Kubernetes

Data Infrastructure Insights

NetApp
February 10, 2026

Índice

Kubernetes

Visão geral do cluster Kubernetes

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

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

[Página de lista do Kubernetes]

Lista de clusters

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

- Cluster **Nome**. Clicar no nome de um cluster abrirá o"**página de detalhes**" para esse cluster.
- Porcentagens de **saturação**. Saturação geral é a maior saturação de CPU, memória ou armazenamento.
- Número de **Nós** no cluster. Clicar neste número abrirá a página Lista de nós.
- Número de **Pods** no cluster. Clicar neste número abrirá a página da lista de Pods.
- Número de **Namespaces** no cluster. Clicar neste número abrirá a página da lista de namespaces.
- Número de **Cargas de trabalho** no cluster. Clicar nesse número abrirá a página Lista de carga de trabalho.

Refinando o Filtro

Ao filtrar, ao começar a digitar, você terá a opção de criar um **filtro curinga** com base no texto atual. Selecionar esta opção retornará todos os resultados que correspondem à expressão curinga. Você também pode criar **expressões** usando NOT ou AND, ou pode selecionar a opção "Nenhum" para filtrar valores nulos no campo.

[Filtragem com curinga no K8S Explorer]

Filtros baseados em curingas ou expressões (por exemplo, NOT, AND, "None", etc.) são exibidos em azul escuro no campo de filtro. Os itens selecionados diretamente da lista são exibidos em azul claro.

[Filtro mostrando itens curinga e selecionados]

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 os pods naquele nó *e* naquele namespace.

Observe que a filtragem por curinga e expressão funciona com texto ou listas, mas não com números, datas ou booleanos.

Antes de instalar ou atualizar o NetApp Kubernetes Monitoring Operator

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

Componente	Exigência
Versão do Kubernetes	Kubernetes v1.20 e superior.
Distribuições Kubernetes	AWS Elastic Kubernetes Service (EKS) Azure Kubernetes Service (AKS) Google Kubernetes Engine (GKE) Red Hat OpenShift Rancher Kubernetes Engine (RKE) VMware Tanzu
Sistema operacional Linux	O Data Infrastructure Insights não oferece suporte a nós executados com arquitetura Arm64. Monitoramento de rede: deve estar executando o kernel Linux versão 4.18.0 ou superior. O Photon OS não é suportado.
Etiquetas	O Data Infrastructure Insights oferece suporte ao monitoramento de nós do Kubernetes que executam Linux, especificando um seletor de nós do Kubernetes que procura os seguintes rótulos do Kubernetes nessas plataformas: Kubernetes v1.20 e superior: Kubernetes.io/os = linux Rancher + cattle.io como plataforma de orquestração/Kubernetes: cattle.io/os = linux
Comandos	Os comandos curl e kubectl devem estar disponíveis; para melhores resultados, adicione esses comandos ao PATH.
Conectividade	O kubectl cli está configurado para se comunicar com o cluster K8s de destino e ter conectividade com a Internet para seu ambiente do Data Infrastructure Insights . Se você estiver atrás de um proxy durante a instalação, siga as instruções no"Configurando o suporte a proxy" seção da instalação do Operador. Para auditoria e relatórios de dados precisos, sincronize o horário na máquina do agente usando o Network Time Protocol (NTP) ou o Simple Network Time Protocol (SNTP).
Outro	Se você estiver executando o 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, atualizando-o ou substituindo-o), não há necessidade de criar um novo token de API; você pode reutilizar o token anterior.

Coisas importantes a serem observadas antes de começar

Se você estiver correndo com umprocuração , tem umrepositório personalizado , ou estão usandoOpenShift , leia atentamente as seções a seguir.

Leia também sobrePermissões .

Configurando o suporte a proxy

Há dois lugares onde você pode usar um proxy no seu locatário para instalar o NetApp Kubernetes Monitoring Operator. Esses podem ser os mesmos sistemas proxy ou sistemas separados:

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

Se você usar um proxy para um ou ambos, para instalar o NetApp Kubernetes Operating Monitor, primeiro você deve garantir que seu proxy esteja configurado para permitir uma boa comunicação com seu ambiente do Data Infrastructure Insights . Por exemplo, a partir dos servidores/VMs dos quais você deseja instalar o Operator, você precisa conseguir acessar o Data Infrastructure Insights e conseguir baixar binários do Data Infrastructure Insights.

Para o proxy usado para instalar o NetApp Kubernetes Operating Monitor, antes de instalar o Operator, defina as variáveis de ambiente *http_proxy/https_proxy*. Para alguns ambientes de proxy, talvez você também precise definir a variável de ambiente *no_proxy*.

Para definir a(s) variável(is), execute as seguintes etapas no seu sistema **antes** de instalar o NetApp Kubernetes Monitoring Operator:

1. Defina as variáveis 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_port>
```

Para que o proxy usado no seu cluster Kubernetes se comunique com seu ambiente do Data Infrastructure Insights , instale o NetApp Kubernetes Monitoring Operator depois de ler todas estas instruções.

Configure a seção proxy do AgentConfiguration em operator-config.yaml antes de implantar o NetApp Kubernetes Monitoring Operator.

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

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

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

Usando um repositório docker personalizado ou privado

Por padrão, o NetApp Kubernetes Monitoring Operator extrairá imagens de contêiner do repositório do Data Infrastructure Insights . Se você tiver um cluster Kubernetes usado como destino para monitoramento e esse cluster estiver configurado para extrair apenas imagens de contêiner de um repositório Docker personalizado ou privado ou de um registro de contêiner, você deverá configurar o acesso aos contêineres necessários para o NetApp Kubernetes Monitoring Operator.

Execute o “Image Pull Snippet” do bloco de instalação do NetApp Monitoring Operator. Este comando fará login no repositório do Data Infrastructure Insights , extrairá todas as dependências de imagem do operador e sairá do repositório do Data Infrastructure Insights . Quando solicitado, digite a senha temporária do repositório fornecida. Este comando baixa todas as imagens usadas pelo operador, inclusive para recursos opcionais. Veja abaixo para quais recursos essas imagens são usadas.

Funcionalidade do Operador Principal e Monitoramento do Kubernetes

- monitoramento netapp
- kube-rbac-proxy
- métricas de estado do kube
- telégrafo
- distroless-usuário-root

Registro de eventos

- fluente-bit
- exportador de eventos do kubernetes

Desempenho e Mapa da Rede

- observador ci-net

Envie a imagem do Docker do operador para 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 no seu repositório sejam consistentes com aqueles no repositório do Data Infrastructure Insights .

Edite a implantação do operador de monitoramento em operator-deployment.yaml e modifique todas as referências de imagem para usar seu repositório privado do Docker.

```
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 em operator-config.yaml para refletir o novo local do repositório do Docker. Crie um novo imagePullSecret para seu repositório privado. Para 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 o OpenShift 4.6 ou superior, deverá editar o AgentConfiguration em *operator-config.yaml* para habilitar 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 Logs de Eventos.

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

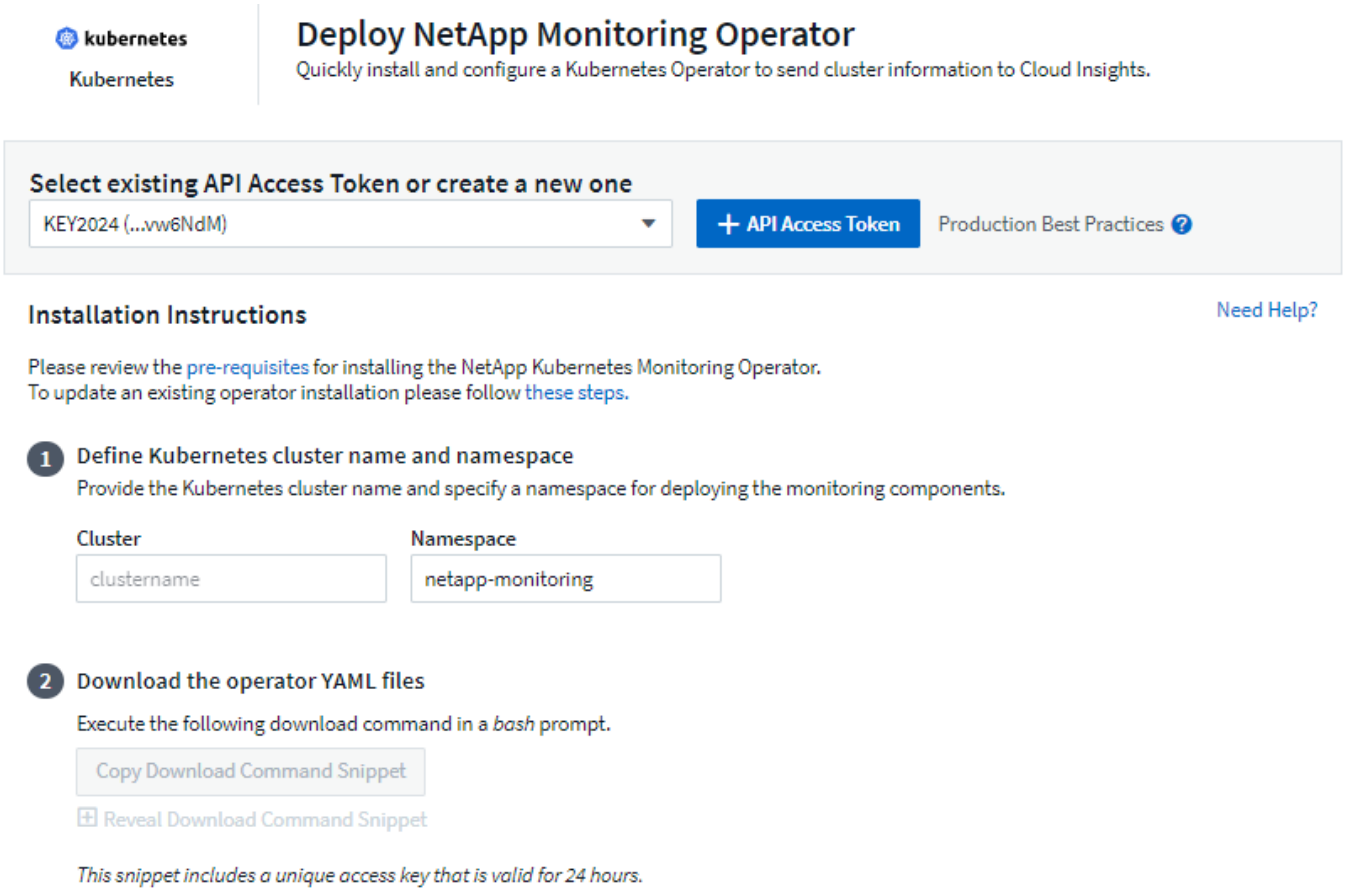
Instalação e configuração do operador de monitoramento do Kubernetes

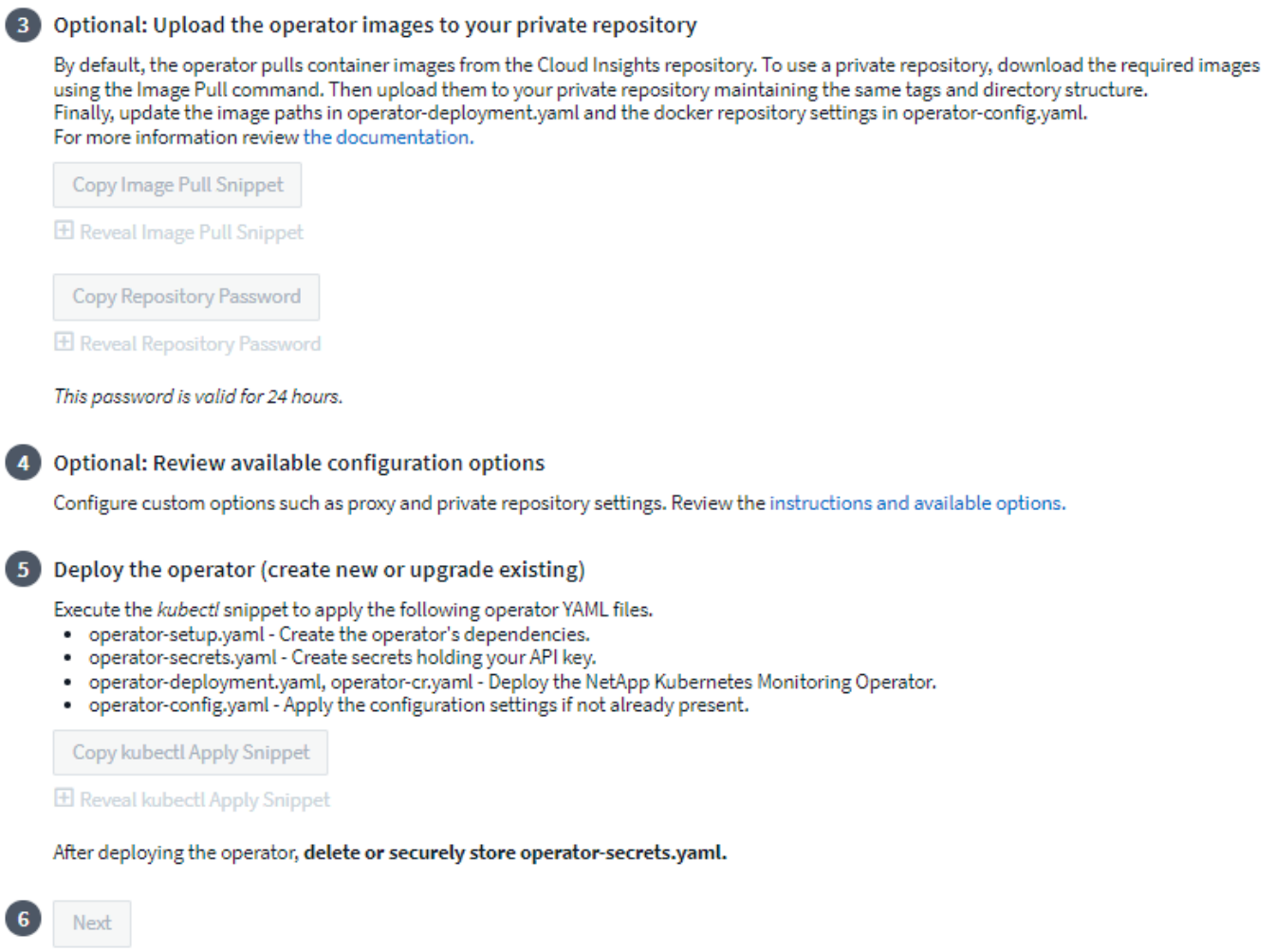
O Data Infrastructure Insights oferece o **Kubernetes Monitoring Operator** para a coleção Kubernetes. Navegue até **Kubernetes > Coletores > +Kubernetes Collector** para implantar um novo operador.

Antes de instalar o Kubernetes Monitoring Operator

Veja o"Pré-requisitos" documentação antes de instalar ou atualizar o Kubernetes Monitoring Operator.

