Kubernetes

Data Infrastructure Insights

NetApp
January 10, 2025

Índice

Kubernetes

Visão geral do cluster do Kubernetes

O Data Infrastructure Insights Kubernetes Explorer é uma ferramenta poderosa para exibir a integridade geral e o uso dos clusters do Kubernetes, além de permitir que você analise facilmente as áreas de investigação.

Clicar em **painéis > Kubernetes Explorer** abre a página de lista de clusters do Kubernetes. Esta página de visão geral contém uma tabela dos clusters do Kubernetes no seu locatário.

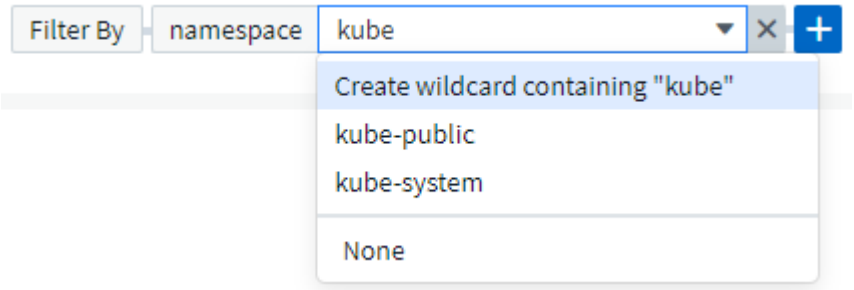
Lista de cluster

A lista de cluster exibe as seguintes informações para cada cluster no locatário:

- Cluster **Nome**. Clicar num nome de cluster abrirá o "**página de detalhes**" para esse cluster.
- **Percentagens de saturação**. A saturação geral é a mais alta da CPU, memória ou saturação de armazenamento.
- Número de * nós* no cluster. Clicar neste número abrirá a página da lista nó.
- Número de **pods** no cluster. Clicar neste número abrirá a página da lista Pod.
- Número de * namespaces* no cluster. Clicar nesse número abrirá a página de lista de namespace.
- Número de **cargas de trabalho** no cluster. Clicar neste número abrirá a página da lista carga de trabalho.

Refinando o filtro

Quando você está filtrando, à medida que você começa a digitar, você recebe a opção de criar um filtro * curinga* com base no texto atual. Selecionar esta opção irá retornar todos os resultados que correspondem à expressão curinga. Você também pode criar **expressões** usando NOT ou E, ou você pode selecionar a opção "nenhum" para filtrar valores nulos no campo.



Os filtros baseados em carateres universais ou expressões (por exemplo, NÃO, E, "nenhum", etc.) são exibidos em azul escuro no campo de filtro. Os itens que você selecionar diretamente da lista são exibidos em

azul claro.

Os filtros do Kubernetes são contextuais, o que significa, por exemplo, que se você estiver em uma página de nó específica, o filtro pod_name listará apenas os pods relacionados a esse nó. Além disso, se você aplicar um filtro para um namespace específico, o filtro pod_name listará apenas pods nesse nó *e* nesse namespace.

Observe que a filtragem de carateres curinga e expressão funciona com texto ou listas, mas não com valores numéricos, datas ou booleanos.

Antes de instalar ou atualizar o Operador de Monitoramento do Kubernetes do NetApp

Leia estas informações antes de instalar ou atualizar o "Operador de monitoramento do Kubernetes".

Componente	Requisito
Versão do Kubernetes	Kubernetes v1,20 e posterior.
Distribuições do Kubernetes	O Google Kubernetes Engine (GKE) Red Hat OpenShift Rancher Kubernetes Engine (RKE) VMware Tanzu é um dos nossos selecionados Jogos de Kubernetes
OS do Linux	O Data Infrastructure Insights não oferece suporte para nós executados com a arquitetura Arm64. Monitoramento de rede: Deve estar executando o kernel Linux versão 4.18.0 ou superior. O sistema operacional de fotão não é suportado.
Etiquetas	O Data Infrastructure Insights dá suporte ao monitoramento de nós do Kubernetes que estão executando o Linux, especificando um seletor de nós do Kubernetes que procura as seguintes etiquetas do Kubernetes nessas plataformas: Kubernetes v1,20 e superior: Kubernetes.io/os como plataforma de orquestração/Kubernetes: Cattle.io/os: linux
Comandos	Os comandos curl e kubectl devem estar disponíveis.; para obter melhores resultados, adicione esses comandos ao CAMINHO.
Conetividade	a cli do kubectl está configurada para se comunicar com o cluster K8s de destino e tem conetividade com a Internet ao seu ambiente Data Infrastructure Insights. Se você estiver atrás de um proxy durante a instalação, siga as instruções "Configurando o suporte Proxy"na seção da instalação do Operador. Para obter relatórios precisos de auditoria e dados, sincronize a hora na máquina do agente usando o Network Time Protocol (NTP) ou o Simple Network Time Protocol (SNTP).

Componente	Requisito
Outros	Se você estiver executando no OpenShift 4,6 ou superior, você deve seguir o "Instruções do OpenShift"além de garantir que esses pré-requisitos sejam atendidos.
Token de API	Se você estiver reimplantando o Operador (ou seja, está atualizando ou substituindo-o), não há necessidade de criar um novo token de API; você pode reutilizar o token anterior.

Coisas importantes a observar antes de começar

Se você estiver executando com um proxy, tiver um repositório personalizado, ou estiver usando OpenShift, leia as seções a seguir cuidadosamente.

Leia também Permissõessobre .

Configurando o suporte Proxy

Há dois lugares onde você pode usar um proxy em seu locatário para instalar o Operador de Monitoramento do Kubernetes do NetApp. Estes podem ser os mesmos ou sistemas proxy separados:

- Proxy necessário durante a execução do snippet de código de instalação (usando "curl") para conetar o sistema onde o snippet é executado ao seu ambiente Data Infrastructure Insights
- Proxy necessário pelo cluster do Kubernetes de destino para se comunicar com seu ambiente Data Infrastructure Insights

Se você usar um proxy para um ou ambos, para instalar o Monitor operacional do Kubernetes do NetApp, primeiro você deve garantir que o proxy esteja configurado para permitir uma boa comunicação com o ambiente Insights da infraestrutura de dados. Por exemplo, a partir dos servidores/VMs a partir dos quais você deseja instalar o Operador, você precisa ter acesso ao Data Infrastructure Insights e poder baixar binários do Data Infrastructure Insights.

Para o proxy usado para instalar o Monitor operacional NetApp Kubernetes, antes de instalar o Operador, defina as variáveis de ambiente *http_proxy/https_proxy*. Para alguns ambientes proxy, você também pode precisar definir a variável *no_proxy environment*.

Para definir a(s) variável(s), execute as seguintes etapas em seu sistema **antes** de instalar o Operador de Monitoramento do Kubernetes do NetApp:

1. Defina a(s) variável(s) de ambiente *https_proxy* e/ou *http_proxy* para o usuário atual:
   a. Se o proxy que está sendo configurado não tiver Autenticação (nome de usuário/senha), execute o seguinte comando:

```
 export https_proxy=<proxy_server>:<proxy_port>
.. Se o proxy que está sendo configurado tiver Autenticação (nome de
usuário/senha), execute este comando:
```

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

Para que o proxy usado para que o cluster do Kubernetes se comunique com o ambiente Insights de infraestrutura de dados, instale o Operador de Monitoramento do Kubernetes do NetApp depois de ler todas essas instruções.

Configure a seção proxy do AgentConfiguration no operator-config.yaml antes de implantar o Operador de Monitoramento do Kubernetes do NetApp.

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

Usando um repositório docker personalizado ou privado

Por padrão, o Operador de Monitoramento do Kubernetes do NetApp coletará imagens de contêiner do repositório de informações de infraestrutura de dados. Se você tiver um cluster do Kubernetes usado como destino para monitoramento e esse cluster estiver configurado para apenas extrair imagens de contentor de um repositório ou Registro de contentor personalizado ou privado do Docker, configure o acesso aos contentores necessários pelo Operador de Monitoramento do Kubernetes do NetApp.

Execute o "trecho de recebimento de imagem" do bloco de instalação do Operador de Monitoramento do NetApp. Esse comando fará login no repositório Data Infrastructure Insights, extrairá todas as dependências de imagem do operador e fará logout do repositório Data Infrastructure Insights. Quando solicitado, insira a senha temporária do repositório fornecida. Este comando transfere todas as imagens utilizadas pelo operador, incluindo as funcionalidades opcionais. Veja abaixo quais recursos essas imagens são usadas.

Funcionalidade do operador principal e monitoramento do Kubernetes

- monitoramento de NetApp
- kube-rbac-proxy
- kube-state-metrics
- telegraf
- distroless-root-user

Registo de eventos

- bit fluente
- kuseurs-event-exporter

Desempenho de rede e mapa

- ci-net-observador

Envie a imagem do docker do operador para o seu repositório docker privado/local/empresarial de acordo com suas políticas corporativas. Certifique-se de que as tags de imagem e os caminhos de diretório para essas imagens em seu repositório sejam consistentes com os do repositório Data Infrastructure Insights.

Edite a implantação do operador de monitoramento no operator-deployment.yaml e modifique todas as referências de imagem para usar seu repositório Docker privado.

```
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>
```

