Kubernetes

Data Infrastructure Insights

NetApp
December 19, 2024

目次

Kubernetes

Kubernetes クラスタの概要

Data Infrastructure Insights Kubernetes Explorerは、Kubernetesクラスタの全体的な健全性と使用状況を表示するための強力なツールであり、調査領域に簡単にドリルダウンできます。

[Dashboards]>[Kubernetes Explorer]をクリックすると、[Kubernetes Cluster]リストページが開きます。この概要ページには、テナント上のKubernetesクラスタのテーブルが含まれます。

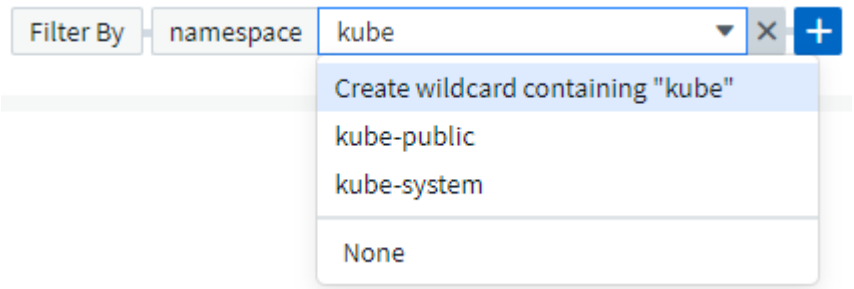
クラスタリスト

クラスタリストには、テナント上の各クラスタについて次の情報が表示されます。

- クラスタ*名*。クラスタ名をクリックすると、そのクラスタのが開き"* 詳細ページ *"ます。
- *彩度*パーセンテージ。Overall Saturation（全体飽和）は、CPU、メモリ、またはストレージの飽和度の最大値です。
- クラスタ内のノード数*。この番号をクリックすると、Node listページが開きます。
- クラスタ内の* Pod*の数。この番号をクリックすると、ポッドリストページが開きます。
- クラスタ内のネームスペースの数*。この番号をクリックすると、名前空間リストページが開きます。
- クラスタ内のワークロードの数*。この数値をクリックすると、ワークロードリストページが開きます。

フィルタを調整しています

フィルタ処理中に、入力を開始すると、現在のテキストに基づいて * ワイルドカードフィルタ * を作成するオプションが表示されます。このオプションを選択すると、ワイルドカード式に一致するすべての結果が返されます。NOTまたはANDを使用して*式*を作成することもできます。また、「なし」オプションを選択してフィールド内のnull値をフィルタリングすることもできます。



ワイルドカードまたは式に基づくフィルタ（例 フィルタフィールドには、「なし」などは濃い青で表示されます。リストから直接選択した項目は、水色で表示されます。

Kubernetes フィルタはコンテキストに応じて表示されます。つまり、特定のノードページ上にある場合、pod _name フィルタはそのノードに関連するポッドのみをリストします。さらに、特定のネームスペースにフィルタを適用すると、名前空間の名前空間ではポッドのみが表示されます。

ワイルドカードおよび式フィルタリングは、テキストまたはリストでは機能しますが、数値、日付、またはブール値では機能しません。

NetApp Kubernetes監視オペレータをインストールまたはアップグレードする前に

をインストールまたはアップグレードする前に、この情報をお読みください"Kubernetes監視オペレータ"。

コンポーネント	要件
Kubernetes のバージョン	Kubernetes v1.20以上：
Kubernetesディストリビューション	AWS Elastic Kubernetes Service（EKS）Azure Kubernetes Service（AKS）Google Kubernetes Engine（GKE）Red Hat OpenShift Rancher Kubernetes Engine（RKE）VMware Tanzu
Linux OS	Data Infrastructure Insightsでは、Arm64アーキテクチャで実行されているノードはサポートされません。ネットワーク監視：Linuxカーネルバージョン4.18.0以上を実行している必要があります。Photon OSはサポートされていません。
ラベル	Data Infrastructure Insightsは、Kubernetesノードセレクタを指定して、これらのプラットフォームで次のKubernetesラベルを検索することで、Linuxを実行しているKubernetesノードの監視をサポートします。Kubernetes v1.20以降：Kubernetes .io/os=linux Rancher + cattle.ioをオーケストレーション/ Kubernetesプラットフォーム：cattle.io/os=linux
コマンド	curlコマンドとkubectlコマンドが使用可能である必要があります。;最良の結果を得るには、これらのコマンドをパスに追加してください。