Instalando o Operador de Monitoramento do Kubernetes





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 éatualizando de um operador Kubernetes anterior, use o mesmo nome de cluster e namespace.
2. Depois de inseri-los, você pode copiar o trecho do Comando de Download para a área de transferência.
3. Cole o snippet em uma janela *bash* e execute-o. Os arquivos de instalação do Operador serão baixados. Observe que o snippet tem uma chave única e é válido por 24 horas.
4. Se você tiver um repositório personalizado ou privado, copie o snippet opcional Image Pull, cole-o em um shell *bash* e execute-o. Depois que as imagens forem extraídas, copie-as para 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 do Docker em *operator-config.yaml*.
5. Se desejar, revise as opções de configuração disponíveis, como configurações de proxy ou repositório privado. Você pode ler mais sobre"opções de configuração" .
6. Quando estiver pronto, implante o Operador copiando o snippet kubectl Apply, baixando-o e executando-o.
7. A instalação prossegue automaticamente. Quando estiver concluído, clique no botão *Avançar*.
8. Quando a instalação estiver concluída, clique no botão *Avançar*. Certifique-se também de excluir ou armazenar com segurança o arquivo *operator-secrets.yaml*.

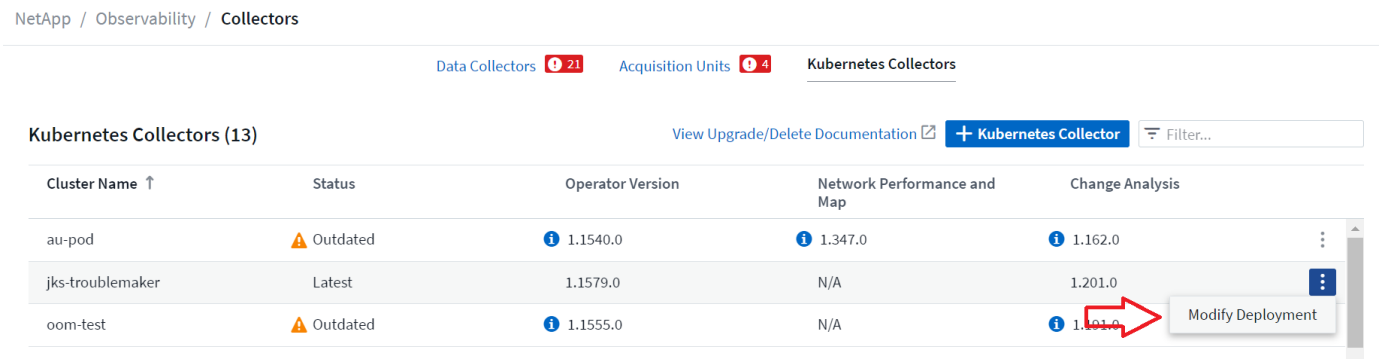
Se você tiver um repositório personalizado, leia sobreusando um repositório docker personalizado/privado .

Componentes de monitoramento do Kubernetes

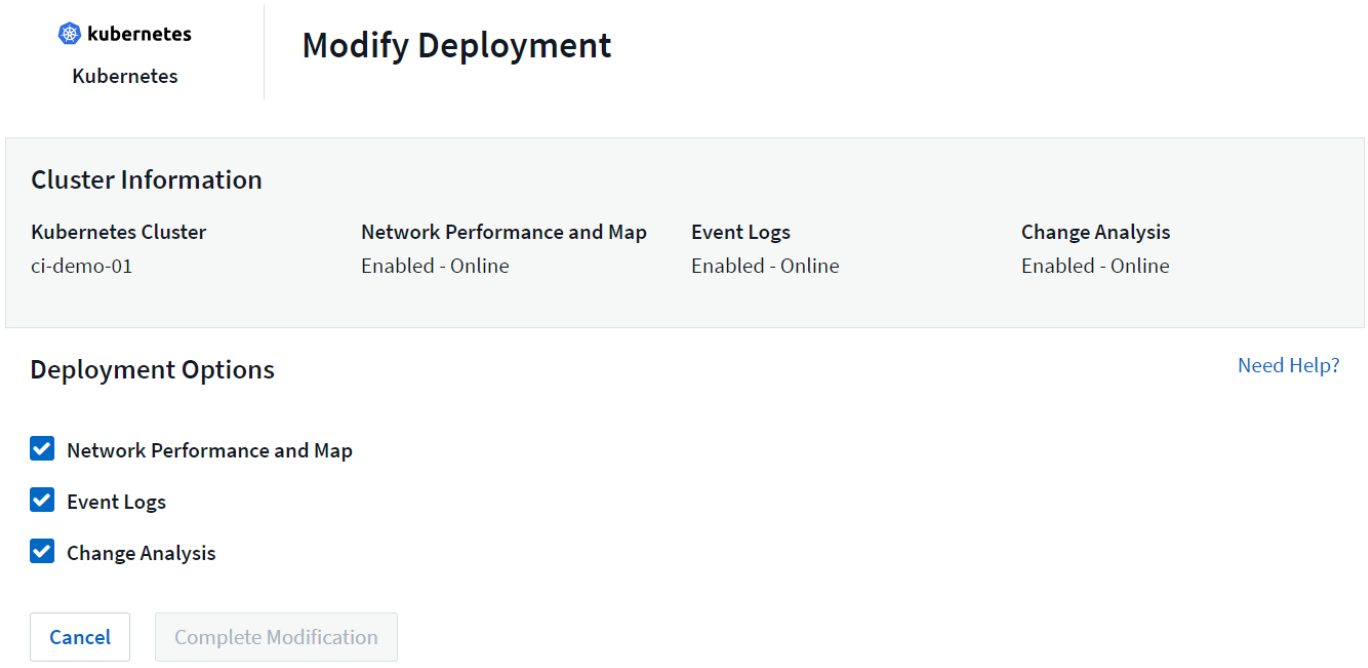
O Data Infrastructure Insights Kubernetes Monitoring é composto por quatro componentes de monitoramento:

- Métricas de Cluster
- Desempenho e mapa de rede (opcional)
- Registros de eventos (opcional)
- Análise de Mudanças (opcional)

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



A tela mostra o estado atual de cada componente e permite que você desabilite ou habilite componentes para aquele coletor, conforme necessário.



Atualizando para o mais recente Kubernetes Monitoring Operator

Atualizações de botão DII

Você pode atualizar o Kubernetes Monitoring Operator por meio da página DII Kubernetes Collectors. Clique no menu ao lado do cluster que você gostaria de atualizar e selecione *Atualizar*. O operador verificará as assinaturas de imagem, fará um snapshot da sua instalação atual e realizará a atualização. Em poucos minutos, você verá o progresso do status do operador, passando de Atualização em andamento para a mais recente. Se você encontrar um erro, pode selecionar o status Erro para obter mais detalhes e consultar a tabela de solução de problemas de atualizações por botão abaixo.

Atualizações por botão com repositórios privados

Se o seu operador estiver configurado para usar um repositório privado, certifique-se de que todas as imagens necessárias para executar o operador e suas assinaturas estejam disponíveis no seu repositório. Se você encontrar um erro durante o processo de atualização de imagens ausentes, basta adicioná-las ao seu repositório e tentar atualizar novamente. Para carregar as assinaturas de imagem no seu repositório, use a ferramenta cosign da seguinte forma, certificando-se de carregar as assinaturas de todas as imagens especificadas em 3 Opcional: Carregue as imagens do operador no seu repositório privado > Image Pull Snippet

```
cosign copy example.com/src:v1 example.com/dest:v1
#Example
cosign copy <DII container registry>/netapp-monitoring:<image version>
<private repository>/netapp-monitoring:<image version>
```

Revertendo para uma versão anterior

Se você atualizou usando o recurso de atualizações por botão e encontrar alguma dificuldade com a versão atual do operador dentro de sete dias após a atualização, você pode fazer o downgrade para a versão anterior usando o snapshot criado durante o processo de atualização. Clique no menu ao lado do cluster que você gostaria de reverter e selecione *Reverter*.

Atualizações manuais

Determine se existe uma *AgentConfiguration* com o operador existente (se o seu namespace não for o padrão *netapp-monitoring*, substitua pelo namespace apropriado):

```
 kubectl -n netapp-monitoring get agentconfiguration netapp-ci-monitoring-
configuration
Se uma _AgentConfiguration_ existir:
```

- Instalaro operador mais recente sobre o operador existente.
  - Certifique-se de que você estápuxando as últimas imagens de contêiner se você estiver usando um repositório personalizado.

Se a *AgentConfiguration* não existir:

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

```
 kubectl -n netapp-monitoring get agent -o
jsonpath='{.items[0].spec.cluster-name}'
* Crie um backup do Operador existente (se o seu namespace não for o
netapp-monitoring padrão, substitua pelo 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,Instalar>>o mais
recente Operador.
```

- Use o mesmo nome de cluster.
- Após baixar os arquivos YAML mais recentes do Operator, transfira quaisquer personalizações encontradas em *agent_backup.yaml* para o *operator-config.yaml* baixado antes de implantar.
- Certifique-se de que você estápuxando as últimas imagens de contêiner se você estiver usando um repositório personalizado.

Parando e iniciando o operador de monitoramento do Kubernetes

Para interromper 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
```

Desinstalando

Para remover o Operador de Monitoramento do Kubernetes

Observe que o namespace padrão para o Kubernetes Monitoring Operator é "netapp-monitoring". Se você tiver definido seu próprio namespace, substitua-o nestes e em todos os comandos e arquivos subsequentes.

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>
Observação: 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 de "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 Kube-state-metrics

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

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

Configurando/Personalizando o Operador

Estas seções contêm informações sobre como personalizar a configuração do seu 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 este recurso antes de implantar o operador editando o arquivo *operator-config.yaml*. Este arquivo inclui exemplos comentados de configurações. Veja a lista de"configurações disponíveis" para a versão mais recente do operador.

Você também pode editar esse recurso depois que o operador for 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 (Não encontrado)”, seu operador
deverá ser atualizado antes que você possa usar o AgentConfiguration.
```

Configurando o suporte a proxy

Há dois lugares onde você pode usar um proxy no seu locatário para instalar o Kubernetes Monitoring Operator. Esses podem ser os mesmos sistemas proxy ou sistemas separados:

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

Se você usar um proxy para um ou ambos, para instalar o Kubernetes Operating Monitor, primeiro você deve garantir que seu proxy esteja configurado para permitir uma boa comunicação com seu ambiente do Data Infrastructure Insights . Se você tiver um proxy e puder acessar o Data Infrastructure Insights do servidor/VM do qual deseja instalar o Operator, é provável que seu proxy esteja configurado corretamente.

Para o proxy usado para instalar o Kubernetes Operating Monitor, antes de instalar o Operator, defina as variáveis de ambiente *http_proxy/https_proxy*. Para alguns ambientes de proxy, talvez você também precise definir a variável de ambiente *no_proxy*.

Para definir a(s) variável(is), execute as seguintes etapas no seu sistema **antes** de instalar o Kubernetes Monitoring Operator:

1. Defina as variáveis 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 no seu cluster Kubernetes se comunique com seu ambiente do Data Infrastructure Insights , instale o Kubernetes Monitoring Operator depois de ler todas estas instruções.

Configure a seção proxy de *AgentConfiguration* em *operator-config.yaml* antes de implantar o Kubernetes Monitoring Operator.

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

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

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

Usando um repositório docker personalizado ou privado

Por padrão, o Kubernetes Monitoring Operator extrairá imagens de contêiner do repositório do Data Infrastructure Insights . Se você tiver um cluster Kubernetes usado como destino para monitoramento e esse cluster estiver configurado para extrair apenas imagens de contêiner de um repositório Docker personalizado ou privado ou de um registro de contêiner, você deverá configurar o acesso aos contêineres necessários para o Kubernetes Monitoring Operator.

Execute o “Image Pull Snippet” do bloco de instalação do NetApp Monitoring Operator. Este comando fará login no repositório do Data Infrastructure Insights , extrairá todas as dependências de imagem do operador e sairá do repositório do Data Infrastructure Insights . Quando solicitado, digite a senha temporária do repositório fornecida. Este comando baixa todas as imagens usadas pelo operador, inclusive para recursos opcionais. Veja abaixo para quais recursos essas imagens são usadas.

Funcionalidade do Operador Principal e Monitoramento do Kubernetes

- monitoramento netapp
- ci-kube-rbac-proxy
- ci-ksm
- ci-telegraf
- distroless-usuário-root

Registro de eventos

- ci-fluent-bit
- ci-kubernetes-event-exporter

Desempenho e Mapa da Rede

- observador ci-net

Envie a imagem do Docker do operador para 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 no seu repositório sejam consistentes com aqueles no repositório do Data Infrastructure Insights .

Edite a implantação do operador de monitoramento em operator-deployment.yaml e modifique todas as referências de imagem para usar seu repositório privado do Docker.

```
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* em *operator-config.yaml* para refletir a nova localização do repositório docker. Crie um novo imagePullSecret para seu repositório privado, para mais detalhes veja *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
```

Token de acesso à API para senhas de longo prazo

Alguns ambientes (ou seja, repositórios proxy) exigem senhas de longa duração para o repositório Data Infrastructure Insights docker. A senha fornecida na interface do usuário no momento da instalação é válida por apenas 24 horas. Em vez de usar essa senha, pode-se usar um API Access Token como senha do repositório docker. Essa senha será válida enquanto o API Access Token for válido. Pode-se gerar um novo API Access Token para esse propósito específico ou usar um já existente.

"Leia aqui" para obter instruções sobre como criar um novo token de acesso à API.

Para extrair um token de acesso à API existente de um arquivo *operator-secrets.yaml* baixado, os usuários podem executar o seguinte:

```
grep '\.dockerconfigjson' operator-secrets.yaml |sed 's/.*\.dockerconfigjson:
//g' |base64 -d |jq
```

Para extrair um token de acesso à API existente de uma instalação de operador em execução, os usuários podem executar o seguinte:

```
kubectl -n netapp-monitoring get secret netapp-ci-docker -o
jsonpath='{.data.\.dockerconfigjson}' |base64 -d |jq
```

Instruções do OpenShift

Se você estiver executando o OpenShift 4.6 ou superior, você deve editar o *AgentConfiguration* em *operator-config.yaml* para habilitar 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.

Tolerâncias e Manchas

Os DaemonSets *netapp-ci-telegraf-ds*, *netapp-ci-fluent-bit-ds* e *netapp-ci-net-observer-l4-ds* devem agendar um pod em cada nó do cluster para coletar dados corretamente em todos os nós. O operador foi configurado para tolerar algumas **manchas** bem conhecidas. Se você configurou alguma contaminação personalizada em seus nós, impedindo assim que os pods sejam executados em todos os nós, você pode criar uma **tolerância** para essas contaminações"na *Configuração do Agente*" . Se você tiver aplicado taints personalizados a todos os nós do cluster, também deverá adicionar as tolerâncias necessárias à implantação do operador para permitir que o pod do operador seja agendado e executado.

Saiba mais sobre o Kubernetes"Manchas e Tolerâncias" .