Edite o AgentConfiguration no operator-config.yaml para refletir o novo local de repo do docker. Crie uma nova imagePullSecret para o seu repositório privado, para obter mais detalhes consulte *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
```

Instruções do OpenShift

Se você estiver executando no OpenShift 4,6 ou superior, você deve editar o AgentConfiguration em *operator-config.yaml* para ativar a configuração *runPrivileged*:

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

O OpenShift pode implementar um nível adicional de segurança que pode bloquear o acesso a alguns componentes do Kubernetes.

Permissões

Se o cluster que você está monitorando contiver recursos personalizados que não tenham um ClusterRole que "agregados para visualizar", você precisará conceder manualmente ao operador acesso a esses recursos para monitorá-los com Registros de eventos.

1. Edite *operator-additional-permissions.yaml* antes de instalar, ou depois de instalar edite o recurso *ClusterRole/<namespace>-additional-permissions*
2. Crie uma nova regra para os apiGroups e recursos desejados com os verbos ["Get", "Watch", "list"]. Veja https://kubernetes.io/docs/reference/access-authn-authz/rbac/
3. Aplique as alterações ao cluster

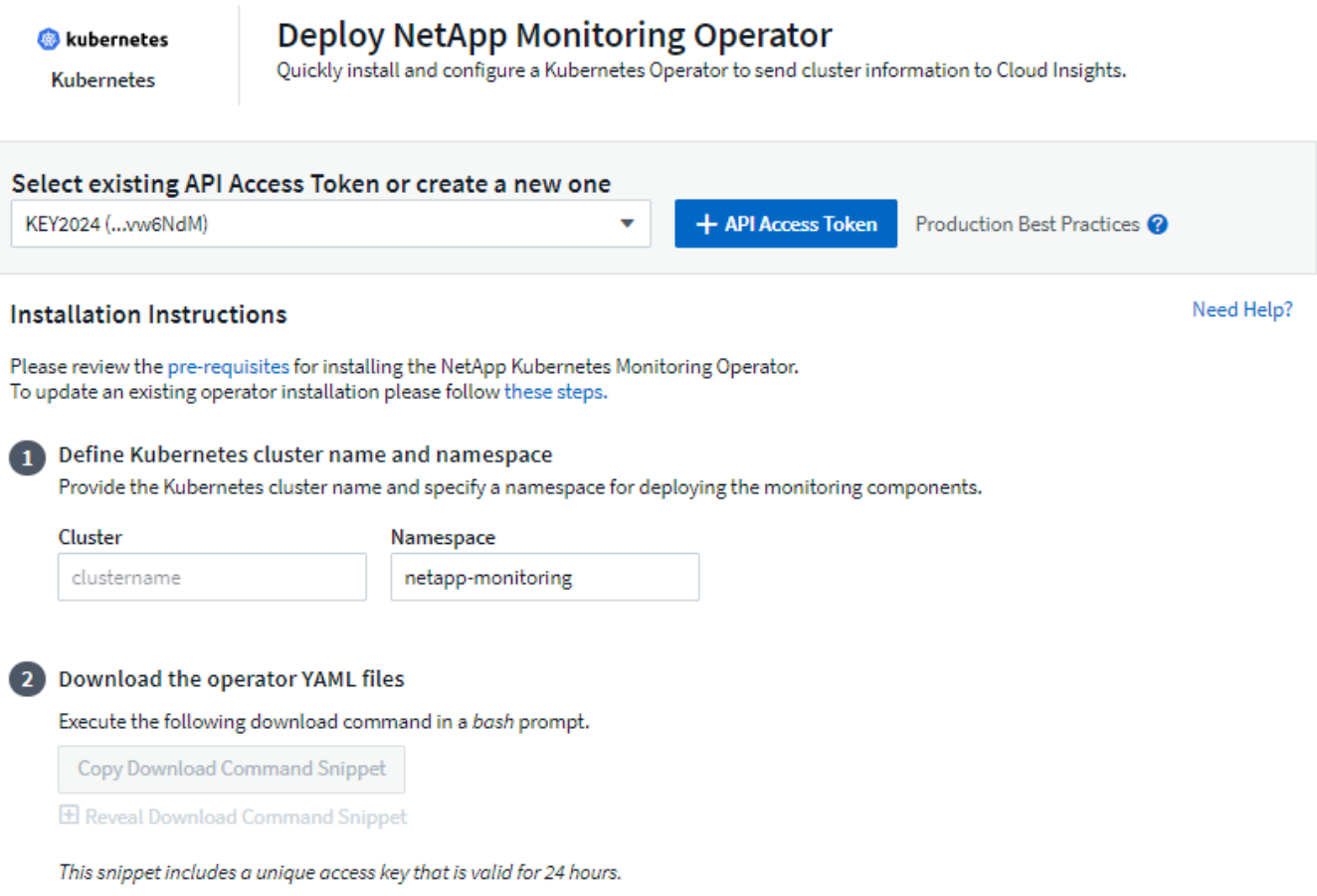
Instalação e configuração do operador de monitoramento Kubernetes

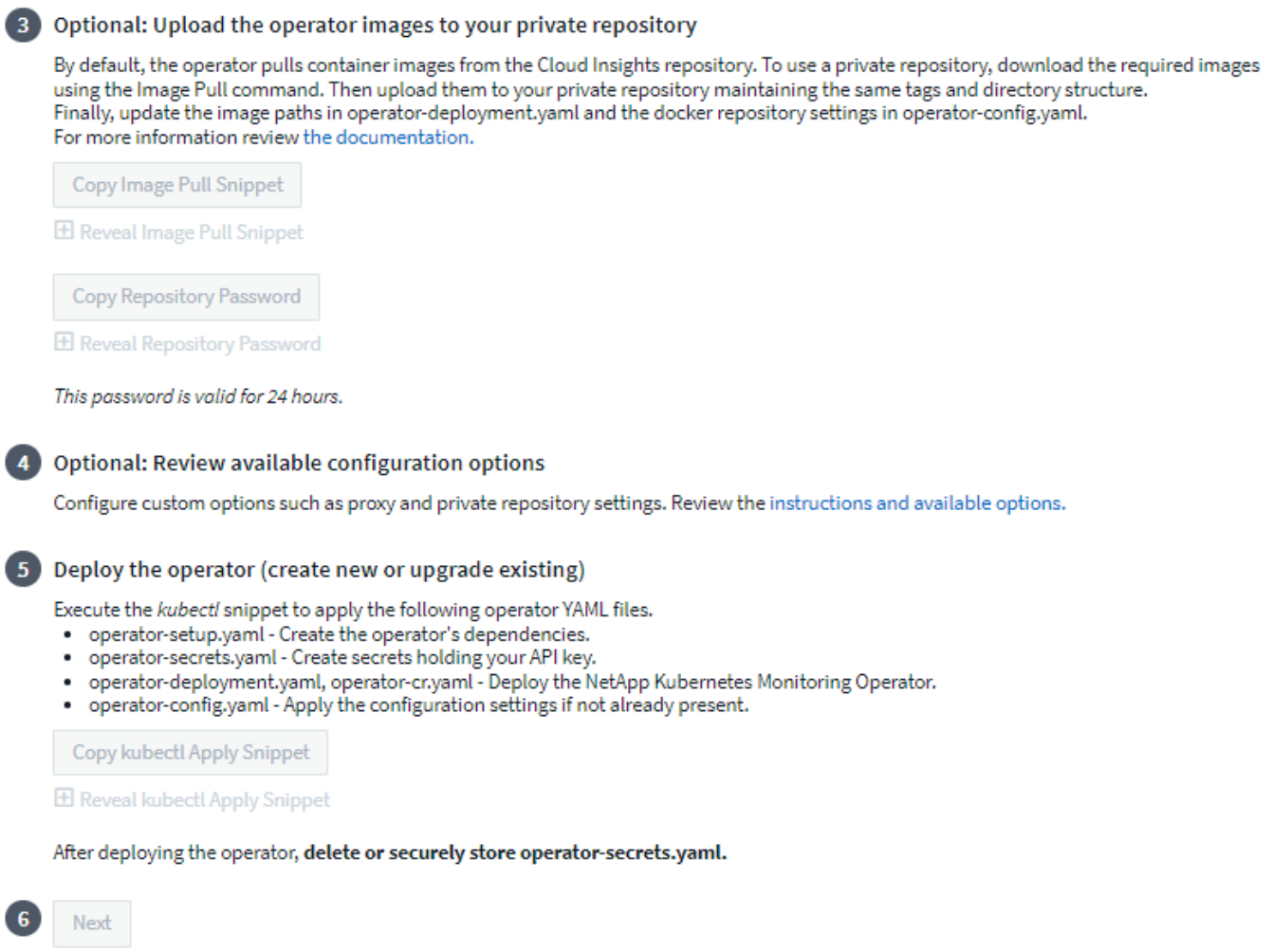
O Data Infrastructure Insights oferece a coleção **Operador de Monitoramento do Kubernetes** para Kubernetes. Navegue até **Kubernetes > Collectors > Kubernetes Collector** para implantar um novo operador.

Antes de instalar o operador de monitoramento do Kubernetes

Consulte "Pré-requisitos"a documentação antes de instalar ou atualizar o Operador de Monitoramento do Kubernetes.

Instalando o Operador de Monitoramento do Kubernetes


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
clustername
Namespace
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.



Etapas para instalar o agente do operador de monitoramento do Kubernetes no Kubernetes:

1. Insira um nome de cluster e um namespace exclusivos. Se você a atualizar é de um operador Kubernetes anterior, use o mesmo nome de cluster e namespace.
2. Uma vez que eles são inseridos, você pode copiar o snippet de comando de download para a área de transferência.
3. Cole o snippet em uma janela *bash* e execute-o. Os ficheiros de instalação do Operador serão transferidos. Observe que o snippet tem uma chave exclusiva e é válido por 24 horas.
4. Se você tiver um repositório personalizado ou privado, copie o trecho opcional Image Pull, cole-o em um shell *bash* e execute-o. Depois que as imagens tiverem sido puxadas, copie-as para o seu repositório privado. Certifique-se de manter as mesmas tags e estrutura de pastas. Atualize os caminhos em *operator-deployment.yaml*, bem como as configurações do repositório docker em *operator-config.yaml*.
5. Se desejar, revise as opções de configuração disponíveis, como proxy ou configurações de repositório privado. Você pode ler mais sobre "opções de configuração".
6. Quando estiver pronto, implante o Operador copiando o snippet de aplicação kubectl, baixando-o e executando-o.
7. A instalação prossegue automaticamente. Quando estiver concluído, clique no botão *Next*.
8. Quando a instalação estiver concluída, clique no botão *Next*. Certifique-se também de excluir ou armazenar com segurança o arquivo *operator-secrets.yaml*.

Se estiver usando um proxy, leia sobre configurando proxy.

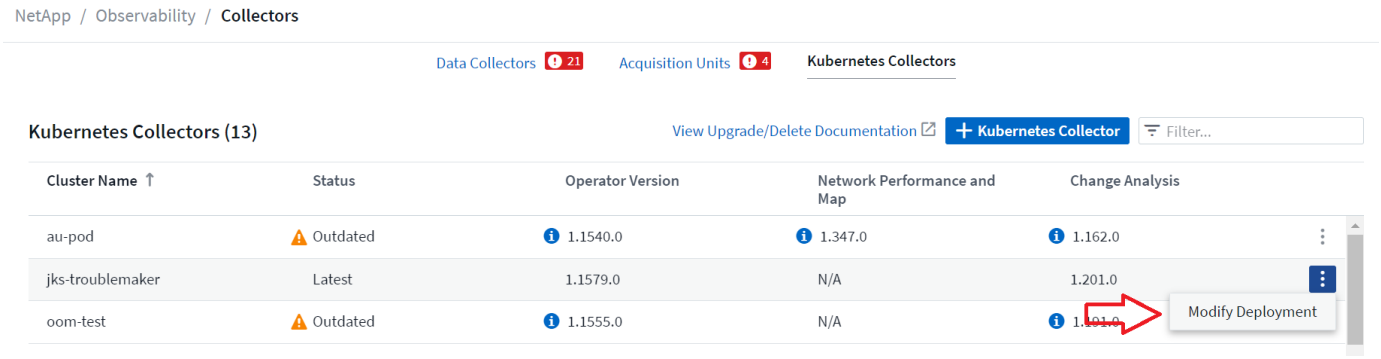
Se você tiver um repositório personalizado, leia sobre usando um repositório docker personalizado/privado.

Componentes de monitoramento do Kubernetes

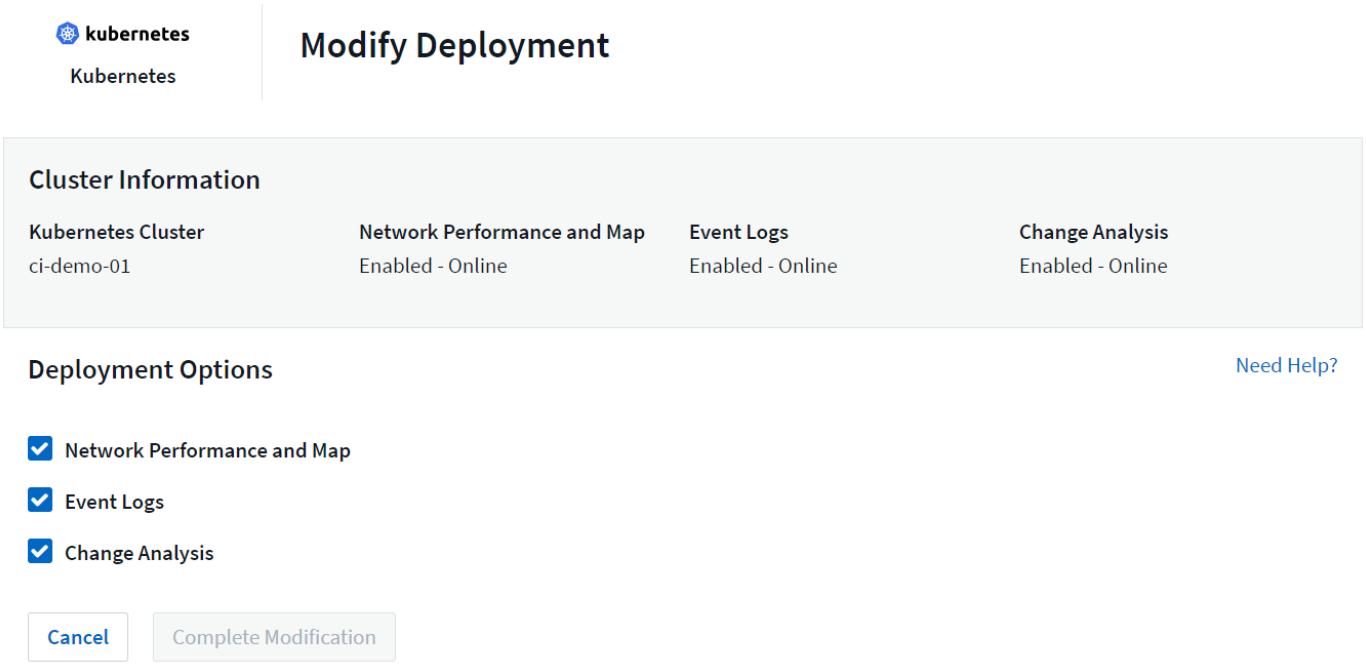
O monitoramento do Kubernetes do Data Infrastructure Insights é composto por quatro componentes de monitoramento:

- Métricas do cluster
- Desempenho de rede e mapa (opcional)
- Registos de eventos (opcional)
- Análise de mudança (opcional)

Os componentes opcionais acima são ativados por padrão para cada coletor do Kubernetes; se você decidir que não precisa de um componente para um coletor específico, você pode desativá-lo navegando para **Kubernetes > coletores** e selecionando *Modificar implantação* no menu "três pontos" do coletor à direita da tela.



O ecrã mostra o estado atual de cada componente e permite desativar ou ativar componentes para esse coletor, conforme necessário.



Atualização para o operador de monitoramento mais recente do Kubernetes

Determine se existe um AgentConfiguration com o Operador existente (se o seu namespace não for o *NetApp-monitoring* padrão, substitua o namespace apropriado):

```
 kubectl -n netapp-monitoring get agentconfiguration netapp-monitoring-
configuration
Se existir uma configuração AgentConfiguration:
```

- Instale O operador mais recente sobre o operador existente.
  - Certifique-se de que está puxando as imagens mais recentes do recipiente se estiver a utilizar um repositório personalizado.

Se o AgentConfiguration não existir:

- Anote o nome do cluster conforme reconhecido pelo Data Infrastructure Insights (se o namespace não for o monitoramento padrão do NetApp, substitua o namespace apropriado):

```
 kubectl -n netapp-monitoring get agent -o
jsonpath='{.items[0].spec.cluster-name}'
* Crie uma cópia de segurança do Operador existente (se o seu namespace
não for o NetApp-monitoring predefinido, substitua o namespace
apropriado):
```

```
 kubectl -n netapp-monitoring get agent -o yaml > agent_backup.yaml
* <<to-remove-the-kubernetes-monitoring-operator,Desinstalar>> O
operador existente.
* <<installing-the-kubernetes-monitoring-operator,Instale>> O operador
mais recente.
```

  - Use o mesmo nome de cluster.
  - Depois de baixar os arquivos YAML do Operador mais recentes, coloque as personalizações encontradas no Agent_backup.yaml para o operador-config.yaml baixado antes de implantar.
  - Certifique-se de que está puxando as imagens mais recentes do recipiente se estiver a utilizar um repositório personalizado.

Parando e iniciando o Operador de Monitoramento do Kubernetes

Para parar o operador de monitoramento do Kubernetes:

```
 kubectl -n netapp-monitoring scale deploy monitoring-operator
--replicas=0
Para iniciar o operador de monitoramento do Kubernetes:
```

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

Desinstalação

Para remover o operador de monitoramento do Kubernetes

Observe que o namespace padrão para o Operador de Monitoramento do Kubernetes é "NetApp-monitoring". Se você tiver definido seu próprio namespace, substitua esse namespace nesses e todos os comandos e arquivos subsequentes.

As versões mais recentes do operador de monitoramento podem ser desinstaladas com os seguintes comandos:

```
kubectl -n <NAMESPACE> delete agent -l installed-by=nkmo-<NAMESPACE>
kubectl -n <NAMESPACE> delete
clusterrole,clusterrolebinding,crd,svc,deploy,role,rolebinding,secret,sa
-l installed-by=nkmo-<NAMESPACE>
```

Se o operador de monitoramento foi implantado em seu próprio namespace dedicado, exclua o namespace:

```
 kubectl delete ns <NAMESPACE>
Se o primeiro comando retornar "nenhum recurso encontrado", use as
instruções a seguir para desinstalar versões mais antigas do operador de
monitoramento.
```

Execute cada um dos seguintes comandos em ordem. Dependendo da sua instalação atual, alguns desses comandos podem retornar mensagens "objeto não encontrado". Essas mensagens podem ser ignoradas com segurança.

```
kubectl -n <NAMESPACE> delete agent agent-monitoring-netapp
kubectl delete crd agents.monitoring.netapp.com
kubectl -n <NAMESPACE> delete role agent-leader-election-role
kubectl delete clusterrole agent-manager-role agent-proxy-role agent-
metrics-reader <NAMESPACE>-agent-manager-role <NAMESPACE>-agent-proxy-role
<NAMESPACE>-cluster-role-privileged
kubectl delete clusterrolebinding agent-manager-rolebinding agent-proxy-
rolebinding agent-cluster-admin-rolebinding <NAMESPACE>-agent-manager-
rolebinding <NAMESPACE>-agent-proxy-rolebinding <NAMESPACE>-cluster-role-
binding-privileged
kubectl delete <NAMESPACE>-psp-nkmo
kubectl delete ns <NAMESPACE>
```

Se uma restrição de contexto de segurança foi criada anteriormente:

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

Sobre o Kube-State-metrics

O Operador de Monitoramento do Kubernetes do NetApp instala suas próprias métricas de estado do kube para evitar conflitos com outras instâncias.

Para obter informações sobre métricas Kube-State, "esta página"consulte .

Configurar/personalizar o Operador

Essas seções contêm informações sobre como personalizar a configuração do operador, trabalhar com proxy, usar um repositório docker personalizado ou privado ou trabalhar com o OpenShift.

Opções de configuração

As configurações mais comumente modificadas podem ser configuradas no recurso personalizado *AgentConfiguration*. Você pode editar esse recurso antes de implantar o operador editando o arquivo *operator-config.yaml*. Este arquivo inclui exemplos comentados de configurações. Consulte a lista de "definições disponíveis" para obter a versão mais recente do operador.

Você também pode editar esse recurso depois que o operador tiver sido implantado usando o seguinte comando:

```
 kubectl -n netapp-monitoring edit AgentConfiguration
Para determinar se a versão implantada do operador suporta
AgentConfiguration, execute o seguinte comando:
```

```
 kubectl get crd agentconfigurations.monitoring.netapp.com
Se você vir uma mensagem "erro do servidor (NotFound)", seu operador deve
ser atualizado antes de poder usar o AgentConfiguration.
```

Configurando o suporte Proxy

Há dois lugares onde você pode usar um proxy em seu locatário para instalar o Operador de Monitoramento do Kubernetes. Estes podem ser os mesmos ou sistemas proxy separados:

- Proxy necessário durante a execução do snippet de código de instalação (usando "curl") para conetar o sistema onde o snippet é executado ao seu ambiente Data Infrastructure Insights
- Proxy necessário pelo cluster do Kubernetes de destino para se comunicar com seu ambiente Data Infrastructure Insights

Se você usar um proxy para um ou ambos, para instalar o Monitor operacional Kubernetes, primeiro você deve garantir que o proxy esteja configurado para permitir uma boa comunicação com o ambiente Insights da infraestrutura de dados. Se você tiver um proxy e puder acessar o Data Infrastructure Insights do servidor/VM a partir do qual deseja instalar o Operador, o proxy provavelmente estará configurado corretamente.

Para o proxy usado para instalar o Monitor operacional Kubernetes, antes de instalar o Operador, defina as variáveis de ambiente *http_proxy/https_proxy*. Para alguns ambientes proxy, você também pode precisar definir a variável *no_proxy environment*.

Para definir a(s) variável(s), execute as seguintes etapas em seu sistema **antes** de instalar o Operador de Monitoramento do Kubernetes:

1. Defina a(s) variável(s) de ambiente *https_proxy* e/ou *http_proxy* para o usuário atual:
   a. Se o proxy que está sendo configurado não tiver Autenticação (nome de usuário/senha), execute o seguinte comando:

```
 export https_proxy=<proxy_server>:<proxy_port>
.. Se o proxy que está sendo configurado tiver Autenticação (nome de
usuário/senha), execute este comando:
```

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

Para que o proxy usado para que o cluster do Kubernetes se comunique com o ambiente Data Infrastructure Insights, instale o Operador de Monitoramento do Kubernetes depois de ler todas essas instruções.

Configure a seção proxy do AgentConfiguration no operator-config.yaml antes de implantar o Operador de Monitoramento do Kubernetes.

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

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

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

Usando um repositório docker personalizado ou privado

Por padrão, o operador de monitoramento do Kubernetes coletará imagens de contentor do repositório Data Infrastructure Insights. Se você tiver um cluster do Kubernetes usado como destino para monitoramento e esse cluster estiver configurado para extrair apenas imagens de contentor de um repositório ou Registro de contentor personalizado ou privado do Docker, configure o acesso aos contentores necessários pelo Operador de Monitoramento do Kubernetes.

Execute o "trecho de recebimento de imagem" do bloco de instalação do Operador de Monitoramento do NetApp. Esse comando fará login no repositório Data Infrastructure Insights, extrairá todas as dependências de imagem do operador e fará logout do repositório Data Infrastructure Insights. Quando solicitado, insira a senha temporária do repositório fornecida. Este comando transfere todas as imagens utilizadas pelo operador, incluindo as funcionalidades opcionais. Veja abaixo quais recursos essas imagens são usadas.

Funcionalidade do operador principal e monitoramento do Kubernetes

- monitoramento de NetApp
- ci-kube-rbac-proxy
- ci-ksm
- ci-telegraf
- distroless-root-user

Registo de eventos

- ci-fluente-bit
- ci-kurein-event-exporter

Desempenho de rede e mapa

- ci-net-observador

Envie a imagem do docker do operador para o seu repositório docker privado/local/empresarial de acordo com suas políticas corporativas. Certifique-se de que as tags de imagem e os caminhos de diretório para essas imagens em seu repositório sejam consistentes com os do repositório Data Infrastructure Insights.

Edite a implantação do operador de monitoramento no operator-deployment.yaml e modifique todas as referências de imagem para usar seu repositório Docker privado.