コンポーネント	要件
接続性	kubectl CLIはターゲットのKubernetesクラスタと通信するように設定されており、Data Infrastructure Insights環境にインターネット接続されています。インストール中にプロキシの背後にいる場合は、「オペレータのインストール」のセクションに記載されている手順に従ってください"プロキシサポートを設定しています"。監査およびデータレポートを正確に作成するには、Network Time Protocol（NTP；ネットワークタイムプロトコル）またはSimple Network Time Protocol（SNTP；簡易ネットワークタイムプロトコル）を使用してAgentマシンの時刻を同期します。
その他	OpenShift 4.6以降を実行している場合は、上記の前提条件を満たしていることを確認するだけでなく、も実行する必要があります"OpenShift の手順"。
APIトークン	Operatorを再デプロイする場合(つまり、Operatorを更新または置換する場合)は、新しいAPIトークンを作成する必要はありません。前のトークンを再利用できます。

始める前に注意すべき重要事項

を使用してを実行している場合プロキシ、カスタムリポジトリを使用している場合はOpenShift、次の項をよくお読みください。

についても読んでください権限。

プロキシサポートを設定しています

NetApp Kubernetes Monitoring Operatorをインストールするために、テナントでプロキシを使用できる場所は2つあります。同じプロキシシステムでも、別のプロキシシステムでもかまいません。

- インストールコードスニペット（「curl」を使用）の実行中に、スニペットが実行されるシステムをData Infrastructure Insights環境に接続するために必要なプロキシ
- ターゲットのKubernetesクラスタがData Infrastructure Insights環境と通信するために必要なプロキシ

これらのいずれかまたは両方にプロキシを使用する場合、NetApp Kubernetesオペレーティングモニタをインストールするには、まずプロキシがデータインフラストラクチャインサイト環境との良好な通信を許可するように設定されていることを確認する必要があります。たとえば、Operatorをインストールするサーバ/ VMからData Infrastructure Insightsにアクセスし、Data Infrastructure Insightsからバイナリをダウンロードできるようにする必要があります。

NetApp Kubernetes Operating Monitorのインストールに使用するプロキシとして、オペレータをインストールする前に、_http_proxy/https_proxy_environment変数を設定します。一部のプロキシ環境では'_no_proxy環境変数も設定する必要があります

変数を設定するには、NetApp Kubernetes Monitoring Operatorをインストールする前に、システムで次の手順を実行します。

1. 現在のユーザの *https_proxy* 変数と _http_proxy_environment 変数を設定します。

a. セットアップするプロキシに認証（ユーザ名/パスワード）がない場合は、次のコマンドを実行します。

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

.. セットアップするプロキシに認証（ユーザ名/パスワード）が設定されている場合は、次のコマンドを実行します。

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

KubernetesクラスタがData Infrastructure Insights環境と通信するために使用するプロキシの場合は、これらの手順をすべて読んでからNetApp Kubernetes監視オペレータをインストールします。

NetApp Kubernetes Monitoring Operatorを導入する前に、operator-config.yamlでAgentConfigurationのプロキシセクションを設定します。

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

カスタムまたはプライベートの**Docker**リポジトリを使用する

デフォルトでは、NetApp Kubernetes監視オペレータは、データインフラのインサイトリポジトリからコンテナイメージを取得します。監視のターゲットとして使用されているKubernetesクラスタがあり、カスタムまたはプライベートのDockerリポジトリまたはコンテナレジストリからのみコンテナイメージを取得するようにそのクラスタが設定されている場合は、NetApp Kubernetes Monitoring Operatorで必要なコンテナへのアク

セスを設定する必要があります。

NetApp Monitoring Operatorのインストールタイルから[Image Pull Snippet]を実行します。このコマンドを実行すると、Data Infrastructure Insightsリポジトリにログインし、オペレータが必要とするすべてのイメージを取得して、Data Infrastructure Insightsリポジトリからログアウトします。プロンプトが表示されたら、指定したリポジトリの一時パスワードを入力します。このコマンドは、オプション機能を含む、オペレータが使用するすべてのイメージをダウンロードします。これらの画像がどの機能に使用されるかについては、以下を参照してください。

Core Operator Functionality and Kubernetes Monitoringの略

- ネットアップによる監視
- kube-rbac-proxyの略
- kube-state-metricsの略
- テレグラフ
- distroless-root-user

イベントログ

- Fluent-bit
- kubernetes-event-exporterの略

ネットワークのパフォーマンスとマップ

- ci-net-observerの略

社内のポリシーに従って、オペレータ用の Docker イメージをプライベート / ローカル / エンタープライズ Docker リポジトリにプッシュします。リポジトリ内のこれらのイメージへのイメージタグとディレクトリパスが、Data Infrastructure Insightsリポジトリ内のイメージタグとディレクトリパスと一致していることを確認します。

operator-deployment.yamlでmonitoring-operatorデプロイメントを編集し、プライベートDockerリポジトリを使用するようにすべてのイメージ参照を変更します。

```
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>
```

operator-config.yamlのAgentConfigurationを編集して、新しいDockerリポジトリの場所を反映します。プライベートリポジトリ用に新しいimagePullSecretを作成します。詳細については、_ https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/_を参照してください

```
agent:
  ...
  # An optional docker registry where you want docker images to be pulled
from as compared to CI's docker registry
  # Please see documentation for
xref:{relative_path}task_config_telegraf_agent_k8s.html#using-a-custom-or-
private-docker-repository[using a custom or private docker repository].
  dockerRepo: your.docker.repo/long/path/to/test
  # Optional: A docker image pull secret that maybe needed for your
private docker registry
  dockerImagePullSecret: docker-secret-name
```