Voltar para o"* Página de instalação do operador de monitoramento do NetApp Kubernetes*"

Uma nota sobre segredos

Para remover a permissão do Kubernetes Monitoring Operator para visualizar segredos em todo o cluster, exclua os seguintes recursos do arquivo *operator-setup.yaml* antes da instalação:

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

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

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

Se a Análise de Mudanças estiver habilitada, modifique *AgentConfiguration* ou *operator-config.yaml* para descomentar a seção de gerenciamento de mudanças e incluir *kindsToIgnoreFromWatch: '"secrets"'* na seção de gerenciamento de mudanças. Observe a presença e a posição das 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 do operador e todas as imagens relacionadas que ele implementa são assinadas pela NetApp. Você pode verificar manualmente as imagens antes da instalação usando a ferramenta cosign ou configurar um controlador de admissão do Kubernetes. Para mais detalhes, consulte o"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: Carregar as imagens do operador para 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 Image Pull Snippet
2. Copie e insira a senha do repositório quando solicitado
3. Armazene a chave pública da assinatura da imagem (dii-image-signing.pub no exemplo)
4. Verifique as imagens usando cosign. Consulte o seguinte exemplo de uso de cosigno

```
$ 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 que você pode tentar se tiver problemas ao configurar o Kubernetes Monitoring Operator:

Problema:	Experimente isto:
Não vejo um hiperlink/conexão entre meu Volume Persistente do Kubernetes e o dispositivo de armazenamento de back-end correspondente. Meu volume persistente do Kubernetes é configurado usando o nome do host do servidor de armazenamento.	Siga as etapas para desinstalar o agente Telegraf existente e reinstale o agente Telegraf mais recente. Você deve estar usando o Telegraf versão 2.0 ou posterior, e seu armazenamento de cluster Kubernetes deve ser monitorado ativamente pelo Data Infrastructure Insights.
Estou vendo mensagens nos logs semelhantes às seguintes: E0901 15:21:39.962145 1 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 15:21:43.168161 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: Falha ao listar *v1.Lease: o servidor não conseguiu encontrar o recurso solicitado (obter leases.coordination.k8s.io) etc.	Essas mensagens podem ocorrer se você estiver executando o kube-state-metrics versão 2.0.0 ou superior com versões do Kubernetes inferiores à 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='{..image}'* Para evitar que essas mensagens aconteçam, os usuários podem modificar sua implantação do kube-state-metrics para desabilitar os seguintes Leases: *mutatingwebhookconfigurations validatingwebhookconfigurations volumeattachments resources* Mais especificamente, eles podem usar o seguinte argumento da CLI: resources=certificatesigningrequests,configmaps,cronjobs,daemonsets, deployments,endpoints,horizontalpodautoscalers,ingresses,jobs,limitranges, namespaces,networkpolicies,nodes,persistentvolumeclaims,persistentvolumes, poddisruptionbudgets,pods,replicasets,replicationcontrollers,resourcequotas, segredos, serviços, conjuntos de estado, classes de armazenamento A lista de recursos padrão é: "certificatesigningrequests, configmaps, cronjobs, daemonsets, implantações, endpoints, horizontalpodautoscalers, ingresses, jobs, leases, limitranges, mutatingwebhookconfigurations, namespaces, networkpolicies, nodes, persistentvolumeclaims, persistentvolumes, poddisruptionbudgets, pods, replicasets, replicationcontrollers, resourcequotas, secrets, services, statefulsets, storageclasses, validatingwebhookconfigurations, volumeattachments"

Problema:	Experimente isto:
Vejo mensagens de erro do Telegraf semelhantes às seguintes, mas o Telegraf inicia e executa: 11 de outubro 14:23:41 ip-172-31-39-47 systemd[1]: Iniciado O agente do servidor controlado por plugin para relatar métricas no InfluxDB. 11 de out. 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="falha ao criar diretório de cache. /etc/telegraf/.cache/snowflake, err: mkdir /etc/telegraf/.ca che: permissão negada. ignorada\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" 11 de out. 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="falha ao abrir. Ignorado. abra /etc/telegraf/.cache/snowflake/ocsp_response_cache.json: arquivo ou diretório inexistente\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" 11 de out. 14:23:41 ip-172-31-39-47 telegraf[1827]: 2021-10-11T14:23:41Z Eu! Iniciando o Telegraf 1.19.3	Este é um problema conhecido. Consulte"Este artigo do GitHub" para mais detalhes. Enquanto o Telegraf estiver funcionando, os usuários podem ignorar essas mensagens de erro.
No Kubernetes, meus pods Telegraf estão relatando o seguinte erro: "Erro no processamento de informações de mountstats: falha ao abrir o arquivo mountstats: /hostfs/proc/1/mountstats, erro: abrir /hostfs/proc/1/mountstats: permissão negada"	Se o SELinux estiver habilitado e em execução, é provável que ele esteja impedindo que o(s) pod(s) Telegraf acessem o arquivo /proc/1/mountstats no nó do Kubernetes. Para superar essa restrição, edite a configuração do agente e ative a configuração runPrivileged. Para mais detalhes, consulte as instruções do OpenShift.
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/kubernetes/pki/etcd/server.crt:/etc/kubernetes/pki/etcd/server.key: aberto /etc/kubernetes/pki/etcd/server.crt: nenhum arquivo ou diretório desse tipo	O pod Telegraf ReplicaSet foi projetado para ser executado em um nó designado como mestre ou para etcd. Se o pod ReplicaSet não estiver em execução em um desses nós, você receberá esses erros. Verifique se seus nós mestre/etcd têm contaminações. Se isso acontecer, adicione as tolerâncias necessárias ao Telegraf ReplicaSet, telegraf-rs. Por exemplo, edite o ReplicaSet… kubectl edit rs telegraf-rs …e adicione as tolerâncias apropriadas à especificação. Em seguida, reinicie o pod ReplicaSet.

Problema:	Experimente isto:
Tenho um ambiente PSP/PSA. Isso afeta meu operador de monitoramento?	Se o seu cluster Kubernetes estiver em execução com a Política de Segurança de Pod (PSP) ou a Admissão de Segurança de Pod (PSA) em vigor, você deverá atualizar para a versão mais recente do Operador de Monitoramento do Kubernetes. Siga estas etapas para atualizar para a Operadora atual com suporte para PSP/PSA: 1. Desinstalar o operador de monitoramento anterior: kubectl delete agent 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-metrics-reader kubectl delete clusterrolebinding agent-manager-rolebinding agent-proxy-rolebinding agent-cluster-admin-rolebinding 2. Instalar a versão mais recente do operador de monitoramento.
Tive problemas ao tentar implantar o Operador e tenho o PSP/PSA em uso.	1. Edite o agente usando o seguinte comando: kubectl -n <name-space> edit agent 2. Marque 'security-policy-enabled' como 'false'. Isso desabilitará as Políticas de Segurança do Pod e a Admissão de Segurança do Pod e permitirá que o Operador faça a implantação. Confirme usando os seguintes comandos: kubectl get psp (deve mostrar que a Política de Segurança do Pod foi removida) kubectl get all -n <namespace>
grep -i psp (deve mostrar que nada foi encontrado)	Erros "ImagePullBackoff" vistos
Esses erros podem ser vistos se você tiver um repositório docker personalizado ou privado e ainda não tiver configurado o Kubernetes Monitoring Operator para reconhecê-lo corretamente. Ler mais sobre 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 seguintes comandos 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 (Mapa de Carga de Trabalho) no namespace Operator estão em CrashLoopBackOff

Problema:	Experimente isto:
Esses pods correspondem ao coletor de dados do Workload Map para Network Observability. Tente isto: • 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","environment":"prod","level":"error","msg":"falha na validação. Motivo: a versão do kernel 3.10.0 é inferior à versão mínima do kernel 4.18.0","time":"2022-11-09T08:23:08Z"} ---- • Os pods do Net-observer exigem que a versão do kernel Linux seja pelo menos 4.18.0. Verifique a versão do kernel usando o comando “uname -r” e certifique-se de que seja >= 4.18.0	Os pods estão sendo executados no namespace do Operador (padrão: netapp-monitoring), mas nenhum dado é mostrado na IU para o mapa de carga de trabalho ou métricas do Kubernetes em Consultas
Verifique a configuração de tempo nos nós do cluster K8S. Para auditoria e relatórios de dados precisos, 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 do net-observer no namespace do 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 de CPU ou memória. Certifique-se de que a memória e a CPU necessárias estejam disponíveis no nó.	Estou vendo o seguinte em meus logs imediatamente após instalar o Kubernetes Monitoring Operator: [inputs.prometheus] Erro no plugin: erro ao fazer solicitação HTTP para http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: Obter http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: discar tcp: procurar kube-state-metrics.<namespace>.svc.cluster.local: nenhum host
Essa mensagem normalmente só é vista quando um novo operador é instalado e o pod *telegraf-rs* é ativado antes do pod *ksm*. 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 sua versão do Kubernetes (ou seja, `kubectl version`). Se for v1.20.x ou anterior, esta é uma limitação esperada. A versão kube-state-metrics implantada com o Kubernetes Monitoring Operator suporta apenas a v1.CronJob. Com o Kubernetes 1.20.x e versões anteriores, o recurso CronJob está em v1beta.CronJob. Como resultado, o kube-state-metrics não consegue encontrar o recurso CronJob.	Após instalar o operador, os pods telegraf-ds entram em CrashLoopBackOff e os logs dos pods indicam "su: Falha de autenticação".
Edite a seção telegraf em *AgentConfiguration* e defina *dockerMetricCollectionEnabled* como false. Para mais detalhes, consulte o "opções de configuração" do operador. … spec: … telegraf: … - name: docker run-mode: - DaemonSet substitutions: - key: DOCKER_UNIX_SOCK_PLACEHOLDER value: unix:///run/docker.sock … …	Vejo mensagens de erro repetidas 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 ao aguardar cabeçalhos)

Problema:	Experimente isto:
Edite a seção telegraf em *AgentConfiguration* e aumente *outputTimeout* para 10s. Para mais detalhes, consulte o manual do operador"opções de configuração" .	Estou sem dados *involvedobject* para alguns Logs de Eventos.
Certifique-se de ter seguido os passos no"Permissões" seção 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>?
A partir de 12 de outubro de 2023, o Data Infrastructure Insights refatorou o operador para melhor atender nossos usuários; para que essas mudanças sejam totalmente adotadas, você deveremover o operador antigo einstalar o novo .	Meus eventos do Kubernetes pararam inesperadamente de reportar ao Data Infrastructure Insights.
Recupere o nome do pod do 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 do exportador de eventos apropriado encontrado na etapa anterior. Mais especificamente, LOG_FILE deve ser definido como "/var/log/containers/netapp-ci-event-exporter.log" ou "/var/log/containers/event-exporter*.log" …. fluent-bit: … - name: event-exporter-ci substitutions: - key: LOG_FILE values: - /var/log/containers/netapp-ci-event-exporter*.log … …. Alternativamente, também se podedesinstalar ereinstalar o agente.
Estou vendo pod(s) implantado(s) pelo Kubernetes Monitoring Operator travando devido a recursos insuficientes.	Consulte o Operador de Monitoramento do Kubernetes"opções de configuração" para aumentar os limites da CPU e/ou memória conforme necessário.

Problema:	Experimente isto:
Uma imagem ausente ou configuração inválida fez com que os pods netapp-ci-kube-state-metrics falhassem na inicialização ou ficassem prontos. Agora o StatefulSet está travado 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, faça o retorno dos pods netapp-ci-kube-state-metrics.
Os pods netapp-ci-kube-state-metrics falham ao iniciar após executar uma atualização do Kubernetes Operator, gerando ErrImagePull (falha ao extrair a imagem).	Tente redefinir os pods manualmente.
Mensagens "Evento descartado por ser mais antigo que maxEventAgeSeconds" estão sendo observadas no meu cluster Kubernetes na Análise de Log.	Modifique o operador *agentconfiguration* e aumente *event-exporter-maxEventAgeSeconds* (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 o"opções de configuração" página.
O 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 *unprotected* como *true*. Isso instruirá o Telegraf a não tentar reservar páginas de memória bloqueadas. Embora isso possa representar um risco à segurança, pois segredos descriptografados podem ser transferidos para o disco, isso permite a execução em ambientes onde não é possível reservar memória bloqueada. Para mais detalhes sobre as opções de configuração *desprotegidas*, consulte o"opções de configuração" página.
Vejo mensagens de aviso do Telegraf parecidas com as seguintes: *W! [inputs.diskio] Não foi possível coletar o nome do disco para "vdc": erro ao ler /dev/vdc: arquivo ou diretório inexistente*	Para o operador de monitoramento do Kubernetes, essas mensagens de aviso são benignas e podem ser ignoradas com segurança. Como alternativa, edite a seção telegraf em AgentConfiguration e defina *runDsPrivileged* como true. Para mais detalhes, consulte a "opções de configuração do operador".

Problema:	Experimente isto:

Meu pod fluent-bit está falhando com os seguintes erros: [2024/10/16 14:16:23] [erro] [/src/fluent-bit/plugins/in_tail/tail_fs_inotify.c:360 errno=24] Muitos arquivos abertos [2024/10/16 14:16:23] [erro] falha ao inicializar a entrada tail.0 [2024/10/16 14:16:23] [erro] falha na inicialização da entrada [engine]	Tente alterar as configurações do *fsnotify* no seu 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 nas reinicializações dos nós, 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>`

Problema:	Experimente isto:
Os pods do Telegraf DS estão relatando erros referentes ao plug-in de entrada do Kubernetes que não consegue fazer solicitações HTTP devido à incapacidade de validar o certificado TLS. Por exemplo: E! [inputs.kubernetes] Erro no plugin: erro ao fazer solicitação HTTP para"<a href="https://<kubelet_IP>:10250/stats/summary": " class="bare">https://<kubelet_IP>:10250/stats/summary": Pegar"<a href="https://<kubelet_IP>:10250/stats/summary": " class="bare">https://<kubelet_IP>:10250/stats/summary": tls: falha ao verificar o certificado: x509: não é possível validar o certificado para <kubelet_IP> porque ele não contém nenhum SAN IP	Isso ocorrerá se o kubelet estiver usando certificados autoassinados e/ou o certificado especificado não incluir o <kubelet_IP> na lista *Nome alternativo do assunto* dos certificados. Para resolver isso, o usuário pode modificar o"configuração do agente" , e defina *telegraf:insecureK8sSkipVerify* como *true*. Isso configurará o plugin de entrada do Telegraf para pular a verificação. Alternativamente, o usuário pode configurar o kubelet para"servidorTLSBootstrap" , que acionará uma solicitação de certificado da API 'certificates.k8s.io'.
Estou recebendo o seguinte erro nos pods do Fluent-bit e o pod não inicia: 026/01/12 20:20:32] [erro] [sqldb] erro=não foi possível abrir o arquivo de banco de dados [2026/01/12 20:20:32] [erro] [input:tail:tail.0] db: não foi possível criar a tabela 'in_tail_files' [2026/01/12 20:20:32] [erro] [input:tail:tail.0] não foi possível abrir/criar o banco de dados [2026/01/12 20:20:32] [erro] falha ao inicializar a entrada tail.0 [2026/01/12 20:20:32] [erro] [engine] falha na inicialização da entrada	Certifique-se de que o diretório do host em que o arquivo DB reside tenha as permissões de leitura/gravação adequadas. Mais especificamente, o diretório do host deve conceder permissões de leitura/gravação a usuários que não sejam root. O local padrão do arquivo DB é /var/log/, a menos que seja substituído pela opção fluent-bit-dbFile *agentconfiguration*. Se o SELinux estiver habilitado, tente definir a opção fluent-bit-seLinuxOptionsType agentconfiguration como 'spc_t'.

Informações adicionais podem ser encontradas em"Apoiar" página ou no"Matriz de Suporte ao Coletor de Dados" .

Opções de configuração do operador de monitoramento do Kubernetes

O "Operador de monitoramento do Kubernetes" Oferece amplas opções de personalização por meio do arquivo AgentConfiguration. Você pode configurar limites de recursos, intervalos de coleta, configurações de proxy, tolerâncias e configurações específicas de componentes para otimizar o monitoramento do seu ambiente Kubernetes. Use essas opções para personalizar o telegraf, o kube-state-metrics, a coleta de logs, o mapeamento de carga de trabalho, o gerenciamento de alterações e outros componentes de monitoramento.

Arquivo de configuração do agente de amostra

Abaixo segue um exemplo de arquivo *AgentConfiguration*, com descrições para cada opção.

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

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

spec:
  ##
  ## One can modify the following settings to configure and customize the
operator.
  ## Optional settings are commented out with their 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 cluster name
    ## The cluster name must be unique across all clusters in your Data
Infrastructure Insights (DII) environment.
    ##
    clusterName: "my_cluster"

    ##
    ## Proxy settings
    ## If applicable, specify the proxy through which the operator should
communicate with DII.
    ## Refer to additional 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]
    ## Repository from which the operator pulls the required images
    ## By default, the operator pulls from the DII repository. To use a
```

```
private repository, set this field to the
    ## applicable repository name. Refer to additional 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]
    ## Name of the imagePullSecret required for dockerRepo
    ## When using a private repository, set this field to the applicable
secret name.
    ##
    dockerImagePullSecret: 'netapp-ci-docker'

    ##
    ## Automatic expiring API key rotation settings
    ## Allow the operator to automatically rotate its expiring API key,
generating a new API key and
    ## using it to replace the expiring one. The expiring API key itself
must support auto rotation.
    ##
    # tokenRotationEnabled: 'true'
    ##
    ## Threshold (number of days before expiration) at which the operator
should trigger rotation.
    ## The threshold must be less than the total duration of the API key.
    ##
    # tokenRotationThresholdDays: '30'

  push-button-upgrades:
    ##
    ## Allow the operator to be upgraded using the Data Infrastructure
Insights (DII) UI
    ##
    # enabled: 'true'

    ##
    ## Frequency at which the operator polls and checks for upgrade
requests from DII
    ##
    # polltimeSeconds: '60'

    ##
    ## Allow operator upgrade to proceed even if new images are not
present
```

```
    ##
    # ignoreImageNotPresent: 'false'

    ##
    ## Allow operator upgrade to proceed even if image signature
verification fails
    ## Warning: Enabling this setting is dangerous!
    ##
    # ignoreImageSignatureFailure: 'false'

    ##
    ## Allow operator upgrade to proceed even if image signature
verification fails
    ## Warning: Enabling this setting is dangerous!
    ##
    # ignoreYAMLSignatureFailure: 'false'

    ##
    ## Use dockerImagePullSecret to access the image repository and verify
the existence of the new images
    ##
    # imageValidationUseSecret: 'true'

    ##
    ## Time allowed for the old operator pod to shutdown before reporting
an upgrade failure to DII
    ##
    # upgradesShutdownTime: '240'

    ##
    ## Time allowed for the new operator pod to startup before reporting
an upgrade failure to DII
    ##
    # upgradesStartupTime: '600'

  telegraf:
    ##
    ## Frequency at which telegraf collects data
    ## The frequency should not exceed 60s.
    ##
    # collectionInterval: '60s'

    ##
    ## Maximum number of metrics per batch
    ## Telegraf sends metrics to outputs in batches. This controls the
size of those writes.
```

```
    ##
    # batchSize: '10000'

    ##
    ## Maximum number of unwritten metrics per output
    ## Telegraf caches metrics until they are successfully written by the
output. This controls how many metrics
    ## can be cached. Once the buffer is filled, the oldest metrics will
get dropped.
    ##
    # bufferLimit: '150000'

    ##
    ## Rounds collection interval to collectionInterval
    ## If collectionInterval is 60s, collection will occur on-the-minute
    ##
    # roundInterval: 'true'

    ##
    ## Jitter between plugins on collection
    ## Each input plugin sleeps a random amount of time within jitter
before collecting. This can be used to prevent
    ## multiple input plugins from querying the same resources at the same
time. The maximum collection interval would
    ## be collectionInterval + collectionJitter.
    ##
    # collectionJitter: '0s'

    ##
    ## Precision to which collected metrics are rounded
    ## When set to "0s", precision will be set by the units specified by
collectionInterval.
    ##
    # precision: '0s'

    ##
    ## Frequency at which telegraf flushes and writes data
    ## Frequency should not exceed collectionInterval.
    ##
    # flushInterval: '60s'

    ##
    ## Jitter between plugins on writes
    ## Each output plugin sleeps a random amount of time within jitter
before flushing. This can be used to prevent
    ## multiple output plugins from writing the same resources at the same
```

```
time, and causing large spikes. The maximum
    ## flush interval would be flushInterval + flushJitter.
    ##
    # flushJitter: '0s'

    ##
    ## Timeout for HTTP output plugins
    ## Time allowed for http output plugins to successfully writing before
failing.
    ##
    # outputTimeout: '5s'

    ##
    ## CPU/Mem limits and requests for netapp-ci-telegraf-ds DaemonSet
    ##
    # dsCpuLimit: '750m'
    # dsMemLimit: '800Mi'
    # dsCpuRequest: '100m'
    # dsMemRequest: '500Mi'

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

    ##
    ## telegraf runs through the processor plugins a second time after the
aggregators plugins, by default. Use this
    ## option to skip the second run.
    ##
    # skipProcessorsAfterAggregators: 'false'

    ##
    ## Additional tolerations for netapp-ci-telegraf-ds DaemonSet and
netapp-ci-telegraf-rs ReplicaSet
    ## Inspect the netapp-ci-telegraf-rs ReplicaSet and netapp-ci-
telegraf-ds DaemonSet to view the default tolerations.
    ## If additional tolerations are needed, specify them here using the
following abbreviated single line format:
    ##
    ## Example: '{key: taint1, operator: Exists, effect: NoSchedule},{key:
taint2, operator: Exists, effect: NoExecute}'
    ##
```

```
    # dsTolerations: ''
    # rsTolerations: ''

    ##
    ## Additional node selector terms for netapp-ci-telegraf-rs ReplicaSet
    ## Inspect the netapp-ci-telegraf-rs ReplicaSet to view the default
node selectors terms. If additional node
    ## selector terms are needed, specify them here using the following
abbreviated single line format:
    ##
    ## Example: '{"key": "myLabel1","operator": "In","values":
["myVal1"]},{"key": "myLabel2","operator": "In","values": ["myVal2"]}'
    ##
    ## These additional node selector terms will be AND'd with the default
ones via matchExpressions.
    ##
    # rsNodeSelectorTerms: ''

    ##
    ## telegraf uses lockable memory to protect secrets in memory. If
telegraf issues warnings about insufficient
    ## lockable memory, try increasing the limit of lockable memory on the
applicable nodes. If increasing this limit
    ## is not an option for the given environment, set unprotected to true
so telegraf does not attempt to use
    ## lockable memory.
    ##
    # unprotected: 'false'

    ##
    ## Run the netapp-ci-telegraf-ds DaemonSet's telegraf-mountstats-
poller container in privileged mode
    ## The telegraf-mountstats-poller container needs read-only access to
system files such as those in /proc/ (i.e. to
    ## monitor NFS IO metrics, etc.). Some environments impose restricts
that prevent the container from reading these
    ## system files. Unless those restrictions are lifted, users may need
to run this container in privileged mode.
    ##
    # runPrivileged: 'false'

    ##
    ## Run the netapp-ci-telegraf-ds DaemonSet's telegraf container in
privileged mode
    ## The telegraf container needs read-only access to system files such
as those in /dev/ (i.e. for the telegraf
```

```
    ## diskio input plugin to retrieve disk metrics). Some environments
impose restricts that prevent the container from
    ## accessing these system files. Unless those restrictions are lifted,
users may need to run this container in
    ## privileged mode.
    ##
    # runDsPrivileged: 'false'

    ##
    ## Allow the netapp-ci-telegraf-ds DaemonSet's telegraf-ds, telegraf-
init, and telegraf-mountstats-poller containers
    ## to run with escalation privilege. This is needed to access/read
root-protected files (node UUID,
    ## /proc/1/mountstats, etc.). Allowing escalation privilege should
negate the need to run these containers in
    ## privileged mode.
    ##
    # allowDsPrivilegeEscalation: 'true'

    ##
    ## Allow the netapp-ci-telegraf-rs DaemonSet's telegraf-rs and
telegraf-rs-init containers
    ## to run with escalation privilege. This is needed to access/read
root-protected files (node UUID,
    ## etcd credentials when applicable, etc.). Allowing escalation
privilege should negate the need to run these
    ## containers in privileged mode.
    ##
    # allowRsPrivilegeEscalation: 'true'

    ##
    ## Enable collection of block IO metrics (kubernetes.pod_to_storage)
    ##
    # dsBlockIOEnabled: 'true'

    ##
    ## Enable collection of NFS IO metrics (kubernetes.pod_to_storage)
    ##
    # dsNfsIOEnabled: 'true'

    ##
    ## Enable collection of system-specific objects/metrics for managed
k8s clusters
    ## This consists of k8s objects within the kube-system and cattle-
system namespaces for managed k8s clusters
    ## (i.e. EKS, AKS, GKE, managed Rancher, etc.).
```

```
    ##
    # managedK8sSystemMetricCollectionEnabled: 'false'

    ##
    ## Enable collection of pod ephemeral storage metrics
(kubernetes.pod_volume)
    ##
    # podVolumeMetricCollectionEnabled: 'false'

    ##
    ## Declare Rancher cluster is managed
    ## Rancher can be deployed in managed or on-premise environments. The
operator contains logic to try to determine
    ## which type of environment Rancher is running in (i.e. to factor
into managedK8sSystemMetricCollectionEnabled).
    ## If the operator logic misidentifies whether Rancher is running in a
managed environment or not, use this option
    ## to declare Rancher is managed.
    ##
    # isManagedRancher: 'false'

    ##
    ## Locations for the etcd certificate and key files
    ## The operator looks at well-known locations for the etcd certificate
and key files. If this cannot find these
    ## files, the applicable telegraf input plugin will fail. Use this
option to specify the complete filepath to these
    ## files on the nodes.
    ## Note that the well-known locations for these files are typically
root-protected. This is one of the reasons why
    ## the netapp-ci-telegraf-rs ReplicaSet's telegraf-rs-init container
needs to run with escalation privileges.
    ##
    # rsHostEtcdCrt: ''
    # rsHostEtcdKey: ''

    ##
    ## Allow operator/telegraf communications with k8s without TLS
verification
    ## In some environments, TLS verification will not succeed (i.e.
certificates lack IP SANs). To skip the
    ## verification, use this option.
    ##
    # insecureK8sSkipVerify: 'false'

  kube-state-metrics:
```