```
image: <docker repo of the enterprise/corp docker repo>/ci-kube-rbac-
proxy:<ci-kube-rbac-proxy version>
image: <docker repo of the enterprise/corp docker repo>/netapp-
monitoring:<version>
```

Edite o AgentConfiguration no operator-config.yaml para refletir o novo local de repo do docker. Crie uma nova imagePullSecret para o seu repositório privado, para obter mais detalhes consulte *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
```

Instruções do OpenShift

Se você estiver executando no OpenShift 4,6 ou superior, você deve editar o AgentConfiguration em *operator-config.yaml* para ativar a configuração *runPrivileged*:

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

O OpenShift pode implementar um nível adicional de segurança que pode bloquear o acesso a alguns componentes do Kubernetes.

Tolerações e taints

O *NetApp-ci-telegraf-ds*, o *NetApp-CI-Fluent-bit-ds* e o *NetApp-CI-NET-Observer-L4-DS* DaemonSets devem agendar um pod em cada nó do cluster para coletar corretamente os dados em todos os nós. O operador foi

configurado para tolerar alguns **taints** conhecidos. Se você tiver configurado quaisquer taints personalizados em seus nós, impedindo assim que os pods sejam executados em cada nó, você poderá criar uma **tolerância** para essas taints ."Em *AgentConfiguration*" Se você tiver aplicado taints personalizados a todos os nós do cluster, também será necessário adicionar as tolerâncias necessárias à implantação do operador para permitir que o pod do operador seja agendado e executado.

Saiba mais sobre o Kubernetes "Taints e Tolerations".

Volte ao **"Página de Instalação do Operador de Monitoramento do Kubernetes do NetApp"**

Uma Nota sobre Segredos

Para remover a permissão do Operador de Monitoramento do Kubernetes para exibir segredos em todo o cluster, exclua os seguintes recursos do arquivo *operator-setup.yaml* antes de instalar:

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

Se for uma atualização, exclua também os recursos do cluster:

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

Se a análise de mudança estiver ativada, modifique o *AgentConfiguration* ou *operator-config.yaml* para descomentar a seção de gerenciamento de alterações e inclua *kindsToIgnoreFromWatch: '"segredos"'* na seção Gerenciamento de alterações. Observe a presença e a posição de aspas simples e duplas nesta linha.

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

Verificando assinaturas de imagem do Operador de Monitoramento do Kubernetes

A imagem para o operador e todas as imagens relacionadas que ele implanta são assinadas pelo NetApp. Você pode verificar manualmente as imagens antes da instalação usando a ferramenta de cografia ou configurar um controlador de admissão do Kubernetes. Para obter mais detalhes, consulte "Documentação do Kubernetes".

A chave pública usada para verificar as assinaturas de imagem está disponível no bloco de instalação do Operador de Monitoramento em *Opcional: Carregue as imagens do operador para o seu repositório privado >*

chave Pública de assinatura de imagem

Para verificar manualmente uma assinatura de imagem, execute as seguintes etapas:

1. Copie e execute o snippet de recebimento de imagem
2. Copie e insira a senha do repositório quando solicitado
3. Armazenar a chave Pública de assinatura de imagem (dii-image-signing.pub no exemplo)
4. Verifique as imagens usando o cosign. Consulte o exemplo a seguir de uso de cosign

```
$ cosign verify --key dii-image-signing.pub --insecure-ignore-sct
--insecure-ignore-tlog <repository>/<image>:<tag>
Verification for <repository>/<image>:<tag> --
The following checks were performed on each of these signatures:
  - The cosign claims were validated
  - The signatures were verified against the specified public key
[{"critical":{"identity":{"docker-
reference":"<repository>/<image>"},"image":{"docker-manifest-
digest":"sha256:<hash>"},"type":"cosign container image
signature"},"optional":null}]
```

Solução de problemas

Algumas coisas para tentar se você encontrar problemas para configurar o operador de monitoramento do Kubernetes:

Problema:	Tente isto:
Não vejo um hiperlink/conexão entre o meu volume persistente do Kubernetes e o dispositivo de armazenamento de back-end correspondente. Meu volume persistente do Kubernetes é configurado usando o nome de host do servidor de armazenamento.	Siga as etapas para desinstalar o agente Telegraf existente e reinstalar o agente Telegraf mais recente. Você precisa estar usando o Telegraf versão 2,0 ou posterior, e o storage de cluster do Kubernetes precisa ser monitorado ativamente pelo Data Infrastructure Insights.
Estou vendo mensagens nos logs que se assemelham ao seguinte: E0901 15 352:21 v1:39,962145 1 k8s reflector.go:178] k8s.io/kube-State-metrics/internal/store/builder.go:352: Falha ao listar *v1.MutatingWebhookConfiguration: O servidor não conseguiu encontrar o recurso solicitado E0901 15:k8s:43,168161 1 reflector.go:178] 21.io/kube-State-State-lease	Essas mensagens podem ocorrer se você estiver executando o kube-State-metrics versão 2.0.0 ou superior com versões do Kubernetes abaixo de 1,20. Para obter a versão do Kubernetes: *Kubectl version* para obter a versão do kube-State-metrics: *Kubectl get deploy/kube-State-metrics -o jsonpath leases'* para evitar que essas mensagens aconteçam, os usuários podem modificar sua implantação do kube-State-metrics para desativar os seguintes: *Mutatinghookhookhooks*

Problema:	Tente isto:
Vejo mensagens de erro do Telegraf semelhantes às seguintes, mas o Telegraf inicia e executa: Oct 11 14:23:41 ip-172-31-39-47 systemd[1]: Iniciou o agente de servidor orientado a plug-in para relatar métricas no InfluxDB. Oct 11 14:23:41 ip-172-31-39-47 telegraf[1827]: Tempo 2021-10-11T14:23:41Z" não foi possível criar o diretório de cache. /Etc/telegraf/.cache/floco de neve, err: Mkdir /etc/telegraf/.CA che: Permissão negada. Ignorado. Func"gostonflake.(*defaultLogger).Errorf" file "log.go:120" Oct:10 ip-23-41Z-39-47 telegraf[1827]: 2021" 31"erro de 11 14:23:41:172". Abra /etc/telegraf/.cache/floco de neve/ocsp_response_cache.json: nenhum arquivo ou diretório desse tipo de arquivo ou diretório.(*defaultLogger).Errorf" arquivo "log.go:120 23" Oct 2021 41Z:10 ip-172-31-39-47 telegraf[1827]: 11 14-23:41 A iniciar o Telegraf 1.19.3	Este é um problema conhecido. "Este artigo do GitHub"Consulte para obter mais detalhes. Enquanto o Telegraf estiver ativo e em execução, os usuários podem ignorar essas mensagens de erro.
No Kubernetes, meu(s) pod(s) Telegraf está relatando o seguinte erro: "Erro no processamento de informações de mountstats: Failed to open mountstats file: /Hostfs/proc/1/mountstats, error: Open /hostfs/proc/1/mountstats: Permission denied"	Se o SELinux estiver habilitado e aplicando, provavelmente impedirá que o(s) pod(s) Telegraf acesse o arquivo /proc/1/mountstats no nó Kubernetes. Para superar essa restrição, edite a configuração do agentConfiguration e ative a configuração RUNGED Privileged. Para obter mais detalhes, consulte "Instruções do OpenShift"a .
No Kubernetes, meu pod Telegraf ReplicaSet está relatando o seguinte erro: [inputs.prometheus] erro no plugin: Não foi possível carregar o par de chaves /etc/kupere/pki/etcd/Server.crt:/etc/kuGES/pki/etcd/Server.key: Open /etc/kuurge/pki/etcd/Server.crt: nenhum arquivo ou diretório	O pod Telegraf ReplicaSet destina-se a ser executado em um nó designado como mestre ou para o etcd. Se o pod ReplicaSet não estiver sendo executado em um desses nós, você receberá esses erros. Verifique se seus nós master/etcd têm manchetes neles. Se o fizerem, adicione as tolerâncias necessárias ao Telegraf ReplicaSet, telegraf-rs. Por exemplo, edite o ReplicaSet… kubectl edite rs telegraf-RS …e adicione as tolerâncias apropriadas à especificação. Em seguida, reinicie o pod ReplicaSet.
Tenho um ambiente PSP/PSA. Isso afeta meu operador de monitoramento?	Se o seu cluster Kubernetes estiver em execução com a Política de Segurança do Pod (PSP) ou a admissão de Segurança do Pod (PSA), você deverá fazer o upgrade para o Operador de Monitoramento do Kubernetes mais recente. Siga estes passos para atualizar para o Operador atual com suporte para PSP/PSA: 1. Desinstalar o operador de monitoramento anterior: kubectl delete agent-monitoring-NetApp -n NetApp-monitoring kubectl delete ns NetApp-monitoring kubectl delete crd agents.monitoring.NetApp.com kubectl delete clusterrole agent-manager-role agent-proxy-role agent-rolebinding cluster-rolebinding.-rolebinding 2. Instale a versão mais recente do operador de monitorização.

Problema:	Tente isto:
Deparei-me com problemas ao tentar implementar o Operador e tenho PSP/PSA em utilização.	1. Edite o agente usando o seguinte comando: Kubectl -n <name-space> edit Agent 2. Marque "Segurança-política-ativada" como "falsa". Isso desativará as políticas de Segurança do Pod e a admissão de Segurança do Pod e permitirá que o Operador implante. Confirme usando os seguintes comandos: Kubectl Get PSP (deve mostrar a Política de Segurança Pod removida) kubectl get all -n <namespace>
grep -i psp (deve mostrar que nada é encontrado)	Erros "ImagePullBackoff" vistos
Esses erros podem ser vistos se você tiver um repositório docker personalizado ou privado e ainda não tiver configurado o Operador de Monitoramento do Kubernetes para reconhecê-lo adequadamente. Leia mais sobre a configuração para repositório personalizado/privado.	Estou tendo um problema com a implantação do meu operador de monitoramento e a documentação atual não me ajuda a resolvê-lo.

Capture ou anote a saída dos comandos a seguir e entre em Contato com a equipe de suporte técnico.
`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`
	Os pods NET-Observer (Workload Map) no namespace Operator estão em CrashLoopBackOff

Esses pods correspondem ao coletor de dados do mapa de workload para observabilidade de rede. Tente estes: • Verifique os logs de um dos pods para confirmar a versão mínima do kernel. Por exemplo: ---- [ci-tenant-id":"your-tenant-id","Collector-cluster":"your-k8s-cluster-name","ambiente":"prod","nível":"erro","msg":"falhou na validação. Razão: A versão 3.10.0 do kernel é menor que a versão mínima do kernel de 4.18.0","Time":"2022-11-09T08:23:08Z" ---- • os pods do Net-Observer requerem que a versão do kernel do Linux seja pelo menos 4.18.0. Verifique a versão do kernel usando o comando "uname -r" e certifique-se de que eles são > 4.18.0	Os pods estão em execução no namespace do operador (padrão: Monitoramento NetApp), mas nenhum dado é exibido na IU para mapa de workload ou métricas do Kubernetes em consultas

Problema:	Tente isto:
Verifique a configuração de hora nos nós do cluster K8S. Para uma auditoria precisa e relatórios de dados, é altamente recomendável sincronizar a hora na máquina do agente usando o Network Time Protocol (NTP) ou o Simple Network Time Protocol (SNTP).	Alguns dos pods net-observer no namespace Operador estão no estado pendente
NET-Observer é um DaemonSet e executa um pod em cada nó do cluster k8s. • Observe o pod que está no estado pendente e verifique se ele está enfrentando um problema de recurso para CPU ou memória. Certifique-se de que a memória e a CPU necessárias estão disponíveis no nó.	Estou vendo o seguinte em meus logs imediatamente após instalar o Operador de Monitoramento do Kubernetes: [inputs.prometheus] erro no plugin: Erro ao fazer solicitação HTTP para http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: Get http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: Dial tcp: Lookup kube-State-metrics.<namespace>.svc.cluster.local: nenhum host
Normalmente, essa mensagem só é vista quando um novo operador é instalado e o pod *telegraf-rs* está ativo antes do pod *ksm* estar ativo. Essas mensagens devem parar quando todos os pods estiverem em execução.	Não vejo nenhuma métrica sendo coletada para os CronJobs do Kubernetes que existem no meu cluster.
Verifique a versão do Kubernetes (isto é `kubectl version`,). Se for v1,20.x ou inferior, esta é uma limitação esperada. A versão kube-State-metrics implantada com o Operador de Monitoramento do Kubernetes suporta apenas v1.CronJob. Com o Kubernetes 1,20.x e abaixo, o recurso CronJob está em v1beta.CronJob. Como resultado, as métricas de estado do kube não conseguem encontrar o recurso CronJob.	Depois de instalar o operador, os pods telegraf-ds entram em CrashLoopBackOff e os logs do pod indicam "su: Authentication failure".
Edite a seção telegraf em *AgentConfiguration* e defina *dockerMetricCollectionEnabled* como false. Para obter mais detalhes, consulte o "opções de configuração". … spec: … telegraf: … - Name: docker run-mode : - DaemonSet substituições: - Chave: DOCKER_UNIX_SOCK_PLACEHOLDER valor: unix:////run/docker.Sock … …	Vejo mensagens de erro repetitivas semelhantes às seguintes nos meus logs do Telegraf: E! [Agent] erro ao gravar em outputs.http: Post "/https://<tenant_url>/rest/v1/Lake/ingest/influxdb": Prazo de contexto excedido (Client.Timeout excedido enquanto aguarda cabeçalhos)
Edite a seção telegraf em *AgentConfiguration* e aumente *outputTimeout* para 10s. Para obter mais detalhes, consulte o "opções de configuração".	Estou faltando dados *involvedobject* para alguns Registros de eventos.
Certifique-se de que seguiu os passos indicados na "Permissões" secção acima.	Por que estou vendo dois pods de operador de monitoramento em execução, um chamado NetApp-CI-monitoring-operator-<pod> e o outro chamado Monitoring-operator-<pod>?