OpenShift の手順

OpenShift 4.6以降で実行している場合は、_runPrivileged_settingを有効にするには、_operator-config.yaml_でAgentConfigurationを編集する必要があります。

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

OpenShiftは、一部のKubernetesコンポーネントへのアクセスをブロックする可能性のある追加のセキュリティレベルを実装する場合があります。

権限

監視しているクラスタにClusterRoleがないカスタムリソースが含まれている場合は"表示するアグリゲート"、イベントログを使用してリソースを監視するために、オペレータにこれらのリソースへのアクセス権を手動で付与する必要があります。

1. edit_operator -additional-permissions.yaml_インストール前、またはインストール後にresource_ClusterRole/<namespace>-additional-permissions_を編集します。
2. 動詞["get","watch","list"]を使用して、目的のapiGroupsとリソースの新しいルールを作成します。「\https://kubernetes.io/docs/reference/access-authn-authz/rbac/」を参照
3. クラスタに変更を適用します。

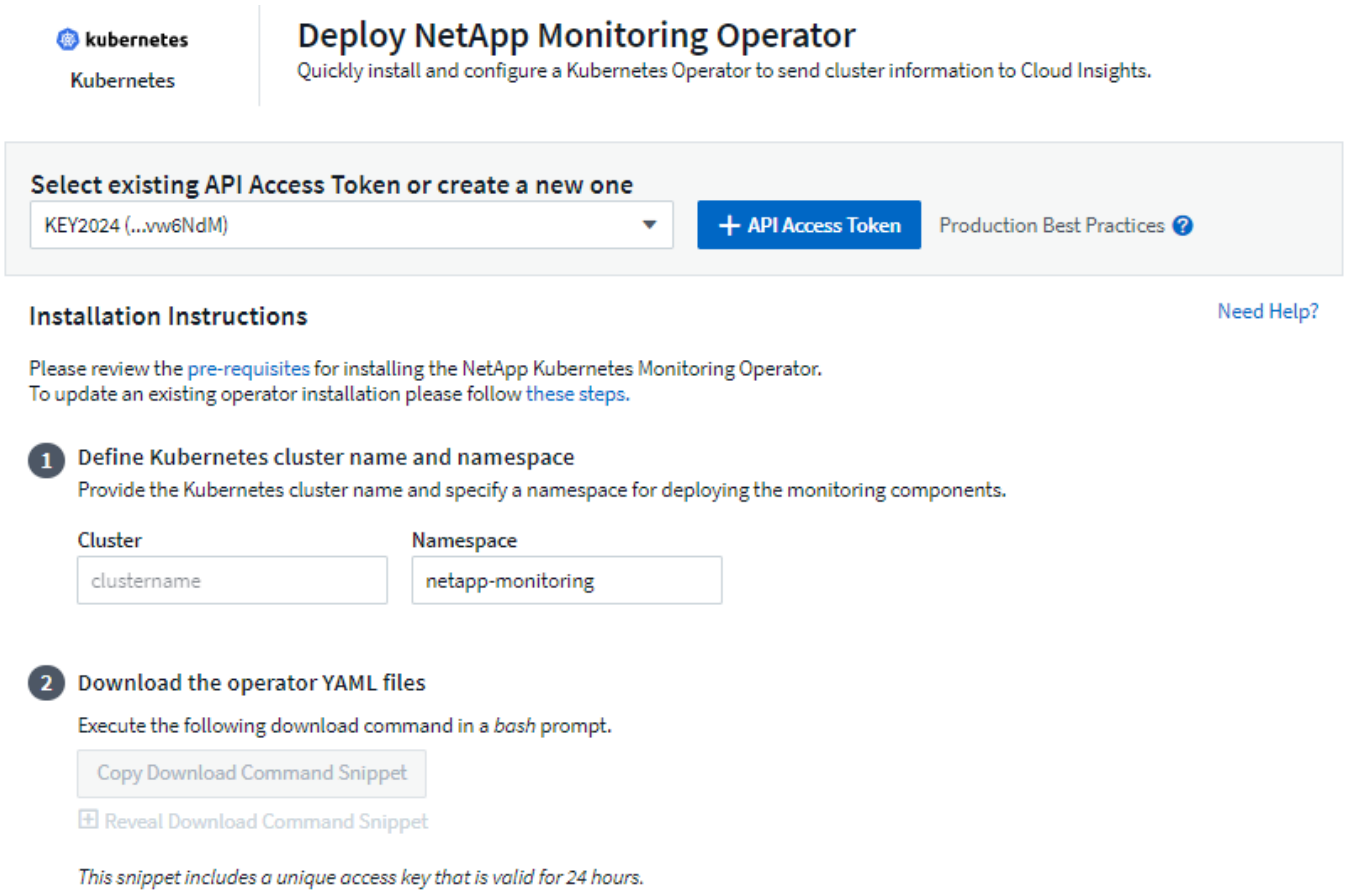
Kubernetes Monitoring Operatorのインストールと設定