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

    ##
    ## Comma-separated list of k8s resources for which to collect metrics
    ## Refer to the kube-state-metrics --resources CLI option
    ##
    # resources:
'cronjobs,daemonsets,deployments,horizontalpodautoscalers,ingresses,jobs,n
amespaces,nodes,persistentvolumeclaims,persistentvolumes,pods,replicasets,
resourcequotas,services,statefulsets'

    ##
    ## Comma-separated list of k8s metrics to collect
    ## Refer to the kube-state-metrics --metric-allowlist CLI option
    ##
    # 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,kube_horizontal
podautoscaler_spec_max_replicas,kube_horizontalpodautoscaler_spec_min_repl
icas,kube_horizontalpodautoscaler_status_condition,kube_horizontalpodautos
caler_status_current_replicas,kube_horizontalpodautoscaler_status_desired_
replicas'

    ##
```

```
    ## Comma-separated list of k8s label keys that will be used to
determine which labels to export/collect
    ## Refer to the kube-state-metrics --metric-labels-allowlist CLI
option
    ##
    # labels:
'cronjobs=[*],daemonsets=[*],deployments=[*],horizontalpodautoscalers=[*],
ingresses=[*],jobs=[*],namespaces=[*],nodes=[*],persistentvolumeclaims=[*]
,persistentvolumes=[*],pods=[*],replicasets=[*],resourcequotas=[*],service
s=[*],statefulsets=[*]'

    ##
    ## Additional tolerations for netapp-ci-kube-state-metrics StatefulSet
    ## Inspect the netapp-ci-kube-state-metrics StatefulSet to view the
default tolerations. If additional
    ## tolerations are needed, specify them here using the following
abbreviated single line format:
    ##
    ## Example: '{key: taint1, operator: Exists, effect: NoSchedule},{key:
taint2, operator: Exists, effect: NoExecute}'
    ##
    # tolerations: ''

    ##
    ## Additional node selector terms for netapp-ci-kube-state-metrics
StatefulSet
    ## Inspect the kube-state-metrics StatefulSet to view the default node
selectors terms. If additional node selector
    ## terms are needed, specify them here using the following abbreviated
single line format:
    ##
    ## Example: '{"key": "myLabel1","operator": "In","values":
["myVal1"]},{"key": "myLabel2","operator": "In","values": ["myVal2"]}'
    ##
    ## These additional node selector terms will be AND'd with the default
ones via matchExpressions.
    ##
    # nodeSelectorTerms: ''

    ##
    ## Number of kube-state-metrics shards
    ## For large clusters, kube-state-metrics may be overwhelmed with
collecting and exporting the amount of metrics
    ## generated. This can lead to collection timeouts for the netapp-ci-
telegraf-rs pod. If this is observed, use this
    ## option to increase the number of kube-state-metrics shards to
```

```
redistribute the workload.
    ##
    # shards: '2'

  logs:
    ##
    ## Allow the netapp-ci-fluent-bit-ds DaemonSet's fluent-bit container
to run with escalation privilege.
    ## This is needed to access/read root-protected files (event-exporter
pod log, fluent-bit DB file, etc.).
    ##
    # fluent-bit-allowPrivilegeEscalation: 'true'

    ##
    ## Read content from the head of the file, not the tail
    ##
    # readFromHead: "true"

    ##
    ## Network protocol for DNS (i.e. UDP, TCP, etc.)
    ##
    # dnsMode: "UDP"

    ##
    ## DNS resolver (i.e. LEGACY, ASYNC, etc.)
    ##
    # fluentBitDNSResolver: "LEGACY"

    ##
    ## Additional tolerations for netapp-ci-fluent-bit-ds DaemonSet
    ## Inspect the netapp-ci-fluent-bit-ds DaemonSet to view the default
tolerations. If additional tolerations are
    ## needed, specify them here using the following abbreviated single
line format:
    ##
    ## Example: '{key: taint1, operator: Exists, effect: NoSchedule},{key:
taint2, operator: Exists, effect: NoExecute}'
    ##
    # fluent-bit-tolerations: ''

    ##
    ## CPU/Mem limits and requests for netapp-ci-fluent-bit-ds DaemonSet
    ##
    # fluent-bit-cpuLimit: '500m'
    # fluent-bit-memLimit: '1Gi'
    # fluent-bit-cpuRequest: '50m'
```

```
    # fluent-bit-memRequest: '100Mi'

    ##
    ## Top-level host path in which the kubernetes container logs reside,
including any symlinks from var/log/containers
    ## For example, if /var/log/containers/*.log is a symlink to
/kubernetes/log to
    ## /kubernetes/var/lib/docker/containers/*/*.log, fluent-bit-
containerLogPath should be set to '/kubernetes'.
    ##
    # fluent-bit-containerLogPath: '/var/lib/docker/containers'

    ## fluent-bit DB file path/location

    ##
    ## fluent-bit DB file path/location
    ## By default, fluent-bit is configured to use /var/log/netapp-
monitoring_flb_kube.db.  This path usually requires
    ## escalated privileges for read/write. Users who want to avoid
escalation privilege can use this option to specify
    ## a different DB file path/location. The custom path/location should
allow non-root users to read/write.
    ## Ideally, the path/location should be persistent.
    ##
    # fluent-bit-dbFile: '/var/log/netapp-monitoring_flb_kube.db'