Problema:	Tente isto:
A partir de 12 de outubro de 2023, o Data Infrastructure Insights refatorou a operadora para melhor atender nossos usuários; para que essas alterações sejam totalmente adotadas, você retire o operador antigo deve e instale o novo.	Os eventos do meu kualves pararam inesperadamente de reportar ao Data Infrastructure Insights.
Recuperar o nome do pod de exportador de eventos: `kubectl -n netapp-monitoring get pods`	grep event-exporter
awk '{print $1}'	sed 's/event-exporter./event-exporter/'` Deve ser "NetApp-CI-event-exporter" ou "event-exporter". Em seguida, edite o agente de monitoramento `kubectl -n netapp-monitoring edit agent` e defina o valor para LOG_FILE para refletir o nome do pod de exportador de eventos apropriado encontrado na etapa anterior. Mais especificamente, LOG_FILE deve ser definido como "/var/log/containers/NetApp-CI-event-exporters.log" ou "/var/log/containers/event-exporters*.log" …. fluent-bit: … - name: event-exporter-ci substitutions: - key: LOG_FILE values: - /var/log/containers/netapp-ci-event-exporter*.log … …. Alternativamente, pode-se desinstalartambém e reinstale o agente.
Estou vendo POD(s) implantado(s) pelo Operador de Monitoramento do Kubernetes travarem devido a recursos insuficientes.	Consulte o Operador de Monitoramento do Kubernetes "opções de configuração"para aumentar os limites de CPU e/ou memória conforme necessário.
Uma imagem ausente ou uma configuração inválida fez com que os pods de métricas de estado do NetApp-ci-kube falhassem na inicialização ou se preparassem. Agora o StatefulSet está preso e as alterações de configuração não estão sendo aplicadas aos pods NetApp-CI-kube-State-metrics.	O StatefulSet está em um "quebrado" estado. Depois de corrigir quaisquer problemas de configuração, salte os pods NetApp-CI-kube-State-metrics.
Os pods de métricas de estado do NetApp-ci-kube falham ao iniciar depois de executar uma atualização do Operador do Kubernetes, lançando o ErrImagePull (falha ao puxar a imagem).	Tente redefinir os pods manualmente.

Problema:	Tente isto:
"Evento descartado como sendo mais antigo do que maxEventAgeSeconds" mensagens estão sendo observadas para o meu cluster Kubernetes em Log Analysis.	Modifique o Operador *agentConfiguration* e aumente o *event-exporter-maxEventAgeds* (ou seja, para 60s), *event-exporter-kubeQPS* (ou seja, para 100) e *event-exporter-kubeBurst* (ou seja, para 500). Para obter mais detalhes sobre essas opções de configuração, consulte a "opções de configuração" página.
Telegraf avisa ou trava por causa de memória bloqueável insuficiente.	Tente aumentar o limite de memória bloqueável para o Telegraf no sistema operacional/nó subjacente. Se aumentar o limite não for uma opção, modifique a configuração do agente NKMO e defina *desprotegido* como *true*. Isto instruirá o Telegraf a não tentar reservar páginas de memória bloqueadas. Embora isso possa representar um risco de segurança, pois segredos descriptografados podem ser trocados para o disco, ele permite a execução em ambientes onde não é possível reservar memória bloqueada. Para obter mais detalhes sobre as opções de configuração *desprotegidas*, consulte a "opções de configuração"página.
Vejo mensagens de aviso do Telegraf que se assemelham às seguintes: *W! [Inputs.diskio] não é possível reunir o nome do disco para "vdc": Erro ao ler /dev/vdc: nenhum arquivo ou diretório*	Para o Operador de Monitoramento do Kubernetes, essa mensagem de aviso é benigna e pode ser ignorada com segurança. Alternativamente, edite a seção telegraf em AgentConfiguration e defina *runDsPrivileged* como true. Para obter mais detalhes, consulte "opções de configuração do operador"a .

Problema:	Tente isto:

Meu pod fluente-bit está falhando com os seguintes erros: [2024 10/16 14/10/16 14 16:16 2024 23:23] [error] [/src/fluent-bit/plugins/in_tail/tail_fs_inotify.c:360 errno.24] muitos arquivos abertos [2024/10/16 14:16:23] [error] falha na inicialização tail,0 [Engine] [input]	Tente alterar suas configurações *fsnotify* no cluster:
	`sudo sysctl fs.inotify.max_user_instances (take note of setting)`
	`sudo sysctl fs.inotify.max_user_instances=<something larger than current setting>`
	`sudo sysctl fs.inotify.max_user_watches (take note of setting)`
	`sudo sysctl fs.inotify.max_user_watches=<something larger than current setting>`
	Reinicie o Fluent-bit.
	Observação: Para tornar essas configurações persistentes entre as reinicializações do nó, você precisa colocar as seguintes linhas em */etc/sysctl.conf*
	`fs.inotify.max_user_instances=<something larger than current setting>`
	`fs.inotify.max_user_watches=<something larger than current setting>`

Informações adicionais podem ser encontradas na "Suporte" página ou no "Matriz de suporte do Data Collector".

Opções de configuração do operador de monitoramento Kubernetes

A "Operador de monitoramento do Kubernetes" configuração pode ser personalizada.

A tabela abaixo lista as opções possíveis para o arquivo *AgentConfiguration*:

Componente	Opção	Descrição
agente		Opções de configuração comuns a todos os componentes que o operador pode instalar. Estes podem ser considerados como opções "globais".
	DockerRepo	Uma substituição dockerRepo para extrair imagens de repositórios docker privados de clientes em comparação com o repositório docker Data Infrastructure Insights. O padrão é repositório docker do Data Infrastructure Insights
	DockerImagePullSecret	Opcional: Um segredo para o repositório privado dos clientes
	Nome exclusivo	Campo de texto livre que identifica exclusivamente um cluster em todos os clusters de clientes. Isso deve ser exclusivo para um locatário do Data Infrastructure Insights. O padrão é o que o cliente insere na IU para o campo "Nome do cluster"
	Proxy Format: Proxy: Server: Port: Username: Password: NoProxy: IsTelegrafProxyEnabled: IsAuProxyEnabled: IsFluentbitProxyEnabled: IsCollectorProxyEnabled:	Opcional para definir proxy. Este é geralmente o proxy corporativo do cliente.
telegraf		Opções de configuração que podem personalizar a instalação do telegraf do Operador
	ColeçãoInterval	Intervalo de coleta de métricas, em segundos (máx. 60s)
	DsCpuLimit	Limite de CPU para telegraf ds
	DsMemLimit	Limite de memória para telegraf ds
	DsCpuRequest	Pedido de CPU para telegraf ds
	DsMemRequest	Pedido de memória para telegraf ds
	RsCpuLimit	Limite de CPU para telegraf rs
	RsMemLimit	Limite de memória para telegraf rs
	RsCpuRequest	Pedido de CPU para telegraf rs
	RsMemRequest	Pedido de memória para telegraf rs
	Privilégios em execução	Execute o recipiente *telegraf-monstats-poller* do DaemonSet no modo privilegiado. Defina isso como verdadeiro se o SELinux estiver habilitado em seus nós do Kubernetes.
	RunDsPrivileged	Defina runDsPrivileged como true para executar o contentor telegraf DaemonSet no modo privilegiado.
	BatchSize	Consulte "Documentação de configuração do Telegraf"

Componente	Opção	Descrição
	BufferLimit	Consulte "Documentação de configuração do Telegraf"
	RoundInterval	Consulte "Documentação de configuração do Telegraf"
	ColeçãoJitter	Consulte "Documentação de configuração do Telegraf"
	precisão	Consulte "Documentação de configuração do Telegraf"
	FlushInterval	Consulte "Documentação de configuração do Telegraf"
	FlushJitter	Consulte "Documentação de configuração do Telegraf"
	OutputTimeout (tempo limite de saída)	Consulte "Documentação de configuração do Telegraf"
	DsTolerations	tolerâncias adicionais do telegraf-ds.
	RsTolerations	tolerâncias adicionais telegraf-rs.
	SkipProcessorsAfterAggregadores	Consulte "Documentação de configuração do Telegraf"
	não protegido	Consulte este "Problema de Telegraf conhecido". A configuração *desprotegido* instruirá o Operador de Monitoramento do Kubernetes a executar o Telegraf com o `--unprotected` sinalizador.
kube-state-metrics		Opções de configuração que podem personalizar a instalação de métricas de estado kube do Operador
	CpuLimit	Limite de CPU para implantação de métricas de estado do kube
	MemLimit	Limite de MEM para implantação de métricas de estado do kube
	CpuRequest	Solicitação de CPU para implantação de métricas de estado do kube
	MemRequest	Solicitação de MEM para implantação de métricas de estado do kube
	recursos	uma lista separada por vírgulas de recursos a serem capturados. exemplo: cronjobs,daemonsets,deployments,ingresses,jobs,namespaces,nodos,persisttentvolumeclaims, persistentvolumes,pods,replicasets,resourcequotas,serviços,statfulsets
	tolerâncias	tolerações adicionais de métricas de estado do kube.
	etiquetas	uma lista separada por vírgulas de recursos que kube-state-metrics deve capturar

Componente	Opção	Descrição
registos		Opções de configuração que podem personalizar a coleta de logs e a instalação do Operador
	ReadFromHead	verdadeiro/falso, deve o bit fluente ler o log da cabeça
	tempo limite	tempo limite, em segundos
	Modo dnsMode	TCP/UDP, modo para DNS
	tolerações de bits fluentes	tolerações adicionais fluent-bit-ds.
	tolerância de evento-exportador	tolerância adicional ao exportador de eventos.
	Evento-exportador-maxEventAgeSeconds	idade máxima do evento exportador de eventos. Consulte https://github.com/jkroepke/resmoio-kubernetes-event-exporter
mapa de workload		Opções de configuração que podem personalizar a coleta e instalação do mapa de carga de trabalho do Operador.
	CpuLimit	Limite de CPU para NET Observer ds
	MemLimit	limite mem para o observador líquido ds
	CpuRequest	Pedido de CPU para NET Observer ds
	MemRequest	pedido de mem para net observer ds
	MetricAggregationInterval	intervalo de agregação métrica, em segundos
	BpfPollInterval	Intervalo de enquete BPF, em segundos
	EnableDNSLookup	Verdadeiro/falso, ative a pesquisa de DNS
	l4-tolerâncias	net-observer-l4-ds tolerâncias adicionais.
	Privilégios em execução	True/FALSE - defina Privileged como true se o SELinux estiver habilitado em seus nós do Kubernetes.
gerenciamento de alterações		Opções de configuração para o Gerenciamento e análise de alterações do Kubernetes
	CpuLimit	Limite de CPU para change-observer-Watch-rs
	MemLimit	Limite MEM para mudança-observador-server-rs
	CpuRequest	Pedido de CPU para change-observer-Watch-rs
	MemRequest	pedido de mem para mudança-observador-watch-rs
	FailureDeclaraçãoIntervalMins	Intervalo em minutos após o qual uma implantação não-bem-sucedida de uma carga de trabalho será marcada como falhou
	DeployAggrIntervalSeconds	Frequência na qual os eventos em andamento de implantação da carga de trabalho são enviados
	NonWorkloadAggrIntervalSeconds	Frequência na qual implantações que não são de carga de trabalho são combinadas e enviadas

Componente	Opção	Descrição
	TermsToRedact	Um conjunto de expressões regulares usadas em nomes env e mapas de dados cujo valor será editado termos de exemplo:"pwd", "password", "token", "apikey", "api-key", "jwt"
	AdicionalKindsToWatch	Uma lista separada por vírgulas de tipos adicionais para assistir do conjunto padrão de tipos observados pelo coletor
	KindsToIgnoreFromWatch	Uma lista separada por vírgulas de tipos a ignorar da observação do conjunto padrão de tipos observados pelo coletor
	LogRecordAggrIntervalSeconds	Frequência com a qual os Registros de log são enviados para IC do coletor
	tolerâncias de relógio	tolerâncias adicionais do change-observer-watch-ds. Apenas formato de linha única abreviada. Exemplo: Tecla: taint1, operador: Existe, efeito: NoSchedule, tecla: taint2, operador: Existe, efeito: NoExecute'

Exemplo de arquivo AgentConfiguration

Abaixo está um exemplo de arquivo *AgentConfiguration*.

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

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

    # # Proxy settings. The proxy that the operator should use to send
metrics to Data Infrastructure Insights.
    # # Please see documentation here: https://docs.netapp.com/us-
```

```
en/cloudinsights/task_config_telegraf_agent_k8s.html#configuring-proxy-
support
    # proxy:
    #   server:
    #   port:
    #   noproxy:
    #   username:
    #   password:
    #   isTelegrafProxyEnabled:
    #   isFluentbitProxyEnabled:
    #   isCollectorsProxyEnabled:

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

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

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

    # # The default time telegraf will wait between inputs for all plugins
(interval). Max=60
    # collectionInterval: '60s'
    # # Maximum number of records per output that telegraf will write in
one batch (metric_batch_size).
    # batchSize: '10000'
    # # Maximum number of records per output that telegraf will cache
pending a successful write (metric_buffer_limit).
```

```
    # bufferLimit: '150000'
    # # Collect metrics on multiples of interval (round_interval).
    # roundInterval: 'true'
    # # Each plugin waits a random amount of time between the scheduled
collection time and that time + collection_jitter before collecting inputs
(collection_jitter).
    # collectionJitter: '0s'
    # # Collected metrics are rounded to the precision specified. When set
to "0s" precision will be set by the units specified by interval
(precision).
    # precision: '0s'
    # # Time telegraf will wait between writing outputs (flush_interval).
Max=collectionInterval
    # flushInterval: '60s'
    # # Each output waits a random amount of time between the scheduled
write time and that time + flush_jitter before writing outputs
(flush_jitter).
    # flushJitter: '0s'
    # # Timeout for writing to outputs (timeout).
    # outputTimeout: '5s'

    # # telegraf-ds CPU/Mem limits and requests.
    # # See https://kubernetes.io/docs/concepts/configuration/manage-
resources-containers/
    # dsCpuLimit: '750m'
    # dsMemLimit: '800Mi'
    # dsCpuRequest: '100m'
    # dsMemRequest: '500Mi'