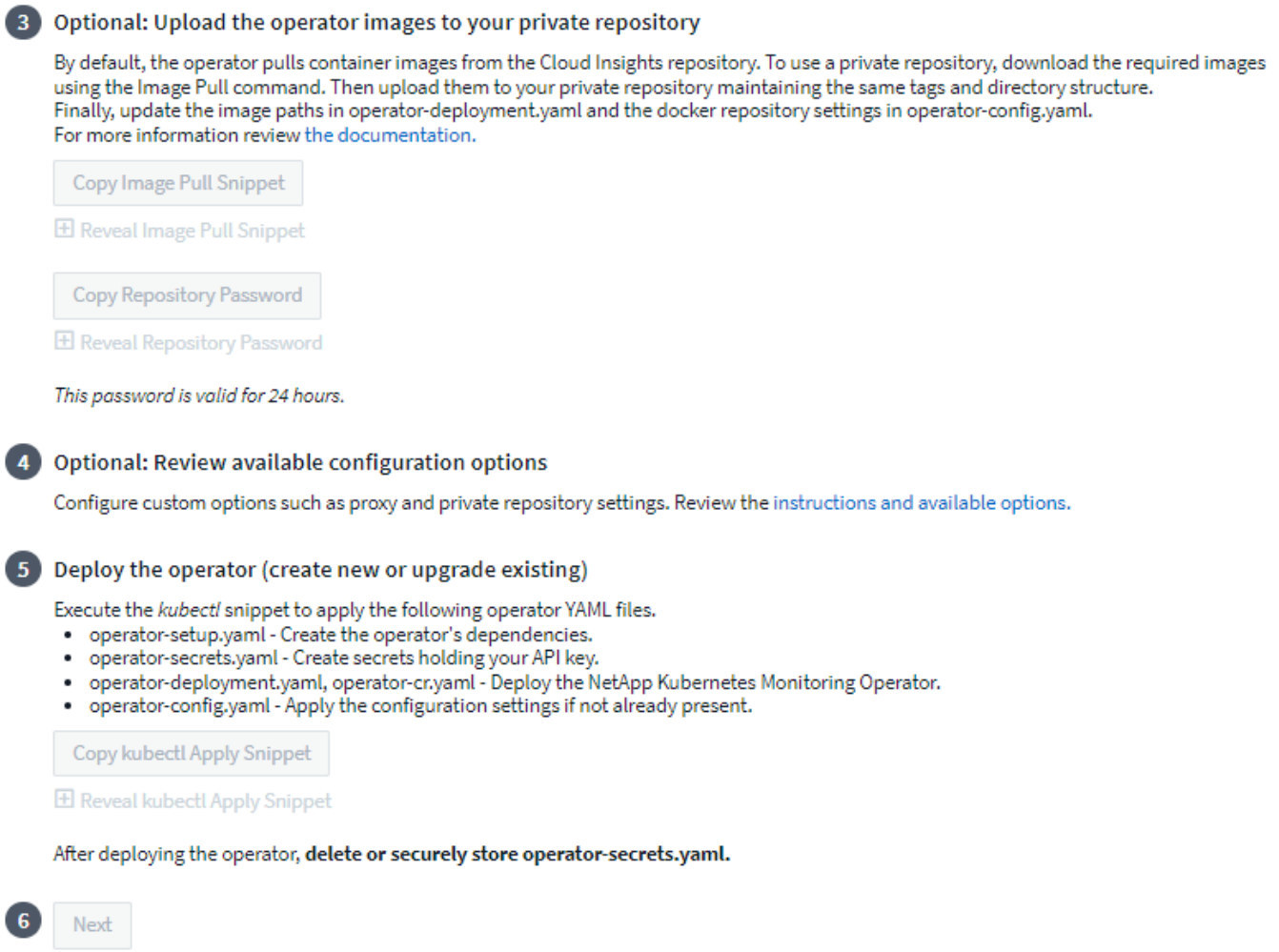
Data Infrastructure Insightsは、Kubernetesコレクション向けの「Kubernetes Monitoring Operator」を提供しています。新しいオペレータを導入するには、* Kubernetes > Collectors >+ Kubernetes Collector *に移動します。