    ##
    ## Additional tolerations for netapp-ci-event-exporter Deployment
    ## Inspect the netapp-ci-event-exporter Deployment to view the default
tolerations. If additional tolerations are
    ## needed, specify them here using the following abbreviated single
line format:
    ##
    ## Example: '{key: taint1, operator: Exists, effect: NoSchedule},{key:
taint2, operator: Exists, effect: NoExecute}'
    ##
    # event-exporter-tolerations: ''

    ##
    ## CPU/Mem limits and requests for netapp-ci-event-exporter Deployment
    ##
    # event-exporter-cpuLimit: '500m'
    # event-exporter-memLimit: '1Gi'
    # event-exporter-cpuRequest: '50m'
    # event-exporter-memRequest: '100Mi'
```

```
    ##
    ## Max age for events to be processed and exported; older events are
discarded
    ##
    # event-exporter-maxEventAgeSeconds: '10'

    ##
    ## Client-side throttling
    ## Set event-exporter-kubeBurst to roughly match event rate
    ## Set event-exporter-kubeQPS to approximately 1/5 of event-exporter-
kubeBurst
    ##
    # event-exporter-kubeQPS: 20
    # event-exporter-kubeBurst: 100

    ##
    ## Additional node selector terms for netapp-ci-event-exporter
Deployment
    ## Inspect the event-exporter Deployment to view the default node
selectors terms. If additional node selector terms
    ## are needed, specify them here using the following abbreviated
single line format:
    ##
    ## Example: '{"key": "myLabel1","operator": "In","values":
["myVal1"]},{"key": "myLabel2","operator": "In","values": ["myVal2"]}'
    ##
    ## These additional node selector terms will be AND'd with the default
ones via matchExpressions.
    ##
    # event-exporter-nodeSelectorTerms: ''

  workload-map:
    ## Run workload-map container with escalation privilege to coordinate
memlocks
    ##
    ## Allow the netapp-ci-net-observer-l4-ds DaemonSet's net-observer
container to run with escalation privilege.
    ## This is needed to coordinate memlocks.
    ##
    # allowPrivilegeEscalation: 'true'

    ##
    ## CPU/Mem limits and requests for netapp-ci-net-observer-l4-ds
DaemonSet
    ##
    # cpuLimit: '500m'
```

```
    # memLimit: '500Mi'
    # cpuRequest: '100m'
    # memRequest: '500Mi'

    ##
    ## Metric aggregation interval (in seconds)
    ## Set metricAggregationInterval between 30 and 120
    ##
    # metricAggregationInterval: '60'

    ##
    ## Interval for bpf polling
    ## Set bpfPollInterval between 3 and 15
    ##
    # bpfPollInterval: '8'

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

    ##
    ## Additional tolerations for netapp-ci-net-observer-l4-ds DaemonSet
    ## Inspect the netapp-ci-net-observer-l4-ds DaemonSet to view the
default tolerations. If additional tolerations
    ## are needed, specify them here using the following abbreviated
single line format:
    ##
    ## Example: '{key: taint1, operator: Exists, effect: NoSchedule},{key:
taint2, operator: Exists, effect: NoExecute}'
    ##
    # l4-tolerations: ''

    ##
    ## Run the netapp-ci-net-observer-l4-ds DaemonSet's net-observer
container in privileged mode
    ## Some environments impose restricts that prevent the net-observer
container from running.
    ## Unless those restrictions are lifted, users may need to run this
container in privileged mode.
    ##
    # runPrivileged: 'false'

  change-management:
    ##
    ## CPU/Mem limits and requests for netapp-ci-change-observer-watch-rs
```

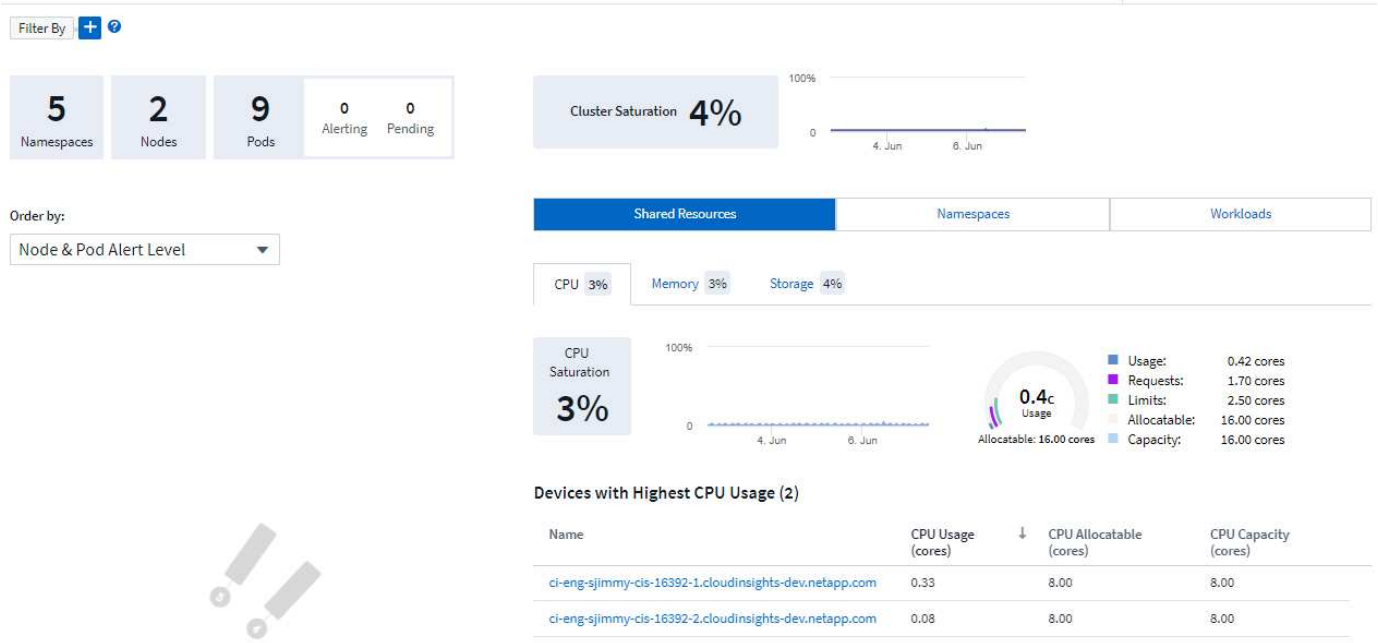
```
ReplicaSet
    ##
    # cpuLimit: '1'
    # memLimit: '1Gi'
    # cpuRequest: '500m'
    # memRequest: '500Mi'

    ##
    ## Interval (in seconds) after which a non-successful deployment of a
workload will be marked as failed
    ##
    # workloadFailureDeclarationIntervalSeconds: '30'

    ##
    ## Frequency (in seconds) at which workload deployments are combined
and sent
    ##
    # workloadDeployAggrIntervalSeconds: '300'

    ##
    ## Frequency (in seconds) at which non-workload deployments are
combined and sent
    ##
    # nonWorkloadDeployAggrIntervalSeconds: '15'

    ##
    ## 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"'

    ##
    ## Additional node selector terms for netapp-ci-change-observer-watch-
rs ReplicaSet
    ## Inspect the netapp-ci-change-observer-watch-rs ReplicaSet to view
the default node selectors terms. If additional
    ## node selector terms are needed, specify them here using the
following abbreviated single line format:
    ##
    ## Example: '{"key": "myLabel1","operator": "In","values":
["myVal1"]},{"key": "myLabel2","operator": "In","values": ["myVal2"]}'
    ##
    ## These additional node selector terms will be AND'd with the default
```

```
ones via matchExpressions.
    ##
    # nodeSelectorTerms: ''

    ##
    ## Comma-separated list of additional kinds to watch
    ## Each kind should be prefixed by its API group. This list in
addition to the default set of kinds watched by the
    ## collector.
    ##
    ## Example: '"authorization.k8s.io.subjectaccessreviews"'
    ##
    # additionalKindsToWatch: ''

    ##
    ## 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: ''

    ##
    ## Comma-separated list of kinds to ignore from watching from the
default set of kinds watched by the collector
    ## Each kind should be prefixed by its API group.
    ##
    ## Example: '"networking.k8s.io.networkpolicies,batch.jobs",
"authorization.k8s.io.subjectaccessreviews"'
    ##
    # kindsToIgnoreFromWatch: ''

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

    ##
    ## Additional tolerations for netapp-ci-change-observer-watch-rs
ReplicaSet
    ## Inspect the netapp-ci-change-observer-watch-rs ReplicaSet to view
the default tolerations. If additional
    ## tolerations are needed, specify them here using the following
abbreviated single line format:
```

```
    ##
    ## 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 Kubernetes exibe uma visão geral detalhada do seu cluster Kubernetes.



Contagens de namespace, nó e pod

As contagens no topo da página mostram o número total de namespaces, nós e pods no cluster, bem como o número de popds que estão atualmente em alerta e pendentes.

Recursos Compartilhados e Saturação

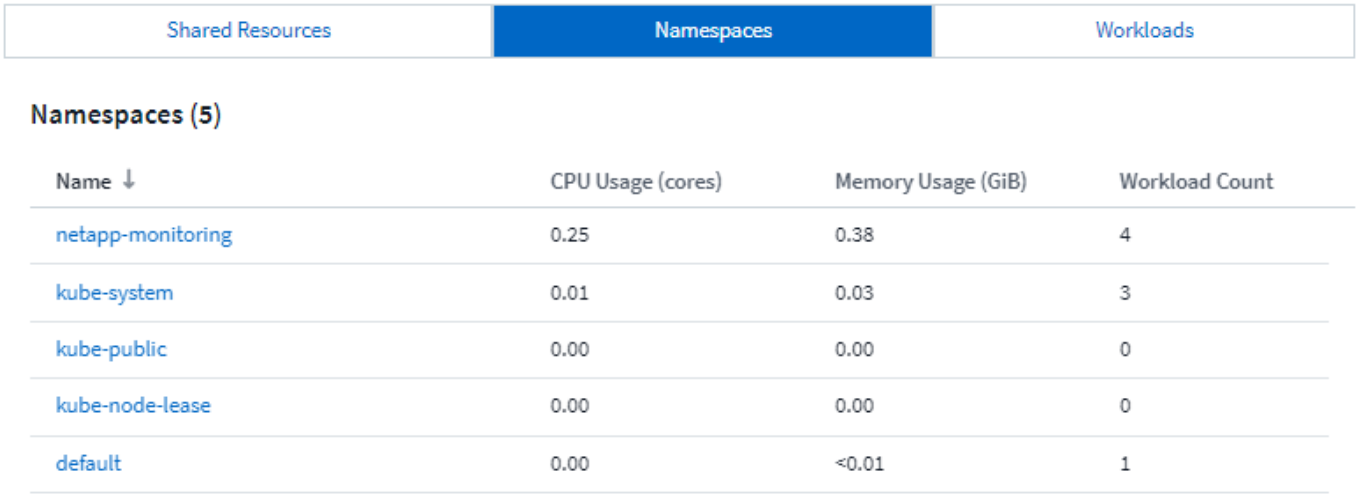
No canto superior direito da página de detalhes está a saturação do cluster como uma porcentagem atual, bem como um gráfico mostrando a tendência recente ao longo do tempo. A saturação do cluster é a maior saturação de CPU, memória ou armazenamento em cada ponto no tempo.

Abaixo disso, a página mostra por padrão o uso de **Recursos Compartilhados**, com guias para CPU, Memória e Armazenamento. Cada aba mostra a porcentagem de saturação e a tendência ao longo do tempo, com detalhes adicionais de uso. Para armazenamento, o valor mostrado é o maior entre a saturação do backend e do sistema de arquivos, que são calculados independentemente.

Os dispositivos com maior uso são mostrados em uma tabela na parte inferior. Clique em qualquer link para explorar esses dispositivos.

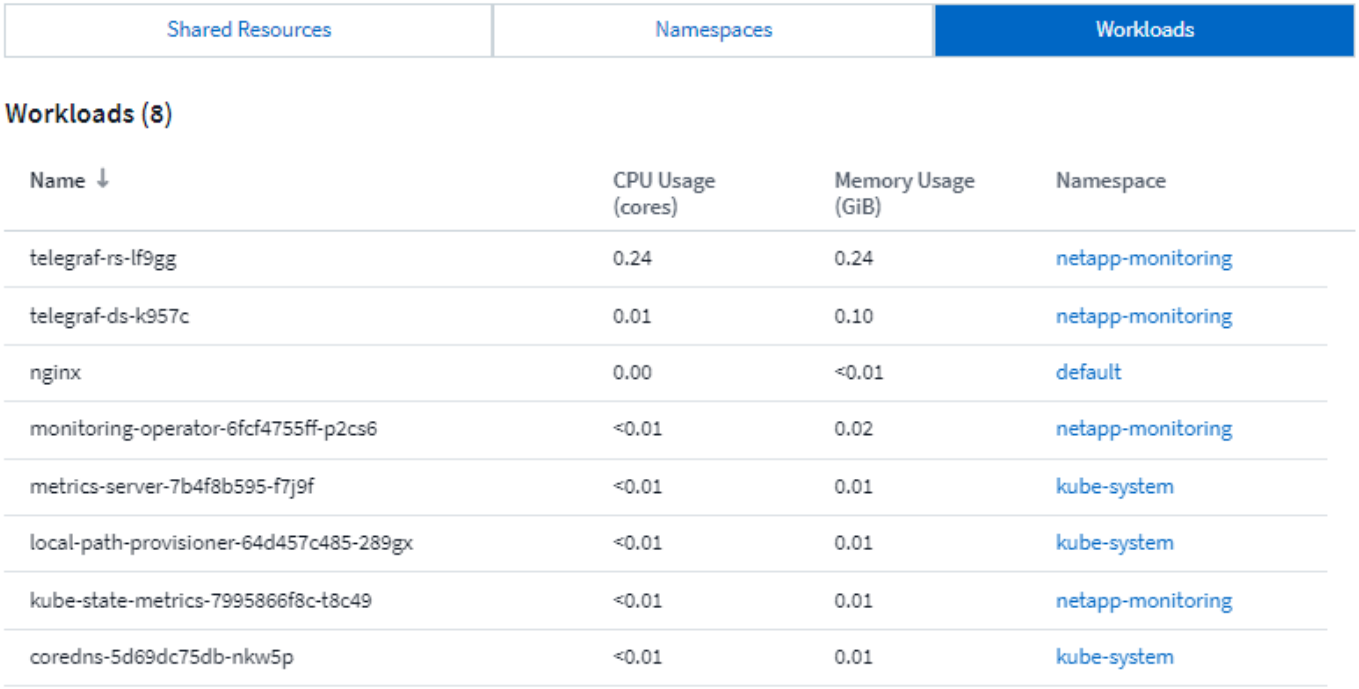
Espaços de nomes

A guia Namespaces exibe uma lista de todos os namespaces no 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.



Cargas de trabalho

Da mesma forma, a guia Cargas de trabalho exibe uma lista das cargas de trabalho em cada namespace, mostrando novamente o uso de CPU e memória. Clicar nos links do Namespace permite que você se aprofunde em cada um deles.