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

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

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

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

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

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

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

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

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

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

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

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

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

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

    # # Comma-separated list of metrics to enable.
    # # See metric-allowlist in https://github.com/kubernetes/kube-state-
metrics/blob/main/docs/cli-arguments.md
    # metrics:
'kube_cronjob_created,kube_cronjob_status_active,kube_cronjob_labels,kube_
daemonset_created,kube_daemonset_status_current_number_scheduled,kube_daem
onset_status_desired_number_scheduled,kube_daemonset_status_number_availab
le,kube_daemonset_status_number_misscheduled,kube_daemonset_status_number_
ready,kube_daemonset_status_number_unavailable,kube_daemonset_status_obser
ved_generation,kube_daemonset_status_updated_number_scheduled,kube_daemons
et_metadata_generation,kube_daemonset_labels,kube_deployment_status_replic
as,kube_deployment_status_replicas_available,kube_deployment_status_replic
as_unavailable,kube_deployment_status_replicas_updated,kube_deployment_sta
tus_observed_generation,kube_deployment_spec_replicas,kube_deployment_spec
_paused,kube_deployment_spec_strategy_rollingupdate_max_unavailable,kube_d
eployment_spec_strategy_rollingupdate_max_surge,kube_deployment_metadata_g
eneration,kube_deployment_labels,kube_deployment_created,kube_job_created,
kube_job_owner,kube_job_status_active,kube_job_status_succeeded,kube_job_s
tatus_failed,kube_job_labels,kube_job_status_start_time,kube_job_status_co
mpletion_time,kube_namespace_created,kube_namespace_labels,kube_namespace_
status_phase,kube_node_info,kube_node_labels,kube_node_role,kube_node_spec
_unschedulable,kube_node_created,kube_persistentvolume_capacity_bytes,kube
_persistentvolume_status_phase,kube_persistentvolume_labels,kube_persisten
tvolume_info,kube_persistentvolume_claim_ref,kube_persistentvolumeclaim_ac
cess_mode,kube_persistentvolumeclaim_info,kube_persistentvolumeclaim_label
s,kube_persistentvolumeclaim_resource_requests_storage_bytes,kube_persiste
ntvolumeclaim_status_phase,kube_pod_info,kube_pod_start_time,kube_pod_comp
letion_time,kube_pod_owner,kube_pod_labels,kube_pod_status_phase,kube_pod_
status_ready,kube_pod_status_scheduled,kube_pod_container_info,kube_pod_co
ntainer_status_waiting,kube_pod_container_status_waiting_reason,kube_pod_c
ontainer_status_running,kube_pod_container_state_started,kube_pod_containe
```

```
r_status_terminated,kube_pod_container_status_terminated_reason,kube_pod_c
ontainer_status_last_terminated_reason,kube_pod_container_status_ready,kub
e_pod_container_status_restarts_total,kube_pod_overhead_cpu_cores,kube_pod
_overhead_memory_bytes,kube_pod_created,kube_pod_deletion_timestamp,kube_p
od_init_container_info,kube_pod_init_container_status_waiting,kube_pod_ini
t_container_status_waiting_reason,kube_pod_init_container_status_running,k
ube_pod_init_container_status_terminated,kube_pod_init_container_status_te
rminated_reason,kube_pod_init_container_status_last_terminated_reason,kube
_pod_init_container_status_ready,kube_pod_init_container_status_restarts_t
otal,kube_pod_status_scheduled_time,kube_pod_status_unschedulable,kube_pod
_spec_volumes_persistentvolumeclaims_readonly,kube_pod_container_resource_
requests_cpu_cores,kube_pod_container_resource_requests_memory_bytes,kube_
pod_container_resource_requests_storage_bytes,kube_pod_container_resource_
requests_ephemeral_storage_bytes,kube_pod_container_resource_limits_cpu_co
res,kube_pod_container_resource_limits_memory_bytes,kube_pod_container_res
ource_limits_storage_bytes,kube_pod_container_resource_limits_ephemeral_st
orage_bytes,kube_pod_init_container_resource_limits_cpu_cores,kube_pod_ini
t_container_resource_limits_memory_bytes,kube_pod_init_container_resource_
limits_storage_bytes,kube_pod_init_container_resource_limits_ephemeral_sto
rage_bytes,kube_pod_init_container_resource_requests_cpu_cores,kube_pod_in
it_container_resource_requests_memory_bytes,kube_pod_init_container_resour
ce_requests_storage_bytes,kube_pod_init_container_resource_requests_epheme
ral_storage_bytes,kube_replicaset_status_replicas,kube_replicaset_status_r
eady_replicas,kube_replicaset_status_observed_generation,kube_replicaset_s
pec_replicas,kube_replicaset_metadata_generation,kube_replicaset_labels,ku
be_replicaset_created,kube_replicaset_owner,kube_resourcequota,kube_resour
cequota_created,kube_service_info,kube_service_labels,kube_service_created
,kube_service_spec_type,kube_statefulset_status_replicas,kube_statefulset_
status_replicas_current,kube_statefulset_status_replicas_ready,kube_statef
ulset_status_replicas_updated,kube_statefulset_status_observed_generation,
kube_statefulset_replicas,kube_statefulset_metadata_generation,kube_statef
ulset_created,kube_statefulset_labels,kube_statefulset_status_current_revi
sion,kube_statefulset_status_update_revision,kube_node_status_capacity,kub
e_node_status_allocatable,kube_node_status_condition,kube_pod_container_re
source_requests,kube_pod_container_resource_limits,kube_pod_init_container
_resource_limits,kube_pod_init_container_resource_requests'

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

    # # A set of regular expressions used in env names and data maps whose
value will be redacted
    # termsToRedact: '"pwd", "password", "token", "apikey", "api-key",
"api_key", "jwt", "accesskey", "access_key", "access-key", "ca-file",
"key-file", "cert", "cafile", "keyfile", "tls", "crt", "salt",
".dockerconfigjson", "auth", "secret"'

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

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

    # # A comma separated list of kinds to ignore from watching from the
default set of kinds watched by the collector
```

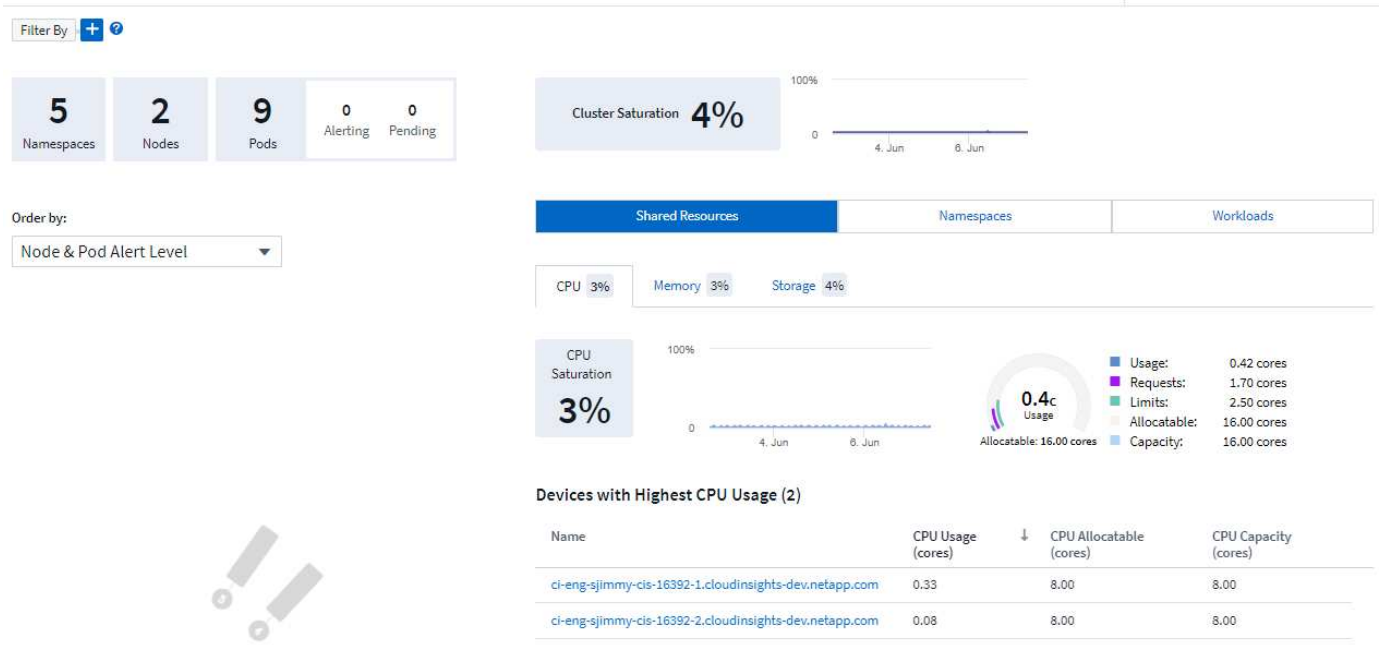
```
    # # Each kind will have to be prefixed by its apigroup
    # # Example: '"networking.k8s.io.networkpolicies,batch.jobs",
"authorization.k8s.io.subjectaccessreviews"'
    # kindsToIgnoreFromWatch: ''

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

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

Página de detalhes do cluster do Kubernetes

A página de detalhes do cluster do Kubernetes exibe uma visão geral detalhada do cluster do Kubernetes.



Contagens de namespace, nó e Pod

As contagens na parte superior da página mostram o número total de namespaces, nós e pods no cluster, bem como o número de popds que estão atualmente alertando e pendentes.

Recursos compartilhados e saturação

No canto superior direito da página de detalhes está a saturação do cluster como uma porcentagem atual,

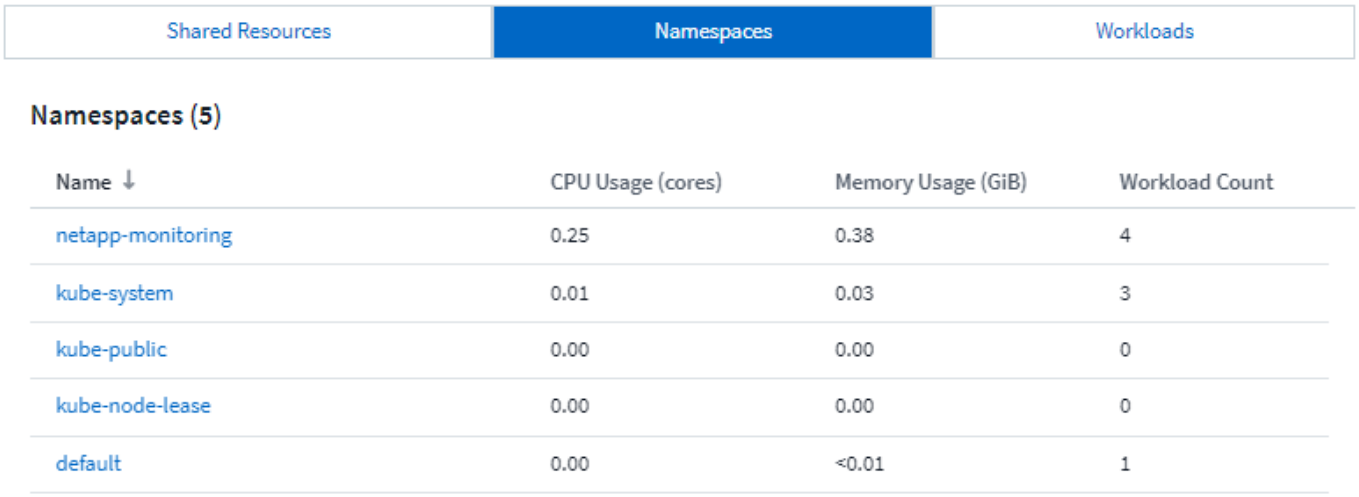
bem como um gráfico mostrando a tendência recente ao longo do tempo. A saturação de cluster é a mais alta da CPU, memória ou saturação de armazenamento em cada ponto do tempo.

Abaixo disso, a página mostra por padrão o uso de **recursos compartilhados**, com guias para CPU, memória e armazenamento. Cada guia mostra a porcentagem de saturação e a tendência ao longo do tempo, com detalhes de uso adicionais. Para armazenamento, o valor mostrado é o maior da saturação de backend e sistema de arquivos, que são calculados independentemente.

Os dispositivos com a maior utilização são apresentados numa tabela na parte inferior. Clique em qualquer link para explorar esses dispositivos.

Namespaces

A guia namespaces exibe uma lista de todos os namespaces em seu ambiente Kubernetes, mostrando o uso de CPU e memória, bem como uma contagem de cargas de trabalho em cada namespace. Clique nos links Nome para explorar cada namespace.



Workloads

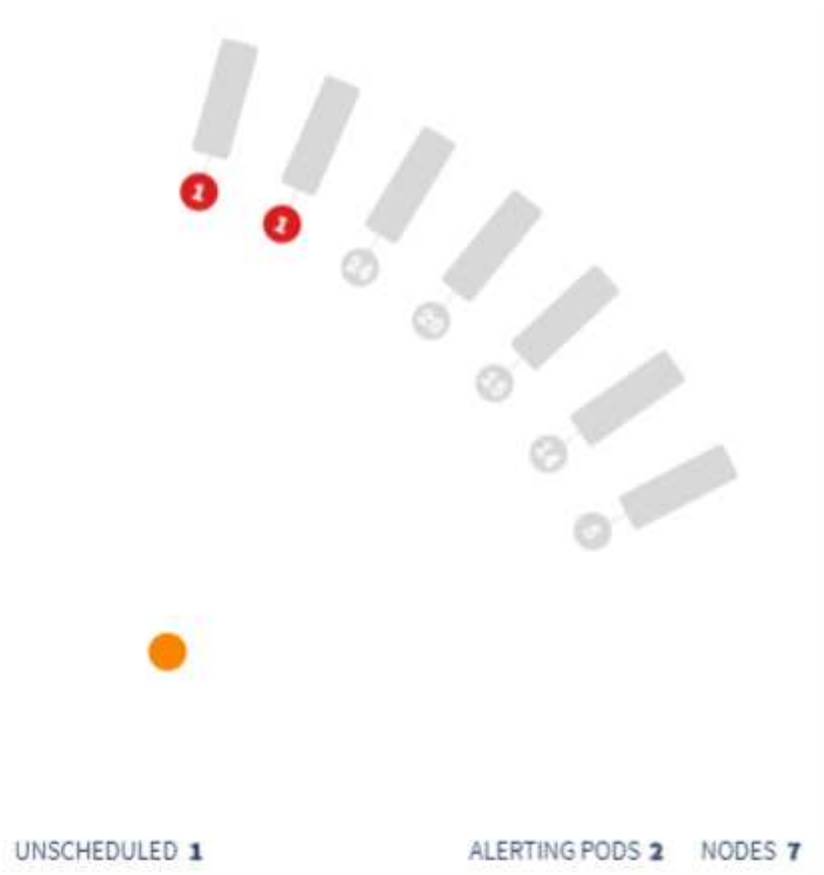
Da mesma forma, a guia cargas de trabalho exibe uma lista das cargas de trabalho em cada namespace, mostrando novamente o uso da CPU e da memória. Clicar nos links de namespace faz drill em cada um.

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

A "roda" do grupo

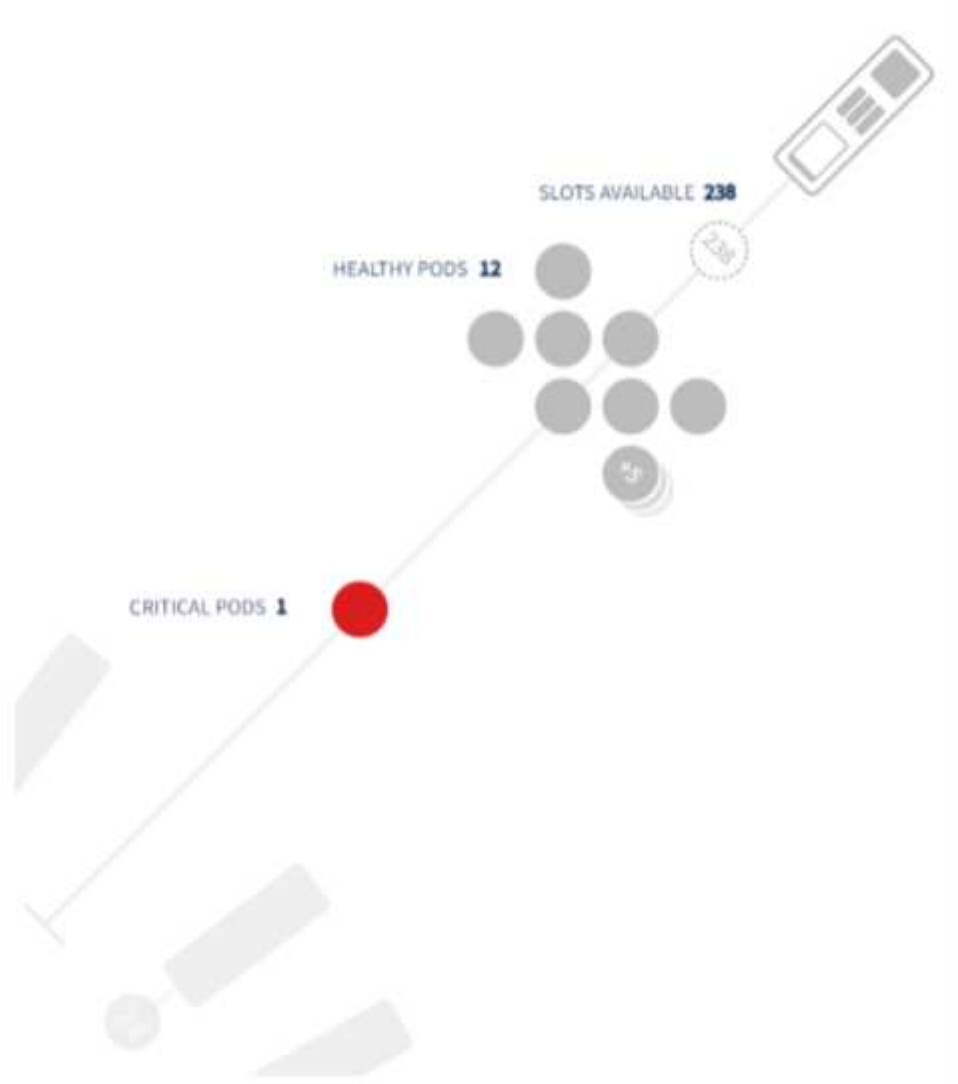


A seção "roda" do cluster fornece o estado do nó e do pod em um relance, que você pode detalhar para obter mais informações. Se o cluster contiver mais nós do que podem ser exibidos nesta área da página, você poderá girar a roda usando os botões disponíveis.