Kubernetes Monitoring Operatorをインストールする前に

Kubernetes Monitoring Operatorをインストールまたはアップグレードする前に、ドキュメントを参照してください"前提条件"。

Kubernetes Monitoring Operatorのインストール





Kubernetesに**Kubernetes Monitoring Operator**エージェントをインストールする手順は次のとおりです。

1. 一意のクラスタ名およびネームスペースを入力してください。以前のKubernetes Operatorの場合はアップグレード、同じクラスタ名とネームスペースを使用します。
2. これらを入力すると、ダウンロードコマンドスニペットをクリップボードにコピーできます。
3. スニペットを a_bash_window に貼り付け、実行します。Operatorインストールファイルがダウンロードされます。スニペットには固有のキーがあり、24時間有効です。
4. カスタムリポジトリまたはプライベートリポジトリがある場合は、オプションのImage Pullスニペットをコピーし、a_bash_shellに貼り付けて実行します。画像がプルされたら、プライベートリポジトリにコピーします。必ず同じタグとフォルダ構造を維持してください。_operator-deployment.yaml_のパスと_operator-config.yaml_のDockerリポジトリ設定を更新します。
5. 必要に応じて、プロキシやプライベートリポジトリの設定など、使用可能な設定オプションを確認します。あなたはについてもっと読むことができます"設定オプション"。
6. 準備ができたら、kubectl Applyスニペットをコピーしてダウンロードし、実行してOperatorをデプロイします。
7. インストールが自動的に開始されます。完了したら、[*Next*]ボタンをクリックします。
8. インストールが完了したら、[*Next*]ボタンをクリックします。また、_operator-secrets.yaml_fileを削除するか、安全に保存してください。

プロキシを使用している場合は、「について」を参照してくださいプロキシを設定します。

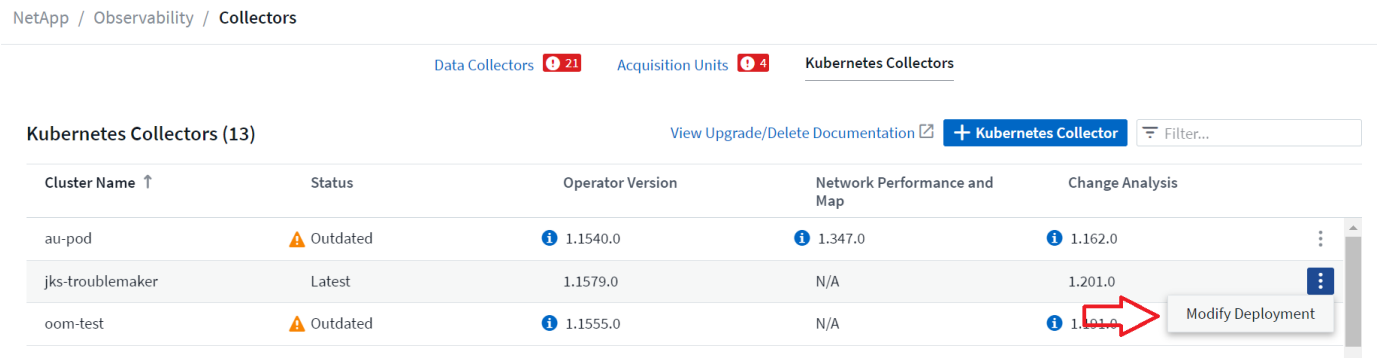
カスタムリポジトリがある場合は、を参照してくださいカスタム/プライベートDockerリポジトリを使用する。