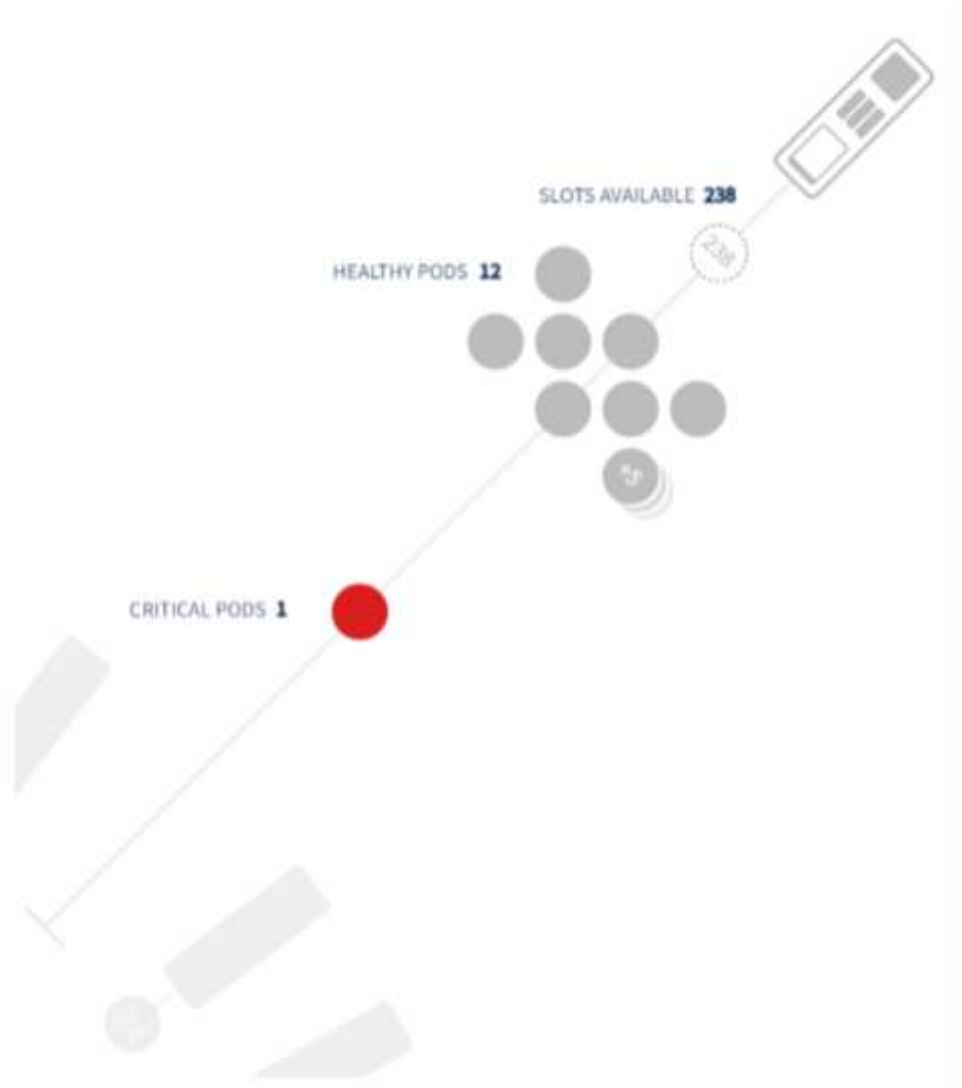
A "Roda" do Cluster

A seção "Roda" do Cluster fornece informações gerais sobre a integridade dos nós e pods, nas quais você pode se aprofundar para obter mais informações. Se o seu cluster contiver mais nós do que podem ser exibidos nesta área da página, você poderá girar a roda usando os botões disponíveis.

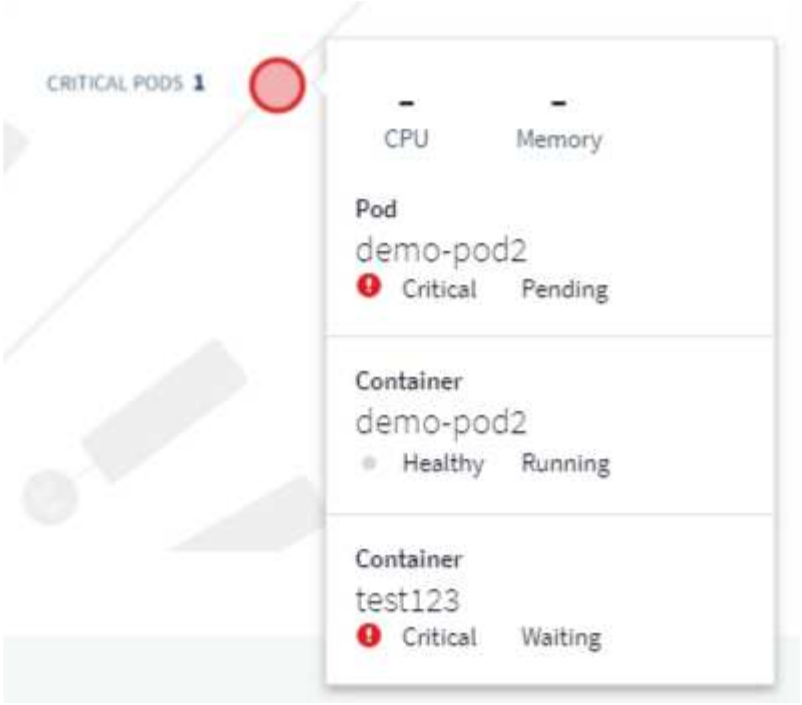
Os pods ou nós de alerta são exibidos em vermelho. As áreas de "aviso" são exibidas em laranja. Os pods não programados (ou seja, não anexados) serão exibidos no canto inferior da "Roda" do Cluster.

Passar o mouse sobre um pod (círculo) ou nó (barra) ampliará a visualização do nó.

Clicar no pod ou nó nessa visualização ampliará a visualização expandida do Nó.



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



Você pode visualizar informações do sistema de arquivos, memória e CPU passando o mouse sobre os elementos do nó.

Uma nota sobre os medidores

Os indicadores de memória e CPU mostram três cores, pois mostram *usado* em relação à *capacidade alocável* e à *capacidade total*.

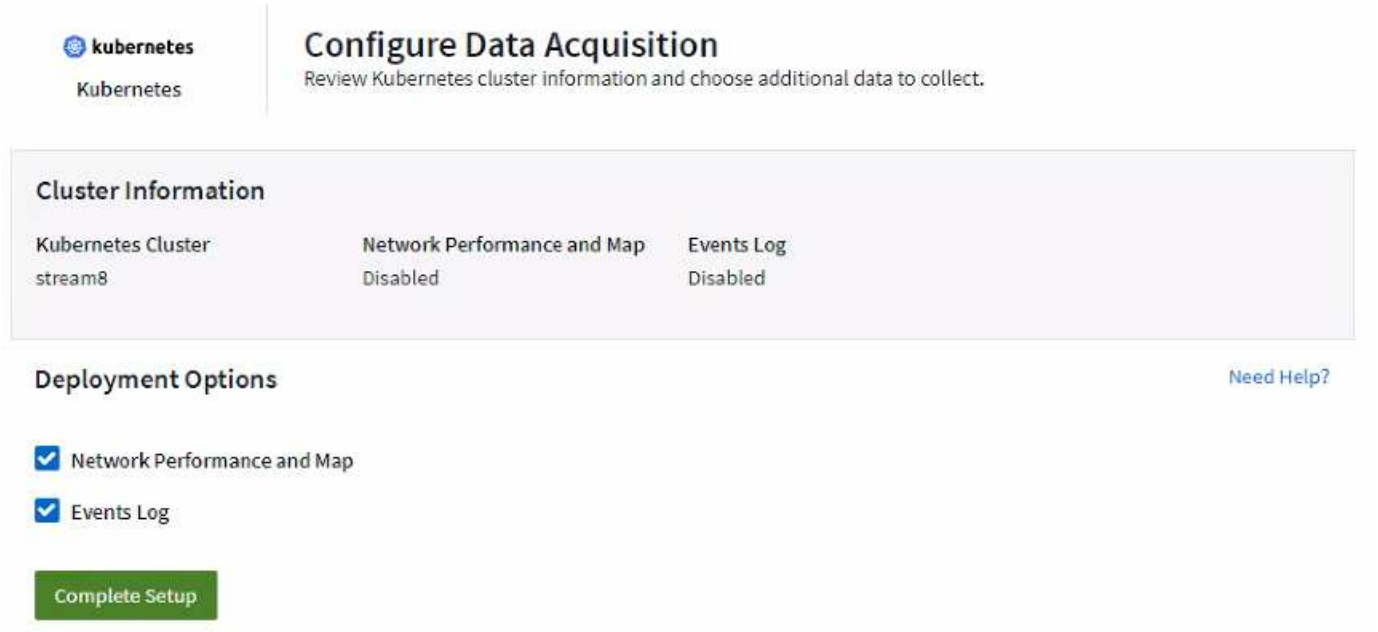
Monitoramento e mapeamento de desempenho da rede Kubernetes

O recurso Monitoramento e Mapeamento de Desempenho de Rede do Kubernetes simplifica a solução de problemas mapeando dependências entre serviços (também chamados de cargas de trabalho) e fornece visibilidade em tempo real das latências e anomalias de desempenho da rede para identificar problemas de desempenho antes que eles afetem os usuários. Esse recurso ajuda as organizações a reduzir 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 tráfego e latência. • Reduza os custos gerais analisando o tráfego de rede de entrada, saída, entre regiões e entre zonas.

Pré-requisitos

Antes de poder usar o Monitoramento e Mapa de Desempenho de Rede do Kubernetes, você deve ter configurado o"Operador de monitoramento do NetApp Kubernetes" para habilitar esta opção. Durante a implantação do Operador, marque a caixa de seleção "Desempenho e Mapa de Rede" para habilitar. Você também pode habilitar essa opção navegando até uma página de destino do Kubernetes e selecionando "Modificar implantação".



Monitores

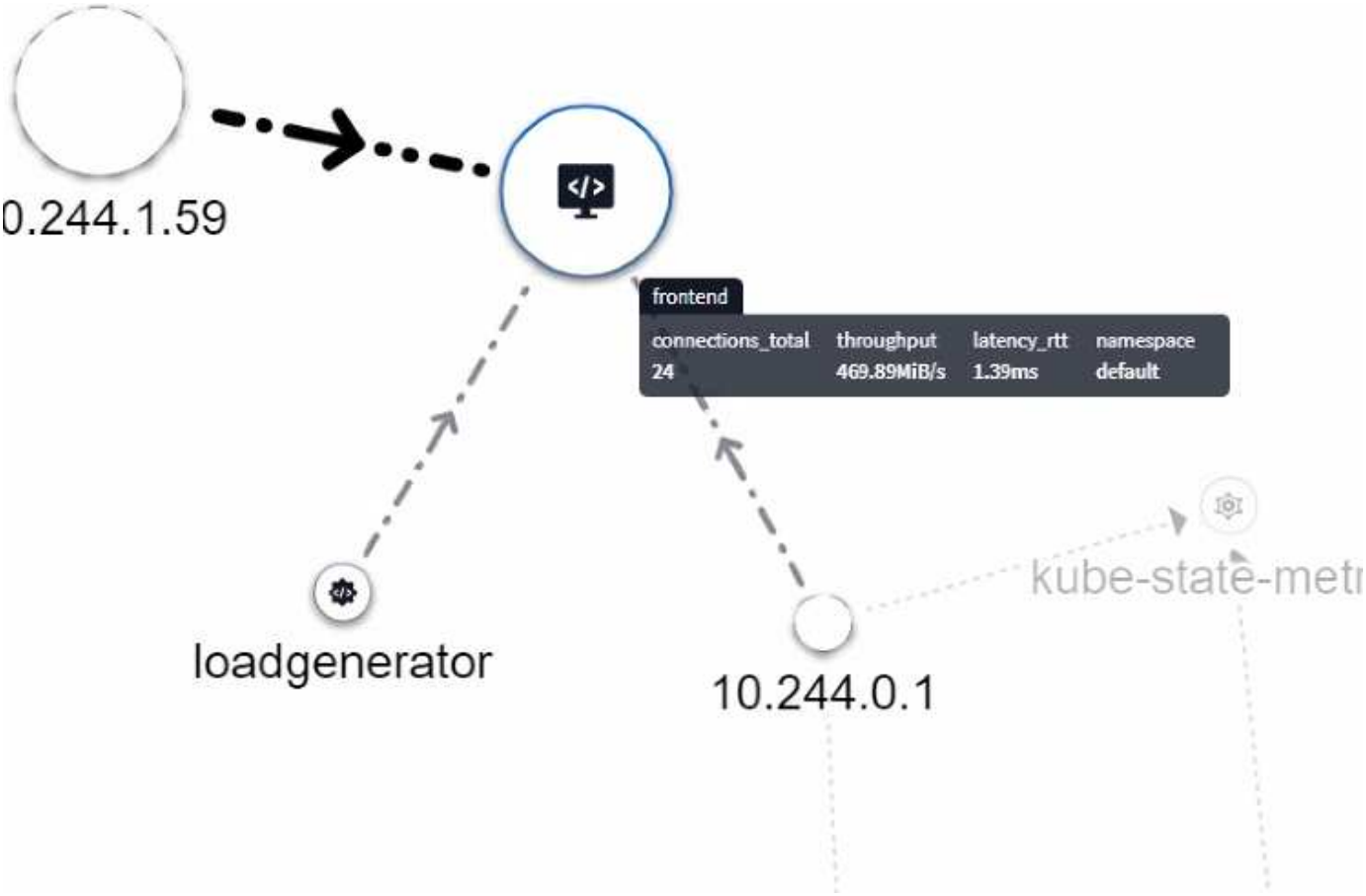
O Mapa de Carga de Trabalho usa"monitores" para derivar informações. O Data Infrastructure Insights fornece vários monitores Kubernetes padrão (observe que eles podem estar *Pausados* por padrão). Você pode *Retomar* (ou seja, habilitar) os monitores que desejar ou pode criar monitores personalizados para objetos do Kubernetes, 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 estejam agrupados pelo tipo de objeto padrão.