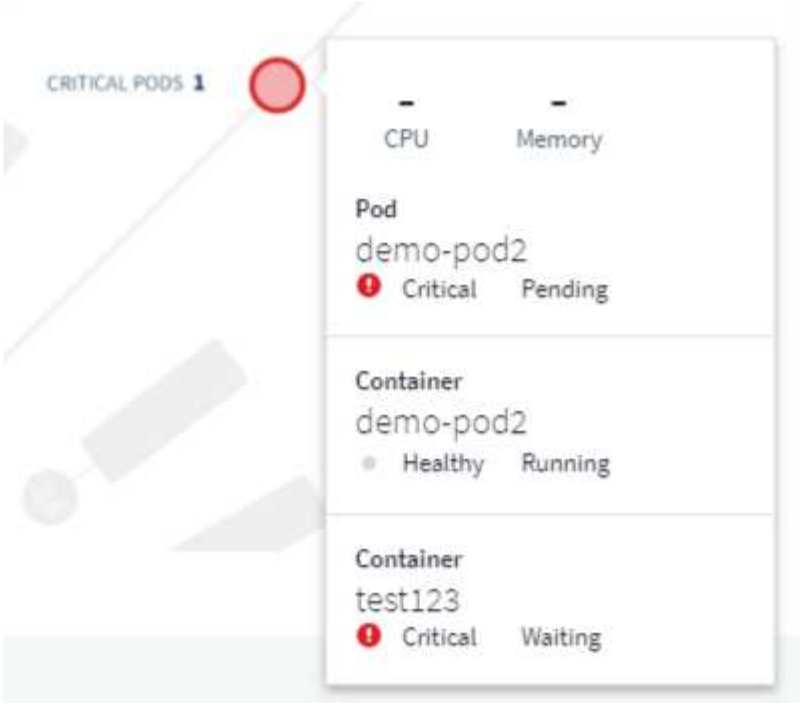
Os pods ou nós de alerta são exibidos em vermelho. As áreas de "aviso" são apresentadas a laranja. Os pods não programados (ou seja, não anexados) serão exibidos no canto inferior da "roda" do cluster.

Passar o Mouse sobre um pod (círculo) ou Node (barra) estenderá a visão do nó.

Clicar no pod ou nó nessa exibição aumentará o zoom para a exibição nó expandido.



A partir daqui, você pode passar o Mouse sobre um elemento para exibir detalhes sobre esse elemento. Por exemplo, passar o Mouse sobre o pod crítico neste exemplo exibe detalhes sobre esse pod.



Você pode visualizar informações de sistema de arquivos, memória e CPU passando o Mouse sobre os elementos Node.

Uma nota sobre os medidores

Os medidores de memória e CPU mostram três cores, uma vez que mostram *used* em relação à capacidade_allocatable_ e *total Capacity*.

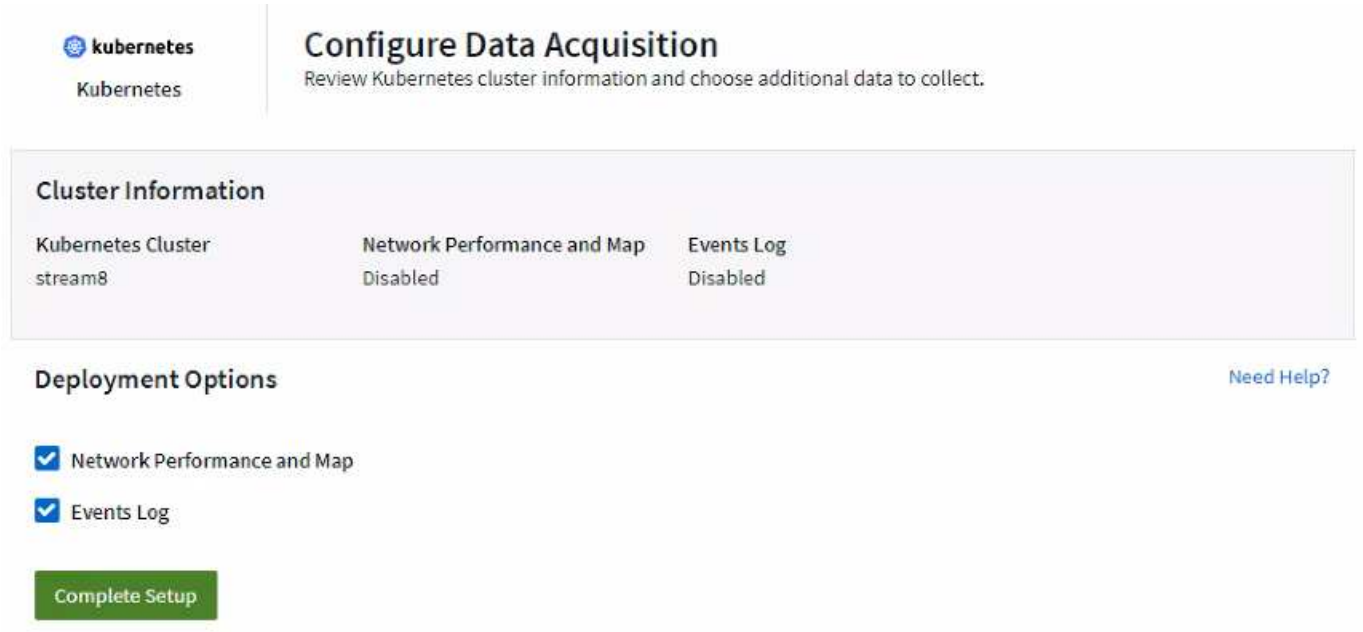
Monitoramento e mapa de desempenho de rede do Kubernetes

O recurso Kubernetes Network Performance Monitoring and Map simplifica a solução de problemas mapeando dependências entre serviços (também chamadas de cargas de trabalho) e oferece visibilidade em tempo real das latências de desempenho e anomalias de rede para identificar problemas de desempenho antes que eles afetem os usuários. Essa funcionalidade ajuda as organizações a reduzir os custos gerais analisando e auditando os fluxos de tráfego do Kubernetes.

Principais recursos: • O mapa de carga de trabalho apresenta dependências e fluxos de carga de trabalho do Kubernetes e destaca problemas de rede e desempenho. • Monitore o tráfego de rede entre pods, cargas de trabalho e nós do Kubernetes; identifique a origem dos problemas de latência e tráfego. • Reduzir os custos gerais analisando o tráfego de rede de entrada, saída, cross-region e cross-zone.

Pré-requisitos

Para poder usar o monitoramento e o mapa de desempenho de rede do Kubernetes, você deve ter configurado o "Operador de monitoramento do Kubernetes do NetApp" para ativar essa opção. Durante a implementação do Operador, selecione a caixa de verificação "Network Performance and Map" (desempenho da rede e mapa) para ativar. Você também pode habilitar essa opção navegando para uma página inicial do Kubernetes e selecionando "Modificar implantação".



Monitores

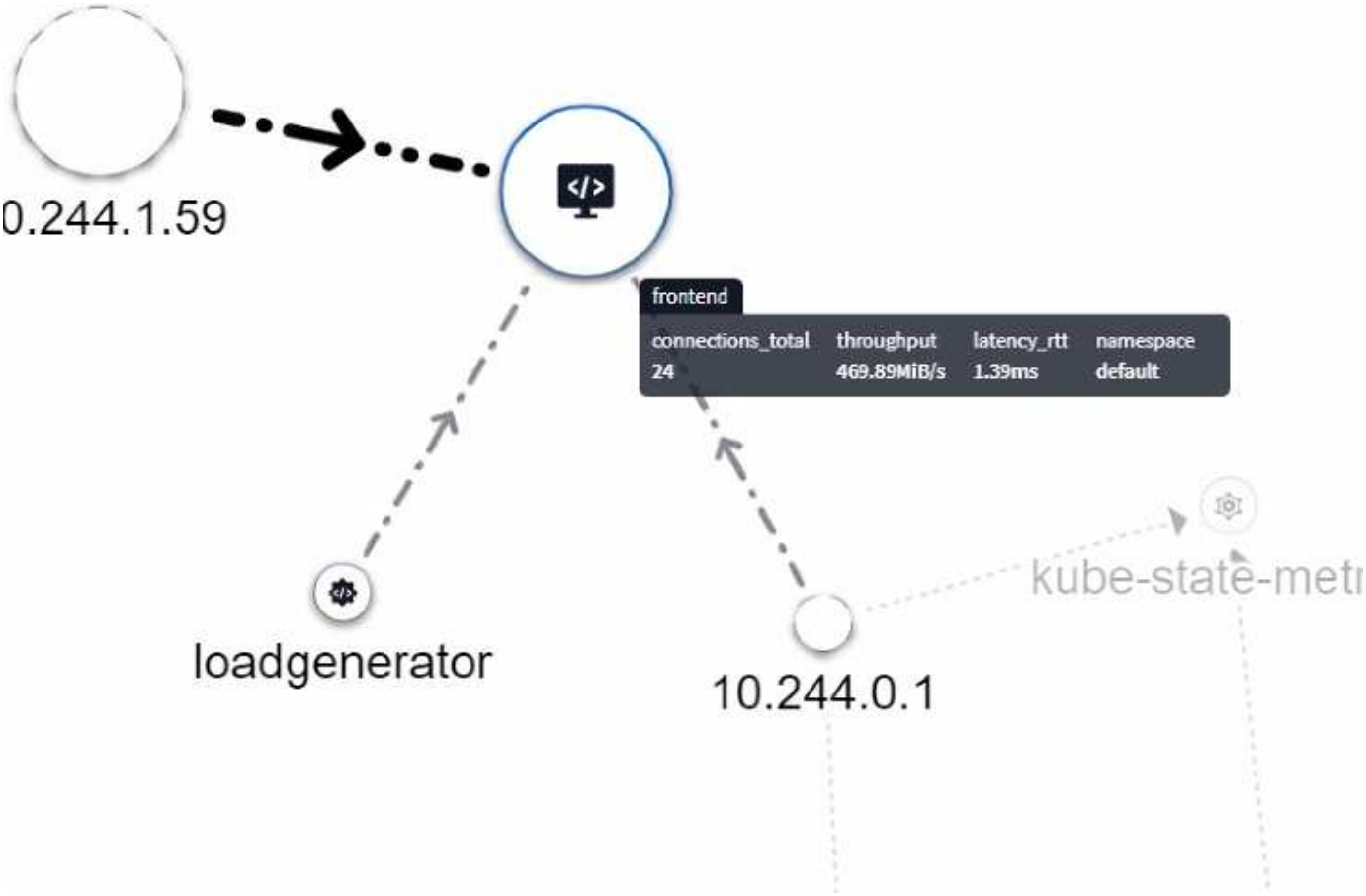
O mapa de carga de trabalho utiliza "monitores" para obter informações. O Data Infrastructure Insights fornece uma série de monitores padrão do Kubernetes (observe que eles podem ser *Pausado* por padrão. Você pode *Resume* (ou seja, ativar) os monitores que você deseja), ou você pode criar monitores personalizados para objetos kujuntos, que o mapa de carga de trabalho também usará.

Você pode criar alertas de métricas do Data Infrastructure Insights em qualquer um dos tipos de objeto abaixo. Certifique-se de que os dados estão agrupados pelo tipo de objeto padrão.

- kuseurea.workflow
- kuasse.daemonset
- kubernetes.deployment
- kuseurs.cronjob
- kuasse.job
- kuseixos.replicaset
- kuasse.statefulset
- kuasse.pod
- kubernetes.network_traffic_l4

O mapa

O mapa mostra os serviços/cargas de trabalho e seus relacionamentos entre si. Setas mostram direções de tráfego. Passar o Mouse sobre uma carga de trabalho exibe informações resumidas para essa carga de trabalho, como você pode ver neste exemplo:



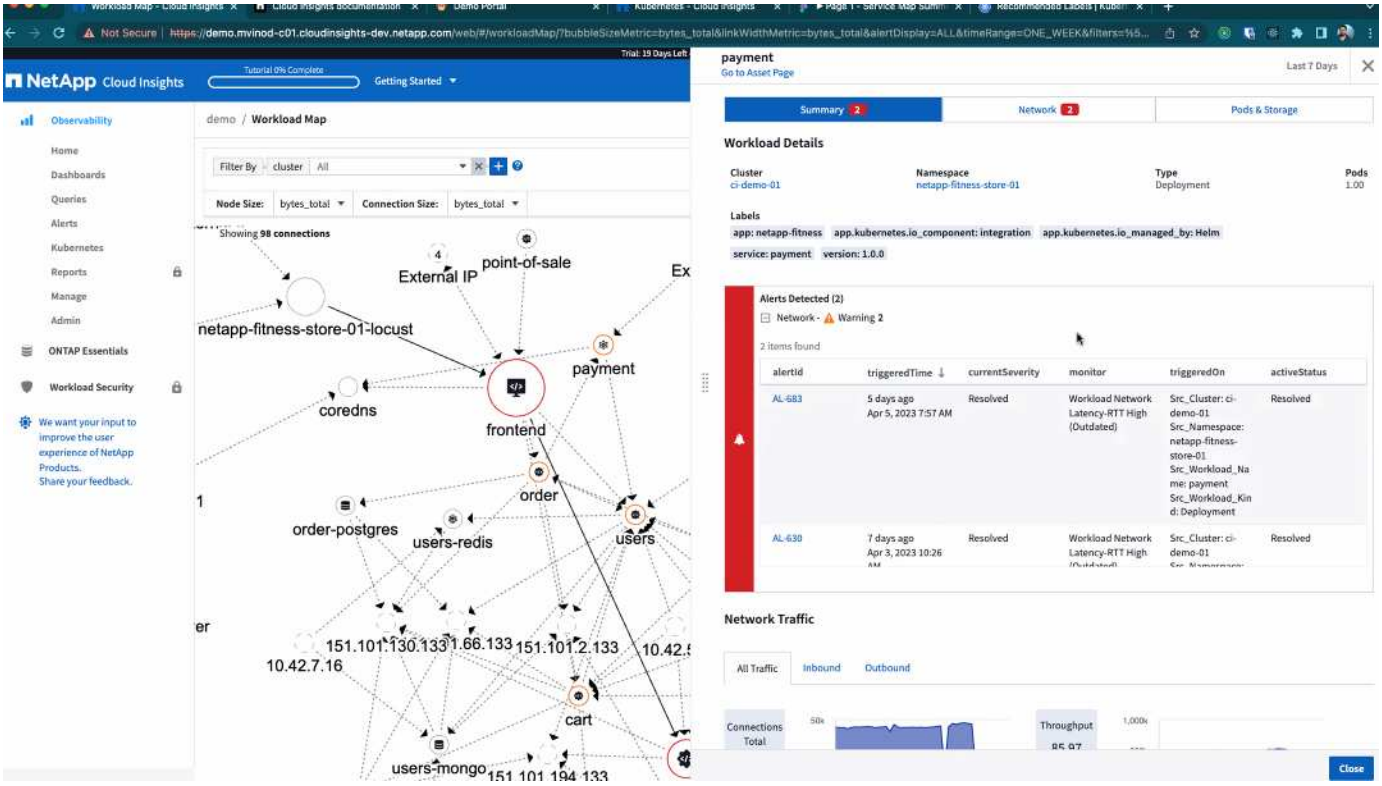
Os ícones dentro dos círculos representam diferentes tipos de serviço. Observe que os ícones só serão visíveis se os objetos subjacentes tiverem etiquetas.