Kubernetes監視コンポーネント

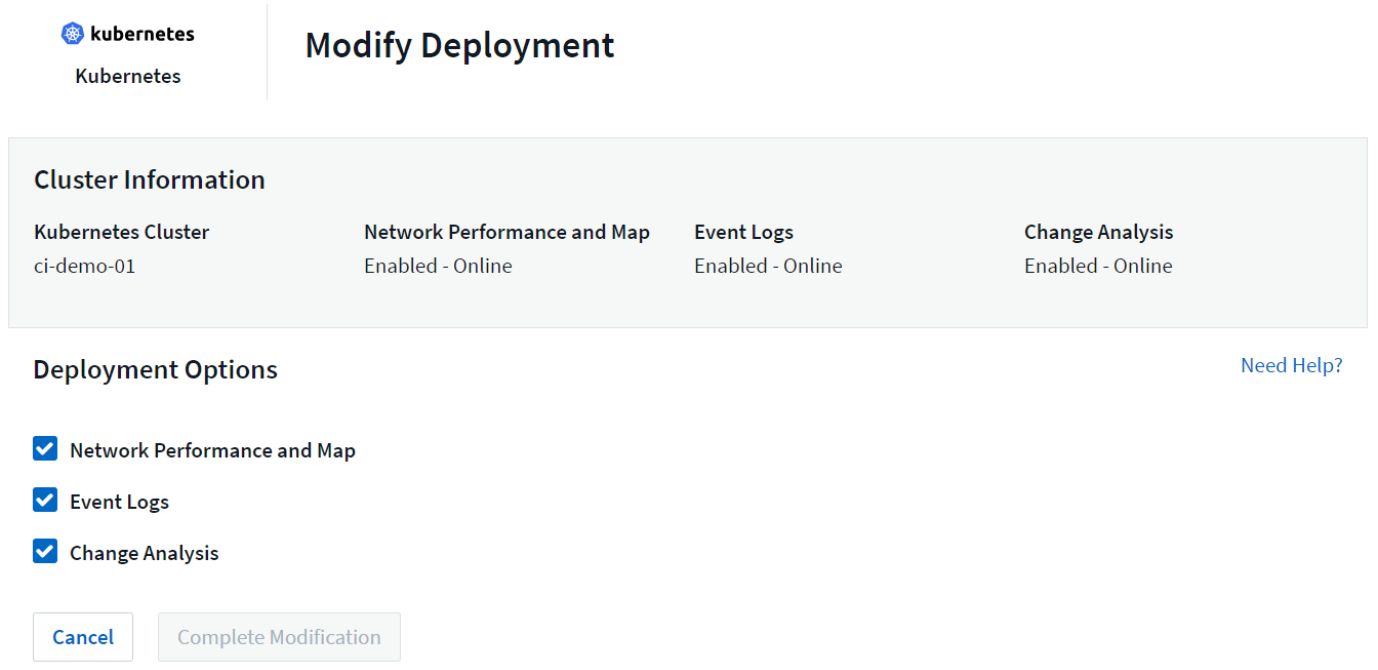
Data Infrastructure Insights Kubernetes Monitoringは、次の4つの監視コンポーネントで構成されます。

- クラスタ指標
- ネットワークパフォーマンスとマップ（オプション）
- イベントログ（オプション）
- 変更分析（オプション）

上記のオプションコンポーネントは、各Kubernetesコレクタに対してデフォルトで有効になっています。特定のコレクタ用のコンポーネントが必要ないと判断した場合は、* Kubernetes > Collectors *に移動し、画面右側のコレクタの「three dots」メニューから_Modify Deployment_を選択して無効にできます。



画面には各コンポーネントの現在の状態が表示され、必要に応じてそのコレクタのコンポーネントを無効または有効にすることができます。



最新のKubernetes Monitoring Operatorへのアップグレード

既存のOperatorにAgentConfigurationが存在するかどうかを確認します（ネームスペースがdefault_netapp-monitoring_でない場合は、適切なネームスペースに置き換えてください）。

```
 kubectl -n netapp-monitoring get agentconfiguration netapp-monitoring-configuration
AgentConfigurationが存在する場合：
```

- インストール既存の演算子の上にある最新の演算子。
  - カスタムリポジトリを使用している場合は、使用していることを確認して最新のコンテナイメージを取得しますください。

AgentConfigurationが存在しない場合は、次の手順を実行します。

- クラスタ名がData Infrastructure Insightsで認識される名前であることをメモします（ネームスペースがデフォルトのNetApp監視機能でない場合は、適切なネームスペースで置き換えてください）。

```
 kubectl -n netapp-monitoring get agent -o jsonpath='{.items[0].spec.cluster-name}'
* 既存の
Operatorのバックアップを作成します（ネームスペースがデフォルトのネットアップ監視機能になっていない場合は、適切なネームスペースで置き換えてください）。
```

```
 kubectl -n netapp-monitoring get agent -o yaml > agent_backup.yaml
* <<to-remove-the-kubernetes-monitoring-operator,アンインストール>>既存の演算子。
* <<installing-the-kubernetes-monitoring-operator,インストール>>最新の演算子。
```

  - 同じクラスタ名を使用してください。
  - 最新のOperator YAMLファイルをダウンロードしたら、展開する前に、agent_backup.yamlにあるカスタマイズをダウンロードしたoperator-config.yamlに移植します。
  - カスタムリポジトリを使用している場合は、使用していることを確認して最新のコンテナイメージを取得しますください。

Kubernetes Monitoring Operatorの停止と起動

Kubernetes Monitoring Operatorを停止するには：

```
 kubectl -n netapp-monitoring scale deploy monitoring-operator
--replicas=0
Kubernetes Monitoring Operatorを起動するには：
```

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

アンインストール中です

Kubernetes Monitoring Operatorを削除するには

Kubernetes Monitoring Operatorのデフォルトのネームスペースは「netapp-monitoring」です。独自のネームスペースを設定した場合は、それらのネームスペースと、以降のすべてのコマンドおよびファイルを置き換えます。

新しいバージョンの監視オペレータは、次のコマンドを使用してアンインストールできます。

```
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>
```