- kubernetes.carga de trabalho
- kubernetes.daemonset
- kubernetes.implantação
- kubernetes.cronjob
- kubernetes.trabalho
- kubernetes.replicaset
- kubernetes.conjunto de estados
- kubernetes.pod
- kubernetes.network_traffic_l4

O Mapa

O mapa mostra serviços/cargas de trabalho e seus relacionamentos entre si. As setas mostram as direções do trânsito. Passar o mouse sobre uma carga de trabalho exibe informações resumidas sobre essa carga de trabalho, como você pode ver neste exemplo:



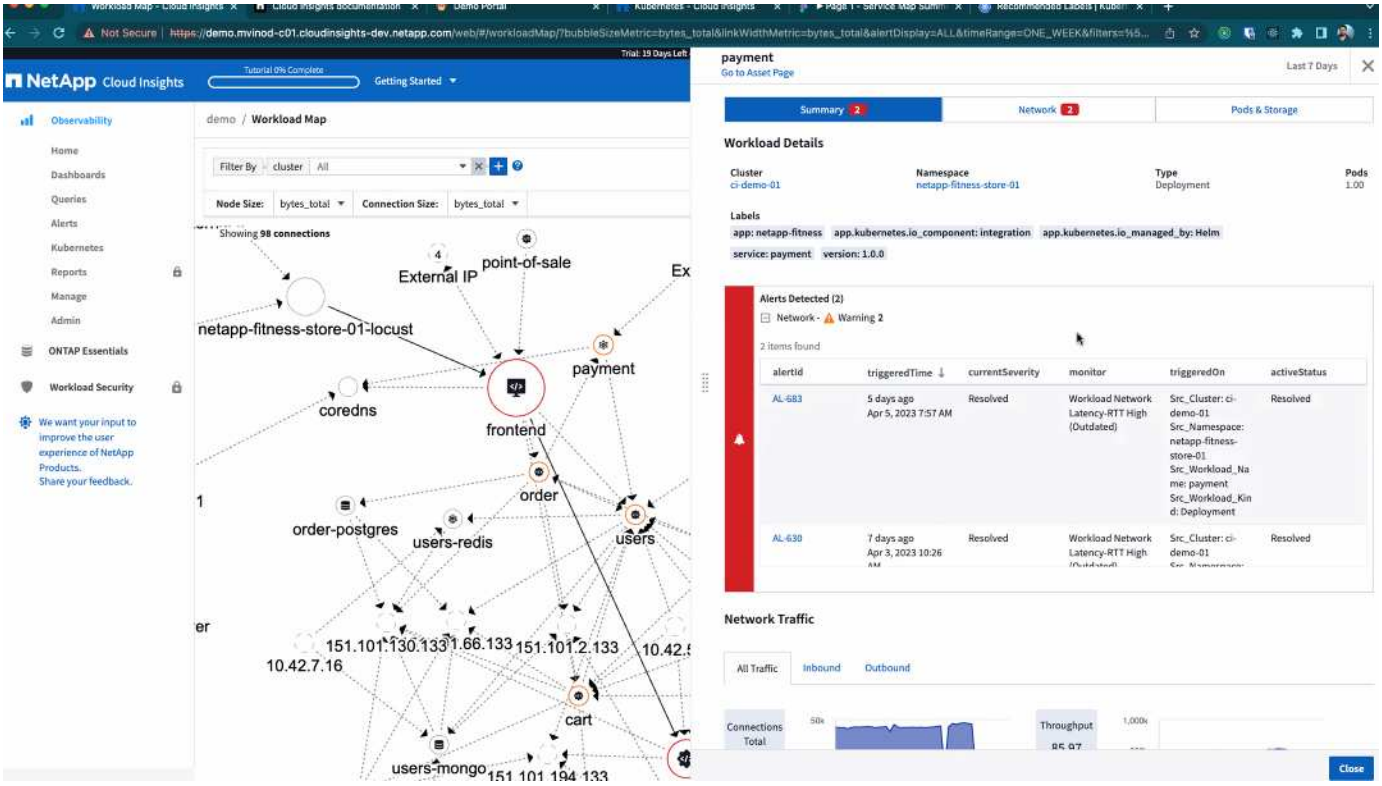
Os ícones dentro dos círculos representam diferentes tipos de serviços. Observe que os ícones só são visíveis se os objetos subjacentes tiveremrótulos .

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

Os números dentro dos círculos indicam o número de conexões externas atualmente processadas pelo serviço.

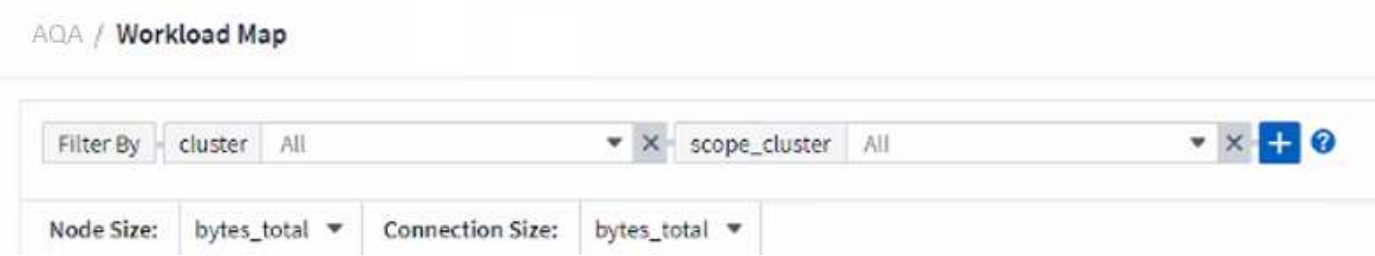
Detalhes e alertas da carga de trabalho

Círculos exibidos em cores indicam um alerta de nível crítico ou de advertência para a carga de trabalho. Passe o mouse sobre o círculo para ver um resumo do problema ou clique no círculo para abrir um painel deslizante com mais detalhes.

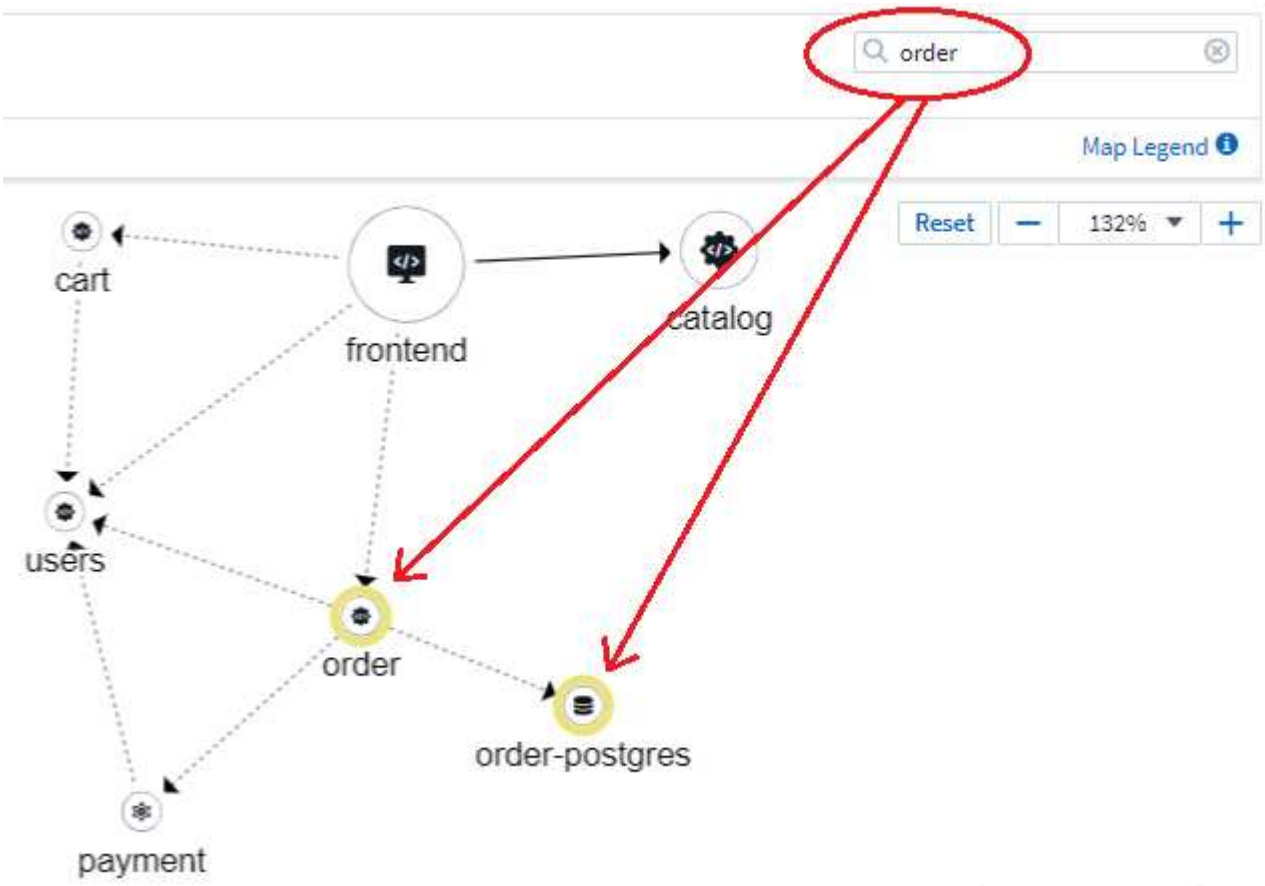


Encontrar e Filtrar

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



Da mesma forma, digitar uma sequência de caracteres no campo *Localizar* destacará as cargas de trabalho correspondentes.



Rótulos de carga de trabalho

Os rótulos de carga de trabalho são necessários se você quiser que o Mapa identifique os tipos de cargas de trabalho exibidas (ou seja, os ícones circulares). Os rótulos são derivados da seguinte forma:

- Nome do serviço/aplicativo em execução em termos genéricos
- Se a origem for um pod:
  - O rótulo é derivado do rótulo da carga de trabalho do pod
  - Rótulo esperado na carga de trabalho: app.kubernetes.io/component
  - Referência do nome do rótulo: https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels/
  - Etiquetas recomendadas:
    - front-end

- backend
- banco de dados
- esconderijo
- fila
- Kafka

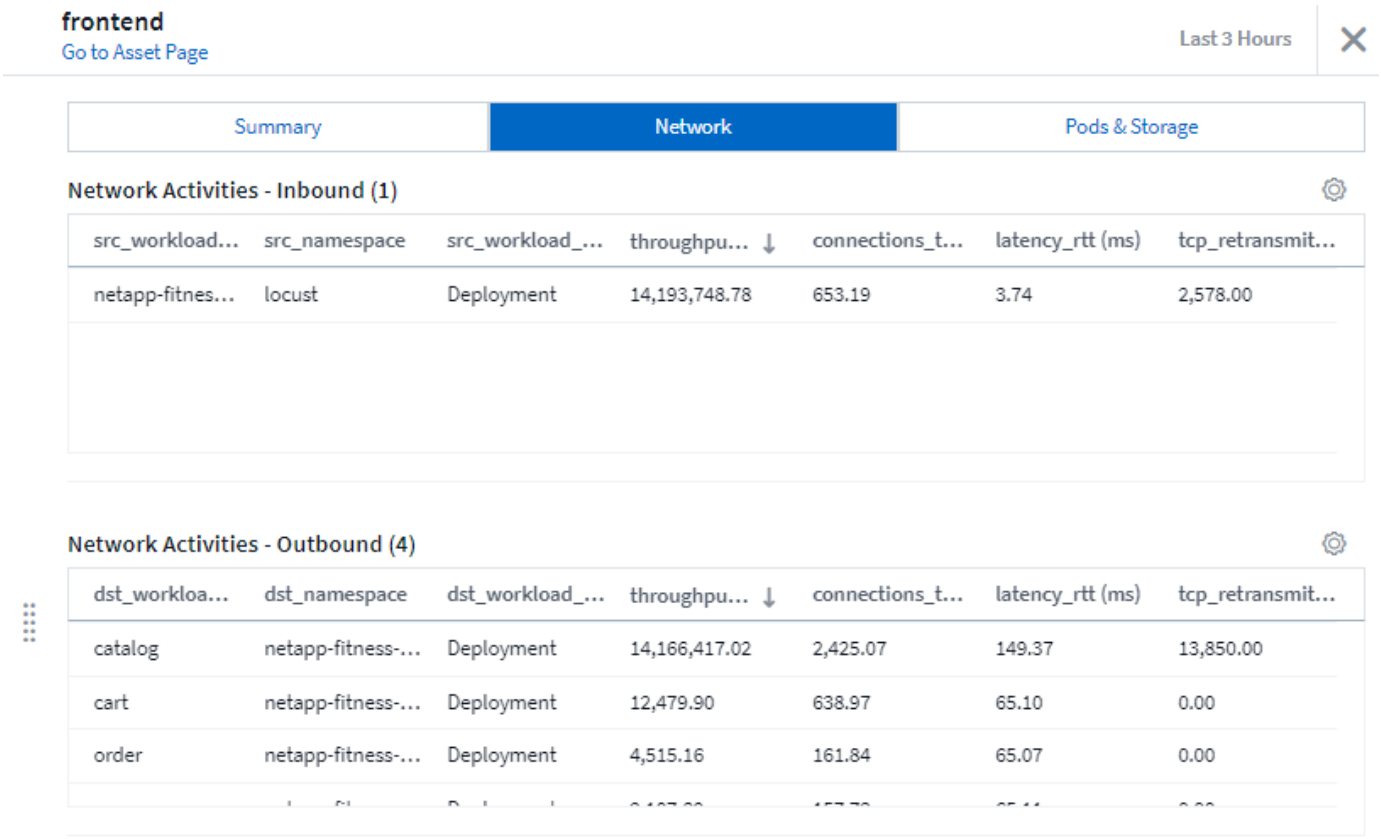
- Se a origem for externa ao cluster do Kubernetes:
  - O Data Infrastructure Insights tentará analisar o nome resolvido do DNS para extrair o tipo de serviço.

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

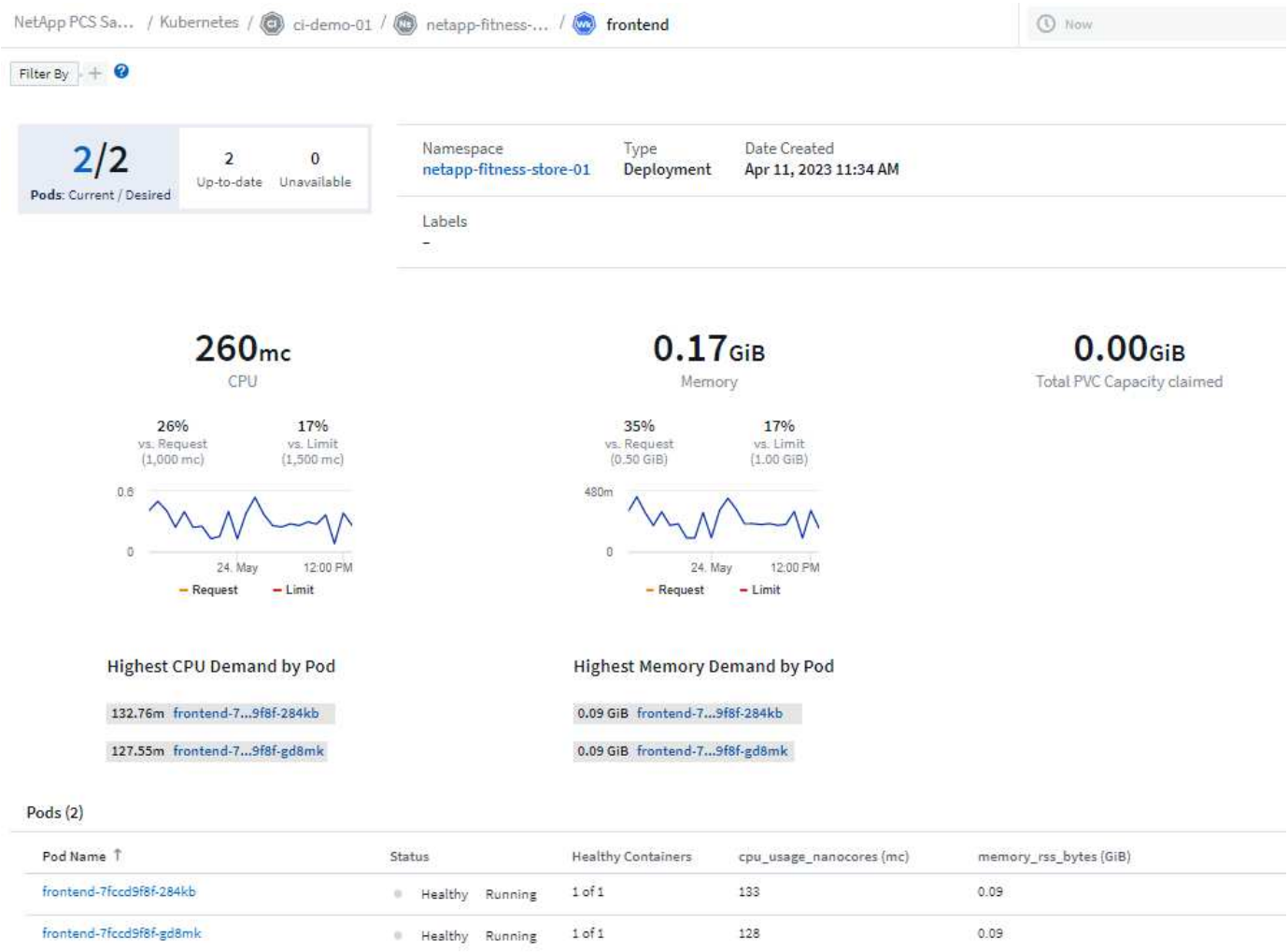
Mergulhe fundo

Clicar com o botão direito do mouse em uma carga de trabalho apresenta opções adicionais para você explorar mais a fundo. Por exemplo, aqui você pode ampliar para visualizar as conexões dessa carga de trabalho.