O tamanho de cada círculo indica o tamanho do nó. Observe que esses tamanhos são relativos, o nível de zoom do navegador ou o tamanho da tela podem afetar os tamanhos reais dos círculos. Da mesma forma, o estilo de linha de tráfego oferece uma visão rápida do tamanho da conexão; linhas sólidas arrojadas são tráfego alto, enquanto linhas pontilhadas de luz são tráfego mais baixo.

Os números dentro dos círculos são o número de conexões externas que estão sendo processadas pelo serviço.

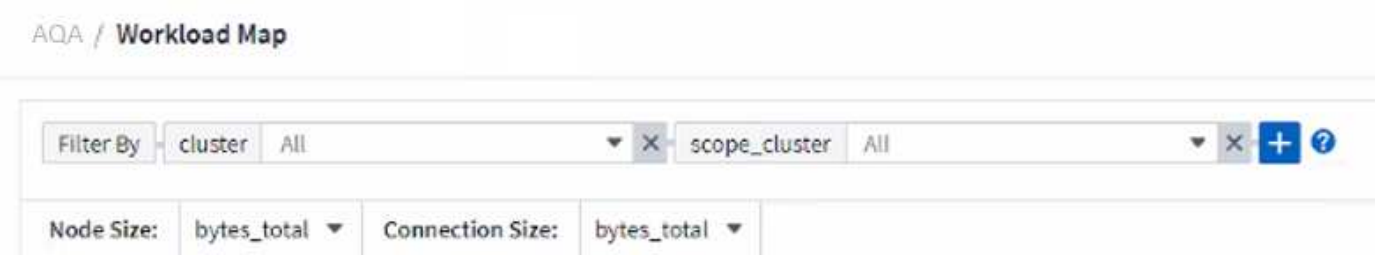
Detalhes e alertas do workload

Os círculos exibidos em cores indicam um alerta de nível crítico ou de aviso para a carga de trabalho. Passe o Mouse sobre o círculo para obter um resumo do problema ou clique no círculo para abrir um painel de deslizamento com mais detalhes.

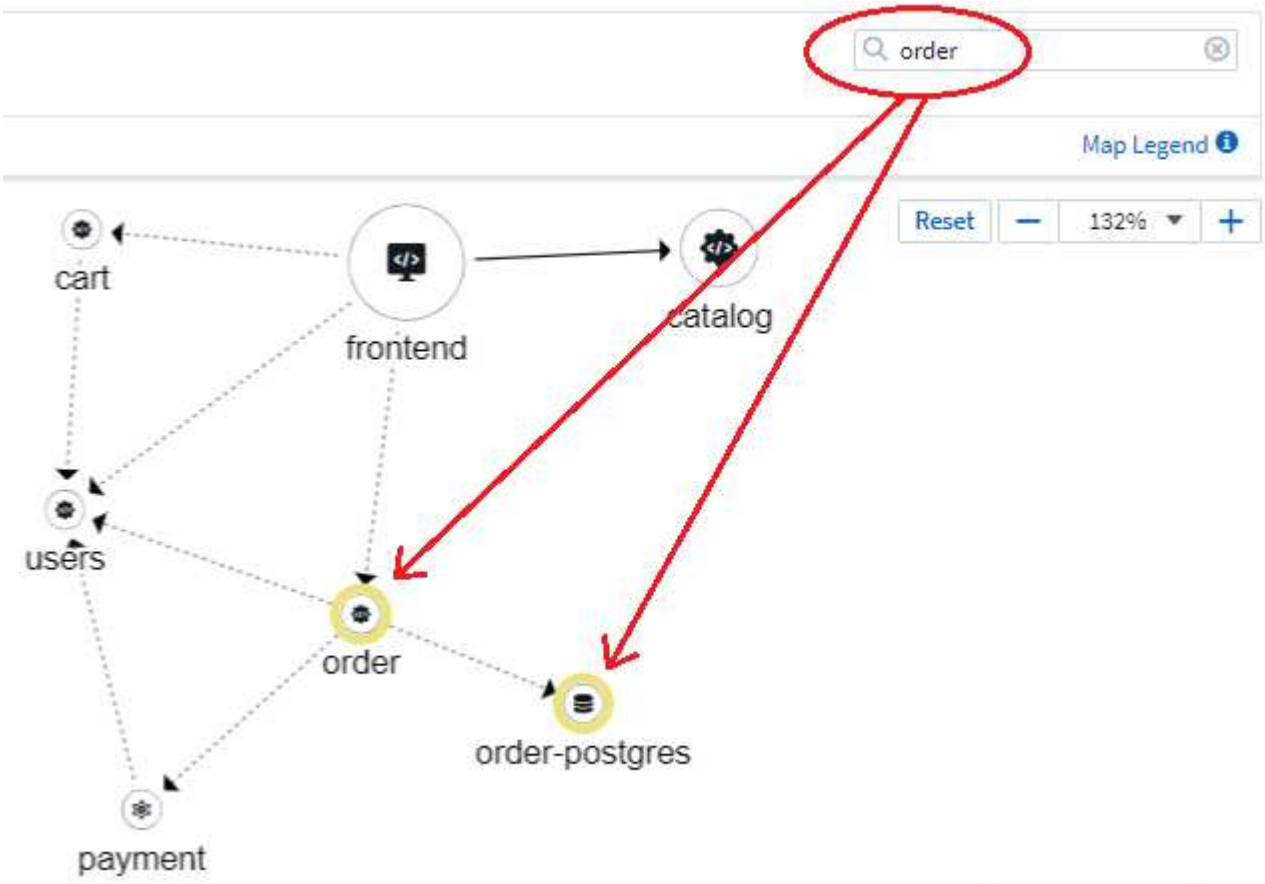


Encontrando e filtrando

Como acontece com outros recursos do Data Infrastructure Insights, você pode facilmente definir filtros para se concentrar nos objetos específicos ou atributos de carga de trabalho desejados.



Da mesma forma, digitar uma string no campo *find* destacará as cargas de trabalho correspondentes.



Etiquetas de workload

As etiquetas de carga de trabalho são necessárias se pretender que o mapa identifique os tipos de cargas de trabalho apresentadas (ou seja, os ícones de círculo). As etiquetas são derivadas da seguinte forma:

- Nome do serviço/aplicativo em execução em termos genéricos
- Se a fonte for um pod:
  - A etiqueta é derivada da etiqueta da carga de trabalho do pod
  - Etiqueta esperada na carga de trabalho: App.kureau.io/component
  - Referência do nome da etiqueta: https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels/
  - Etiquetas recomendadas:
    - frontend

- back-end
- banco de dados
- cache
- fila de espera
- kafka

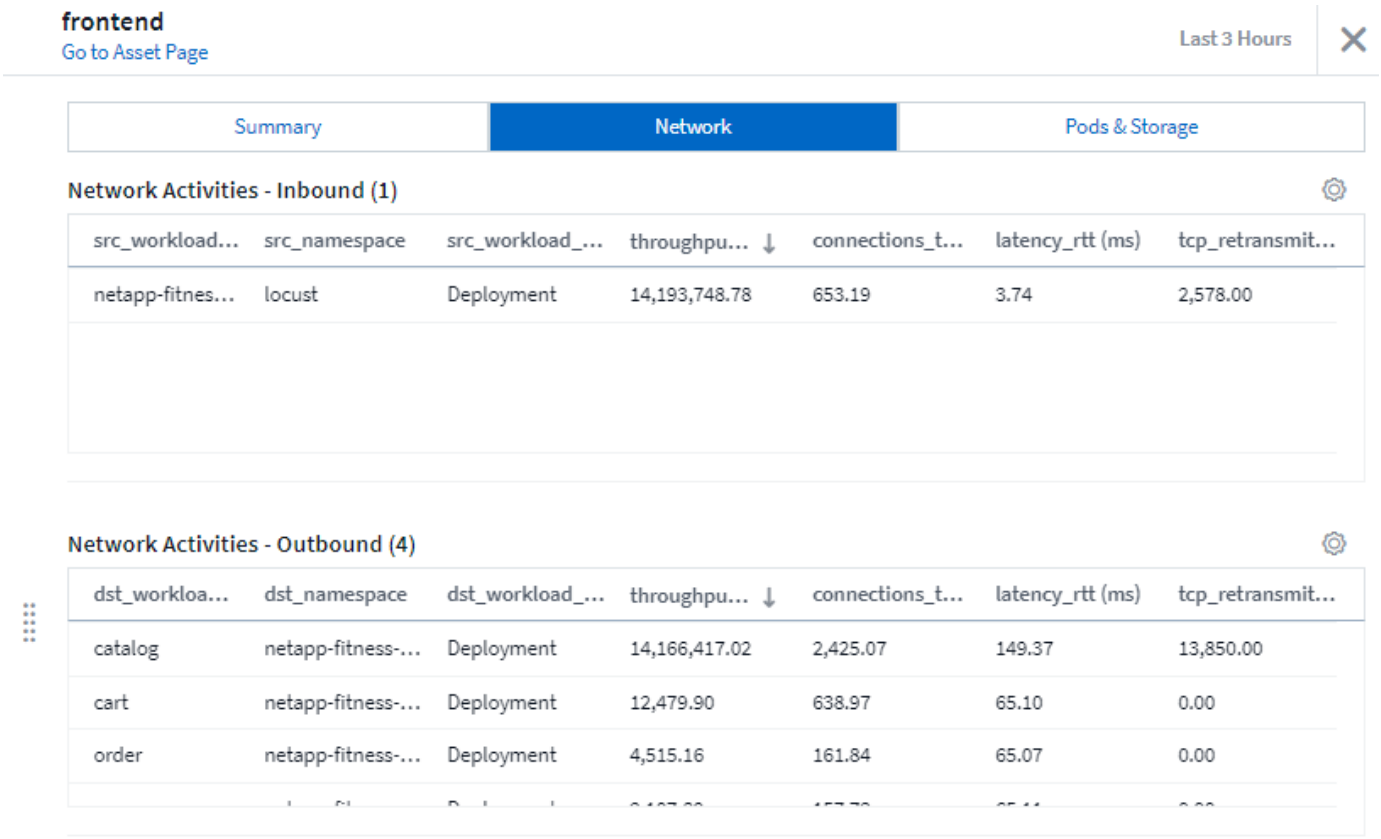
- Se a fonte for externa ao cluster kualves:
  - O Data Infrastructure Insights tentará analisar o nome DNS resolvido para extrair o tipo de serviço.

    Por exemplo, com um nome DNS resolvido de *s3.eu-north-1.amazonaws.com*, o nome resolvido é analisado para obter *S3* como o tipo de serviço.

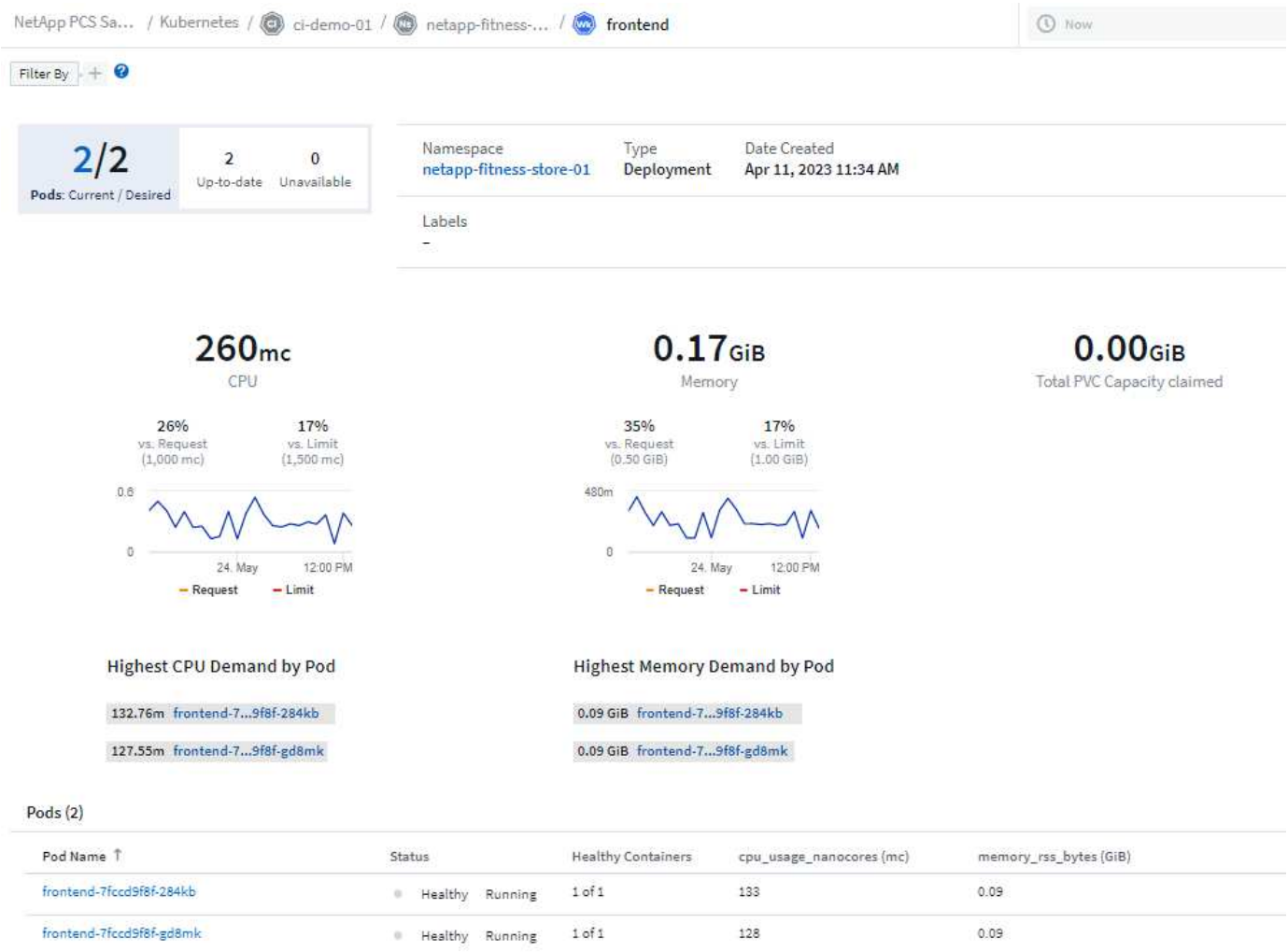
Mergulhe fundo

Clicar com o botão direito do Mouse em uma carga de trabalho apresenta opções adicionais para explorar ainda mais. Por exemplo, a partir daqui, você pode aumentar o zoom para ver as conexões para essa carga de trabalho.

