Active
AL-2982441	once Workload Down copy	11/28/2023 2:32:00 PM	1 minute	Deployment: coredns	Resolved
AL-2982441	once Workload Down copy	11/28/2023 2:32:00 PM		Deployment: coredns	Active

快速状态

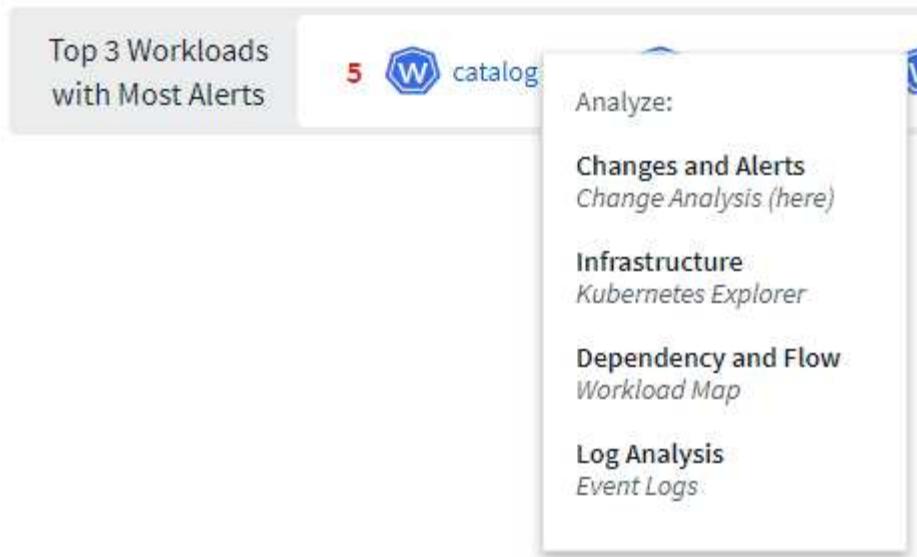
筛选区域下方有许多高级别的指示器。左侧是警报数量(警告和严重)。此数字包括_Active_警报 和 _已解决_警报。要仅查看_Active_alerts、请为"Status"设置筛选器、然后选择"Active"。

Alerts ⚠ 6 ! 17

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"Failed"。

Deploys 💡 36 ✖ 4

下一个警报最多的前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 >

Triggered On : kind

Deployment



netapp-fitness-store-01 >



billing-accounts >

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.deploy.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。