Ou você pode abrir o painel deslizante de detalhes para visualizar diretamente a guia *Resumo*, *Rede* ou *Pod e Armazenamento*.



Por fim, selecionar *Ir para página de ativos* abrirá a página inicial detalhada dos ativos para a carga de trabalho.



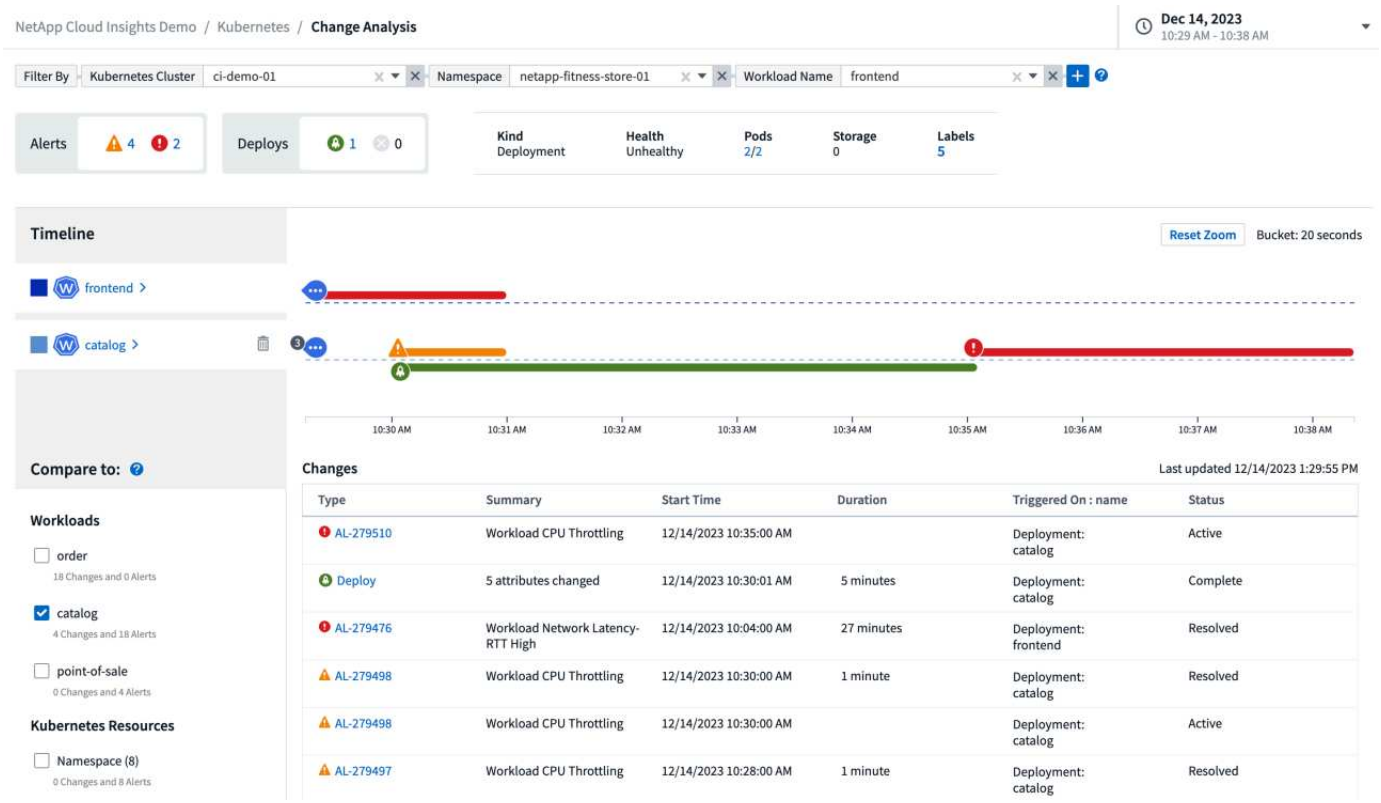
Análise de mudanças do Kubernetes

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

Como a Análise de Mudanças ajuda?

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

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



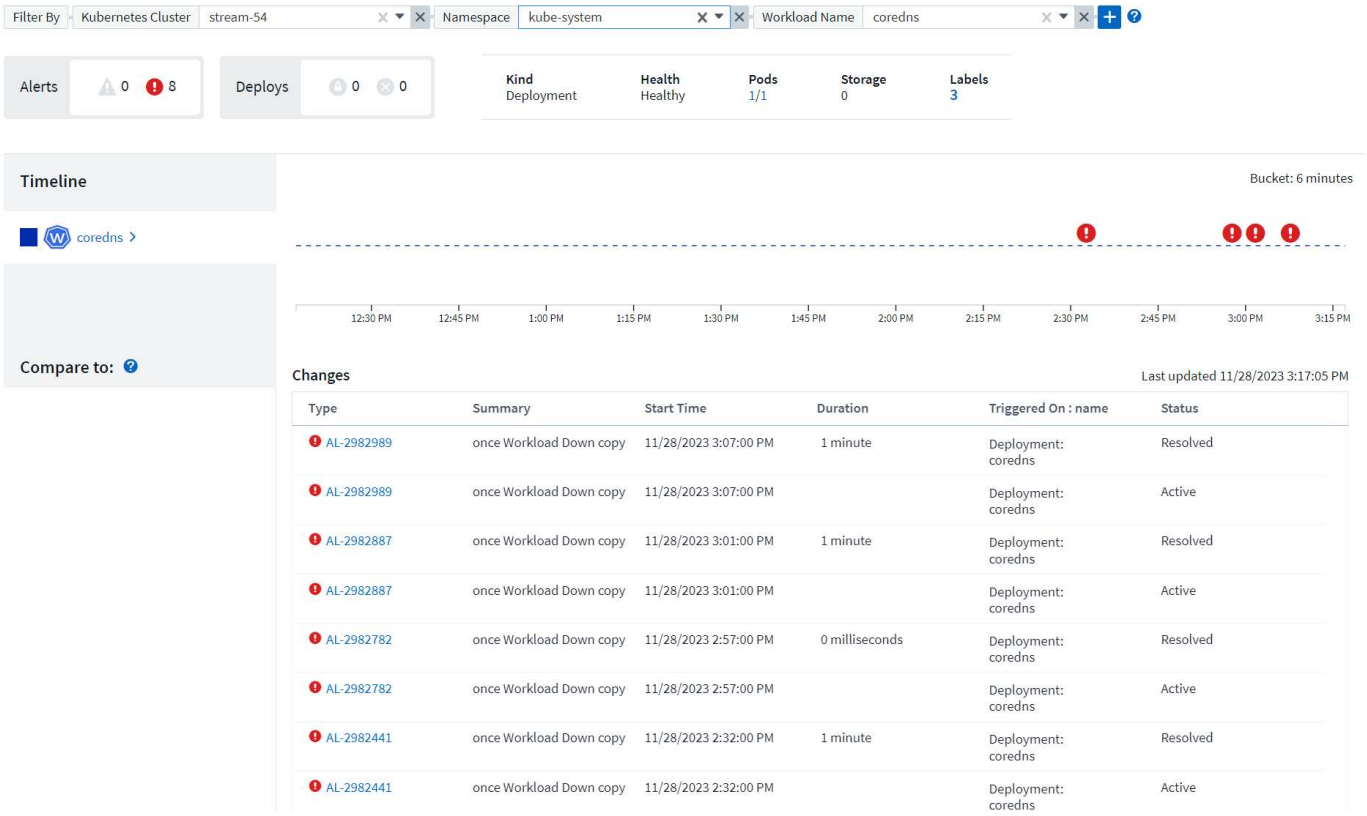
.

A página é atualizada automaticamente com base no intervalo de tempo do Data Infrastructure Insights selecionado no momento. Intervalos de tempo menores significam atualizações de tela mais frequentes.

Filtragem

Assim como acontece com todos os recursos do Data Infrastructure Insights, a filtragem da lista de alterações é intuitiva: na parte superior da página, insira ou selecione valores para seu cluster, namespace ou carga de trabalho do Kubernetes, ou adicione seus próprios filtros selecionando o botão {+].

Ao filtrar para um Cluster, Namespace e Carga de Trabalho específicos (junto com quaisquer outros filtros definidos), é exibido um cronograma de implantações e alertas para essa carga de trabalho naquele namespace naquele cluster. Amplie ainda mais clicando e arrastando no gráfico para focar em um intervalo de tempo mais específico.

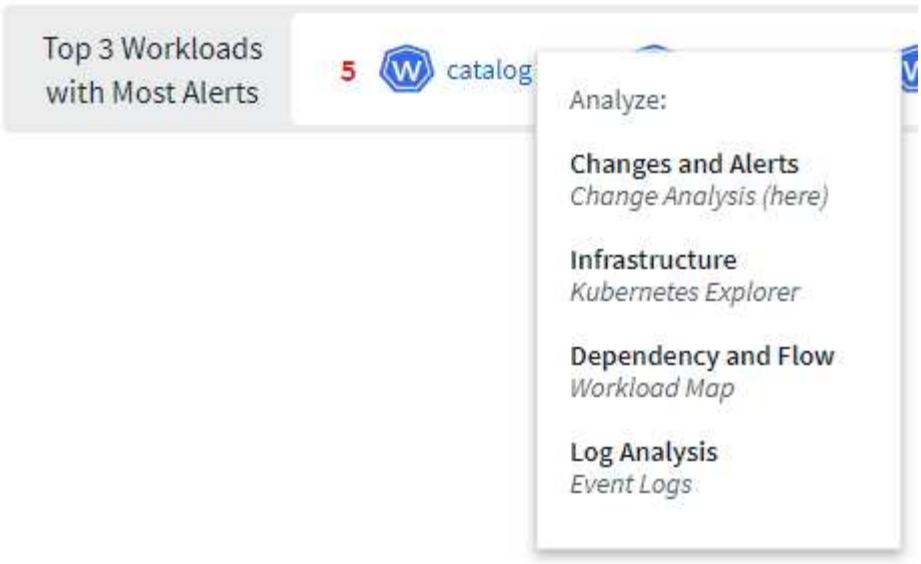


Status rápido

Abaixo da área de filtragem há uma série de indicadores de alto nível. À esquerda está o número de alertas (Aviso e Crítico). Este número inclui alertas *Ativos* e *Resolvidos*. Para ver apenas alertas *Ativos*, defina um filtro para "Status" e escolha "Ativo".

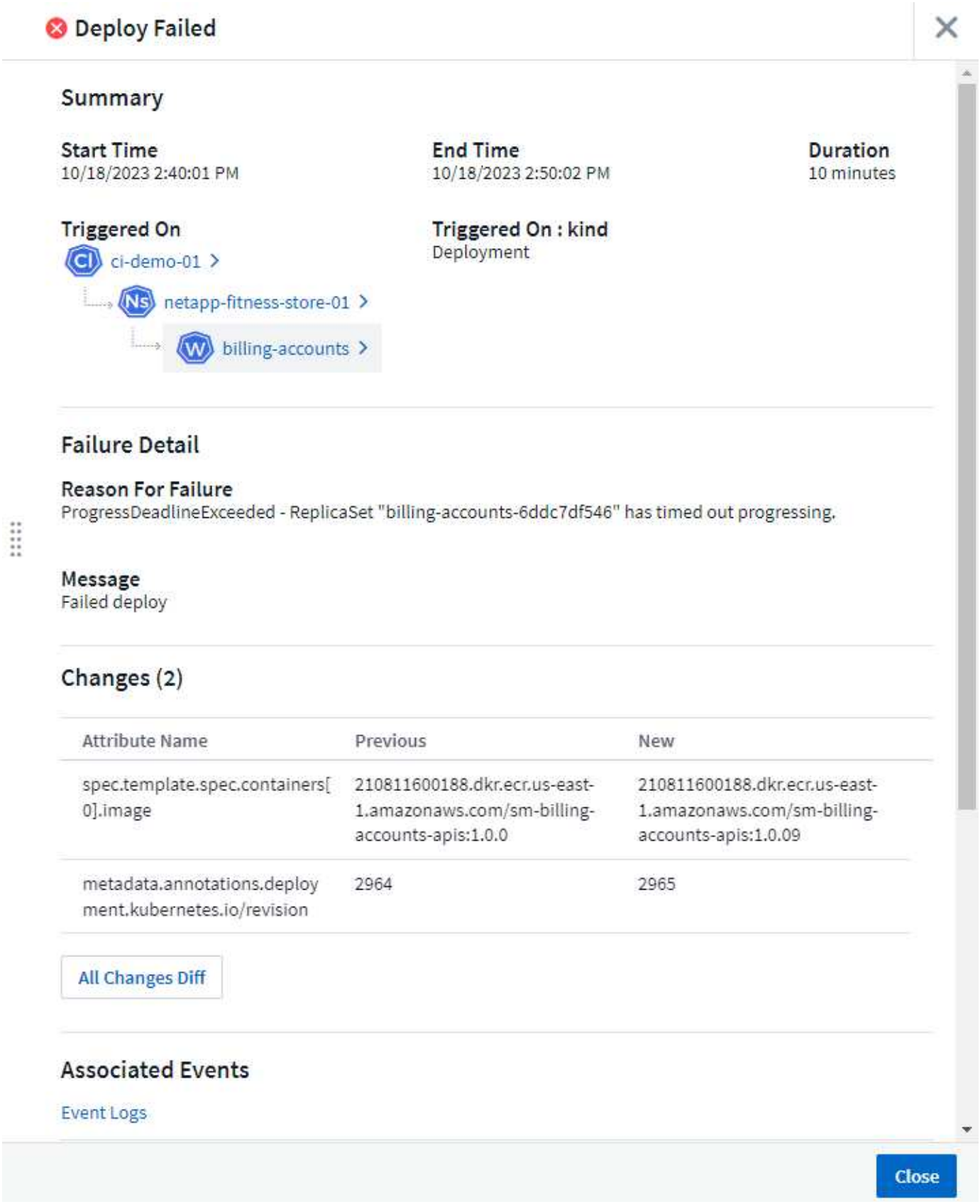
O status da implantação também é mostrado aqui. Novamente, o padrão é mostrar a contagem de implantações *Iniciadas*, *Concluídas* e *Falhadas*. Para ver apenas implantações *com falha*, defina um filtro para "Status" e selecione "Com falha".

As 3 principais cargas de trabalho com mais alertas são as próximas. 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 da carga de trabalho para explorar sua 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 que descreve a alteração com mais detalhes. Por exemplo, selecionar uma implantação com falha mostra um resumo da implantação, com horários de início e término, 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.



Selecionar um alerta também fornece detalhes sobre o alerta, incluindo o monitor que disparou o alerta, bem como um gráfico mostrando uma linha do tempo visual para o alerta.

Informações sobre direitos autorais

Informações sobre marcas comerciais