Ou você pode abrir o painel deslizante de detalhes para visualizar diretamente a guia *Summary*, *Network* ou *Pod & Storage*.



Finalmente, selecionar *Go to Asset Page* abrirá a página de destino detalhada do ativo para a carga de trabalho.



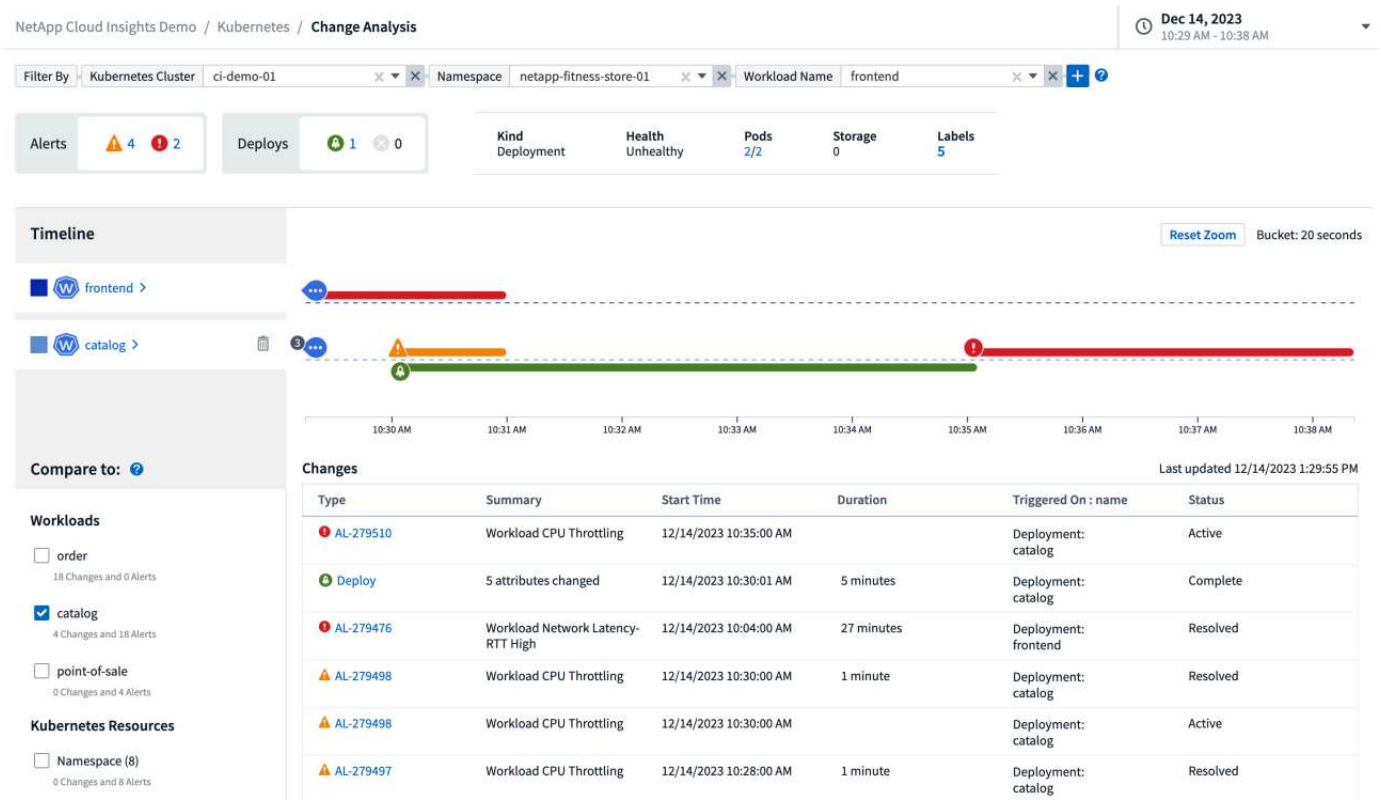
Kubernetes Change Analytics

O Kubernetes Change Analytics oferece uma visão completa das mudanças recentes no seu ambiente K8s. Alertas e status de implantação estão ao seu alcance. Com o Change Analytics, você pode controlar todas as alterações de implantação e configuração e correlacioná-las com a integridade e a performance dos serviços, da infraestrutura e dos clusters do K8s.

Como a análise de mudanças ajuda?

- Em ambientes Kubernetes de vários locatários, as interrupções podem ocorrer devido a alterações mal configuradas. O Change Analytics ajuda com isso, fornecendo um único painel para visualizar e correlacionar a integridade dos workloads e as alterações de configuração. Isso pode ajudar na solução de problemas de ambientes dinâmicos do Kubernetes.

Para exibir o Kubernetes Change Analytics, navegue até **Kubernetes > Change Analysis**.



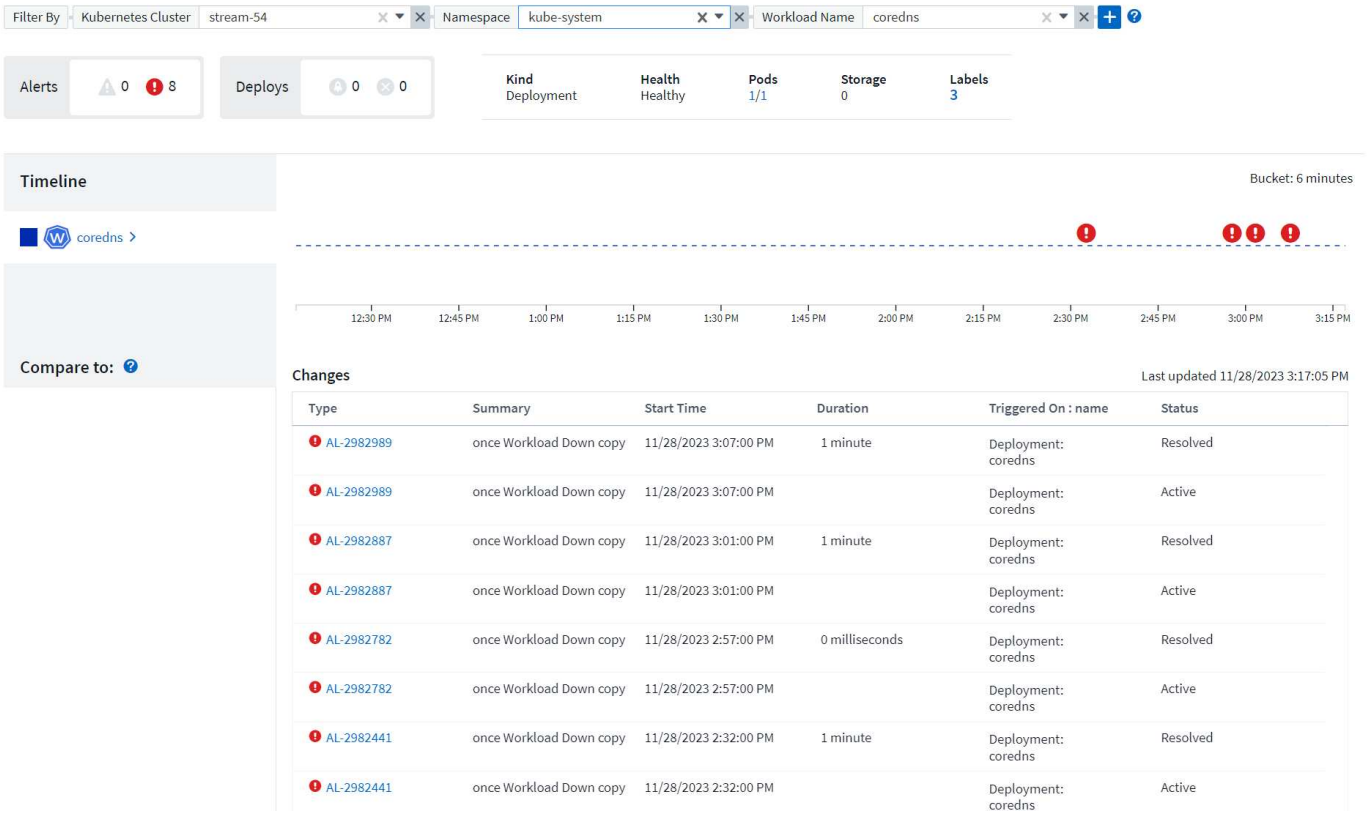
.

A página é atualizada automaticamente com base no intervalo de tempo selecionado no momento. Intervalos de tempo menores significam uma atualização mais frequente do ecrã.

Filtragem

Assim como todos os recursos do Data Infrastructure Insights, filtrar a lista de alterações é intuitivo: Na parte superior da página, insira ou selecione valores para seu cluster, namespace ou workload do Kubernetes.

Ao filtrar para um cluster, namespace e workload específicos (juntamente com quaisquer outros filtros definidos), você verá uma linha do tempo das implantações e alertas para essa carga de trabalho nesse namespace no cluster. Aumente ainda mais o zoom clicando e arrastando no gráfico para focar em um intervalo de tempo mais específico.

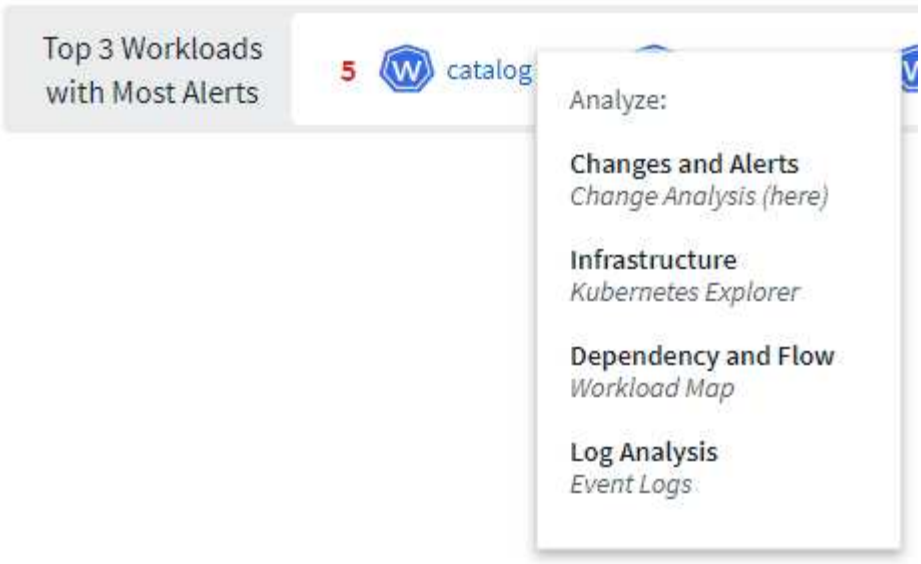


Estado rápido

Abaixo da área de filtragem estão vários indicadores de alto nível. À esquerda está o número de alertas (Aviso e crítico). Este número inclui alertas *ative*, bem como *resolved*. Para ver apenas alertas *ative*, defina um filtro para "Status" e escolha "Ativo".

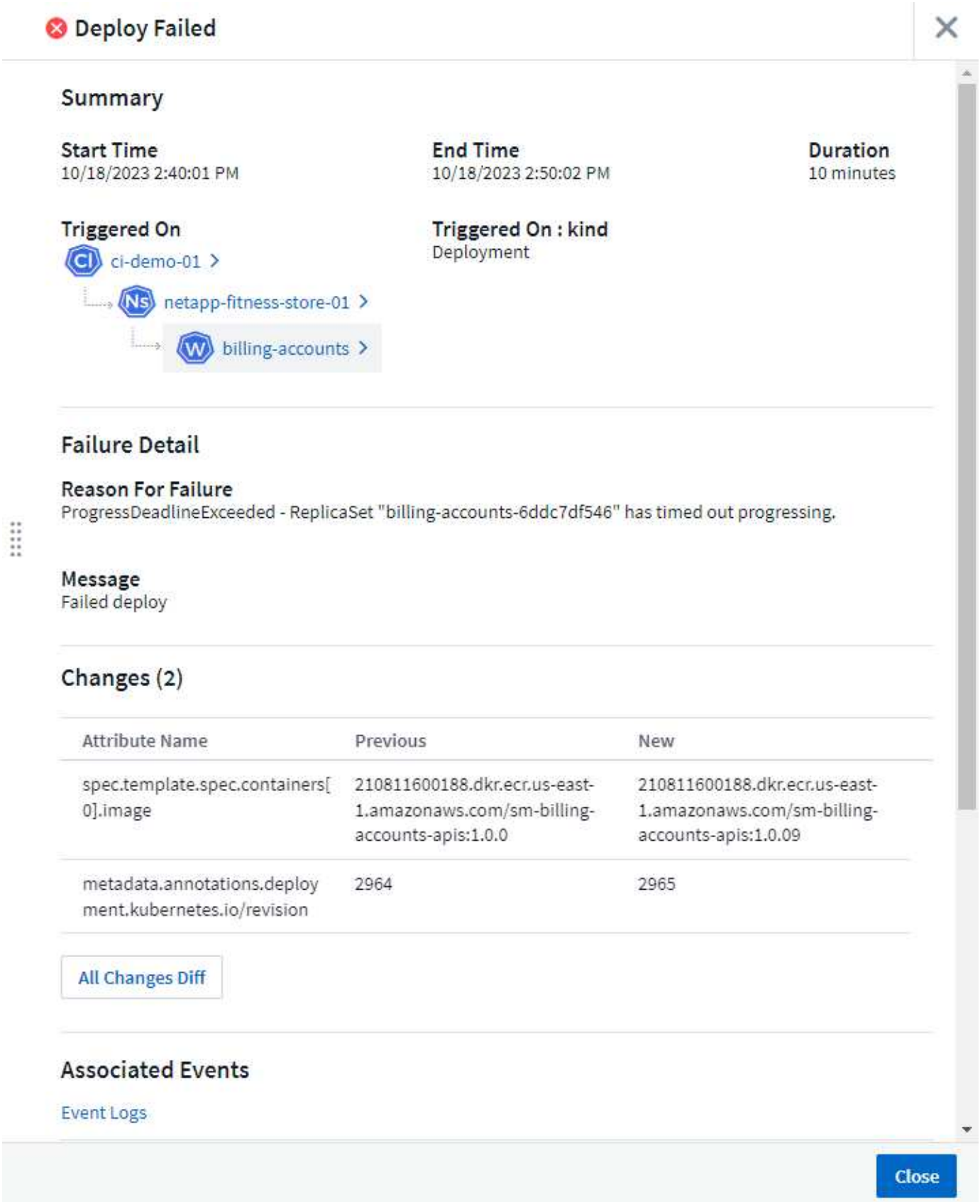
O status de implantação também é mostrado aqui. Novamente, o padrão é mostrar a contagem de implantações *Started*, *Complete* e *Failed*. Para ver apenas implementações *Failed*, defina um filtro para "Status" e selecione "Failed".

Os 3 principais workloads com mais alertas são os próximos. O número em vermelho ao lado de cada carga de trabalho indica o número de alertas relacionados a essa carga de trabalho. Clique no link carga de trabalho para explorar a infraestrutura (Kubernetes Explorer), dependências (mapa de carga de trabalho) ou análise de log (logs de eventos).



Painel de detalhes

Selecionar uma alteração na lista abre um painel descrevendo a alteração com mais detalhes. Por exemplo, selecionar uma implantação com falha mostra um resumo da implantação, com tempos de início e fim, duração e onde a implantação foi acionada, com links para explorar esses recursos. Ele também exibe o motivo da falha, quaisquer alterações relacionadas e quaisquer eventos associados.



A seleção de um Alerta fornece detalhes sobre o alerta, incluindo o monitor que acionou o alerta, bem como um gráfico que mostra uma linha do tempo visual para o alerta.

Informações sobre direitos autorais

Informações sobre marcas comerciais