監視オペレータが専用のネームスペースに配置されている場合は、ネームスペースを削除します。

```
 kubectl delete ns <NAMESPACE>
最初のコマンドが「リソースが見つかりません」を返した場合は、次の手順に従って古いバージョンの監視オペレータをアンインストールします。
```

次の各コマンドを順番に実行します。現在のインストール状況によっては、これらのコマンドの一部で「オブジェクトが見つかりません」というメッセージが返される場合があります。これらのメッセージは無視してかまいません。

```
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>
```

セキュリティコンテキスト制約が事前に作成されている場合は、次の手順を実行します。

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

Kubeステートメトリックについて

NetApp Kubernetes Monitoring Operatorは、他のインスタンスとの競合を回避するために独自のkube-state-metricsをインストールします。

Kube-State-Metricsの詳細については、を参照してください"このページです"。

オペレータの設定/カスタマイズ

これらのセクションでは、オペレータ設定のカスタマイズ、プロキシの操作、カスタムまたはプライベートDockerリポジトリの使用、OpenShiftの操作について説明します。

設定オプション

最も一般的に変更される設定は、_AgentConfiguration_customリソースで構成できます。オペレータを配備する前に、_operator-config.yaml_fileを編集して、このリソースを編集できます。このファイルには、コメントアウトされた設定例が含まれています。演算子の最新バージョンについては、のリストを参照してください"使用可能な設定"。

オペレータが配備された後で、次のコマンドを使用してこのリソースを編集することもできます。

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

展開したオペレータのバージョンが`AgentConfiguration`をサポートしているかどうかを確認するには、次のコマンドを実行します。

```
kubectl get crd agentconfigurations.monitoring.netapp.com
```
「`Error from server (NotFound)`」というメッセージが表示された場合は、`AgentConfiguration`を使用する前にオペレータをアップグレードする必要があります。

プロキシサポートを設定しています

Kubernetes Monitoring Operatorをインストールするために、テナントでプロキシを使用できる場所は2つあります。同じプロキシシステムでも、別のプロキシシステムでもかまいません。

- インストールコードスニペット（「curl」を使用）の実行中に、スニペットが実行されるシステムをData Infrastructure Insights環境に接続するために必要なプロキシ
- ターゲットのKubernetesクラスタがData Infrastructure Insights環境と通信するために必要なプロキシ

これらのいずれかまたは両方にプロキシを使用する場合、Kubernetes Operating Monitorをインストールするには、まず、Data Infrastructure Insights環境との通信が良好になるようにプロキシが設定されていることを確認する必要があります。プロキシがあり、Operatorをインストールするサーバ/ VMからData Infrastructure Insightsにアクセスできる場合は、プロキシが適切に設定されている可能性があります。

Kubernetes Operating Monitorのインストールに使用するプロキシについては、Operatorをインストールする前に、_http_proxy/https_proxy_environment変数を設定します。一部のプロキシ環境では'_no_proxy環境変数も設定する必要があります

変数を設定するには、Kubernetes Monitoring Operatorをインストールする前に、システム*で次の手順を実行します。

1. 現在のユーザの *https_proxy* 変数と _http_proxy_environment 変数を設定します。
   a. セットアップするプロキシに認証（ユーザ名/パスワード）がない場合は、次のコマンドを実行します。

```
export https_proxy=<proxy_server>:<proxy_port>
```
. . セットアップするプロキシに認証（ユーザ名/パスワード）が設定されている場合は、次のコマンドを実行します。

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

KubernetesクラスタがData Infrastructure Insights環境と通信するために使用するプロキシの場合は、以下の手順をすべて読んでからKubernetes Monitoring Operatorをインストールします。

Kubernetes Monitoring Operatorをデプロイする前に、operator-config.yamlのAgentConfigurationのプロキシセクションを設定します。

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

カスタムまたはプライベートの**Docker**リポジトリを使用する

Kubernetes監視オペレータは、デフォルトで、Data Infrastructure Insightsリポジトリからコンテナイメージを取得します。監視のターゲットとして使用されているKubernetesクラスタがあり、そのクラスタがカスタムまたはプライベートのDockerリポジトリまたはコンテナレジストリからコンテナイメージのみをプルするように構成されている場合は、Kubernetes Monitoring Operatorが必要とするコンテナへのアクセスを設定する必要があります。

NetApp Monitoring Operatorのインストールタイルから[Image Pull Snippet]を実行します。このコマンドを実行すると、Data Infrastructure Insightsリポジトリにログインし、オペレータが必要とするすべてのイメージを取得して、Data Infrastructure Insightsリポジトリからログアウトします。プロンプトが表示されたら、指定したリポジトリの一時パスワードを入力します。このコマンドは、オプション機能を含む、オペレータが使用するすべてのイメージをダウンロードします。これらの画像がどの機能に使用されるかについては、以下を参照してください。

Core Operator Functionality and Kubernetes Monitoringの略

- ネットアップによる監視
- ci-kube-rbac-proxy
- CI-KSM
- CI-テレグラフ
- distroless-root-user

イベントログ

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

ネットワークのパフォーマンスとマップ

- ci-net-observerの略

社内のポリシーに従って、オペレータ用の Docker イメージをプライベート / ローカル / エンタープライズ Docker リポジトリにプッシュします。リポジトリ内のこれらのイメージへのイメージタグとディレクトリパスが、Data Infrastructure Insightsリポジトリ内のイメージタグとディレクトリパスと一致していることを確認します。

operator-deployment.yamlでmonitoring-operatorデプロイメントを編集し、プライベートDockerリポジトリを使用するようにすべてのイメージ参照を変更します。

```
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>
```

operator-config.yamlのAgentConfigurationを編集して、新しいDockerリポジトリの場所を反映します。プライベートリポジトリ用に新しいimagePullSecretを作成します。詳細については、_ https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/_を参照してください

```
agent:
  ...
  # An optional docker registry where you want docker images to be pulled
from as compared to CI's docker registry
  # Please see documentation link here:
xref:{relative_path}task_config_telegraf_agent_k8s.html#using-a-custom-or-
private-docker-repository
  dockerRepo: your.docker.repo/long/path/to/test
  # Optional: A docker image pull secret that maybe needed for your
private docker registry
  dockerImagePullSecret: docker-secret-name
```

OpenShift の手順

OpenShift 4.6以降で実行している場合は、_runPrivileged_settingを有効にするには、_operator-config.yaml_でAgentConfigurationを編集する必要があります。

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

OpenShiftは、一部のKubernetesコンポーネントへのアクセスをブロックする可能性のある追加のセキュリテ

ィレベルを実装する場合があります。

公差と接線（**Tolerations and Taints**）

NetApp-ci-telegraf-ds_、*NetApp-CI-fluent-bit-ds*、および_NetApp-CI-net-observer-l4-DS_DaemonSetsは、すべてのノードのデータを正しく収集するために、クラスタ内のすべてのノードでポッドをスケジュールする必要があります。オペレータは、いくつかの既知の*テイント*に耐えられるように設定されています。ノードにカスタムのtaintsを設定して、すべてのノードでポッドが実行されないようにしている場合は、それらのtaintsに* toleration *を作成できます"（*AgentConfiguration*）をクリックします"。クラスタ内のすべてのノードにカスタムテイントを適用した場合は、オペレータの導入に必要な許容範囲を追加して、オペレータポッドをスケジュールおよび実行できるようにする必要があります。

Kubernetesの詳細はこちら"塗料および耐性"をご覧ください。

に戻ります。"* NetApp Kubernetes監視オペレータのインストール*ページ"

秘密に関する注意事項

Kubernetes Monitoring Operatorのシークレットをクラスタ全体で表示する権限を削除するには、インストール前に_operator-setup.yaml_fileから次のリソースを削除します。

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

アップグレードの場合は、クラスタからリソースも削除します。

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

変更分析が有効になっている場合は、_AgentConfiguration_or_operator -config.yaml_を変更して、変更管理セクションのコメントを解除し、変更管理セクションの下に_kindsToIgnoreFromWatch：'"secrets"'_を含めます。この行の一重引用符と二重引用符の存在と位置に注意してください。

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

Kubernetes Monitoring Operatorイメージシグネチャの確認

オペレータ用のイメージと、展開するすべての関連イメージは、NetAppによって署名されています。インストール前にcosignツールを使用してイメージを手動で検証するか、Kubernetesアドミッションコントローラを設定できます。詳細については、を参照してください"Kubernetes のドキュメント"。

イメージシグネチャの検証に使用する公開キーは、Monitoring Operatorインストールタイルの_オプションで使用できます。オペレータイメージをプライベートリポジトリにアップロード> Image Signature Public Key_

画像折丁を手動で確認するには、次の手順に従います。

1. 画像プルスニペットをコピーして実行する
2. プロンプトが表示されたら、リポジトリパスワードをコピーして入力します。
3. イメージ署名公開キーを保存します（この例ではdii-image-signing.pub）。
4. コサインを使用して画像を確認します。次のcosignの使用例を参照してください。

```
$ 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}]
```

トラブルシューティング

Kubernetes Monitoring Operatorの設定で問題が発生した場合に試すべきこと：

問題	次の操作を実行します
Kubernetes 永続ボリュームと対応するバックエンドストレージデバイスの間にハイパーリンク / 接続がありません。My Kubernetes Persistent Volume がストレージサーバのホスト名を使用して設定されます。	手順に従って既存の Tegraf エージェントをアンインストールし、最新の Tegraf エージェントを再インストールします。Telegrafバージョン2.0以降を使用しており、KubernetesクラスタストレージがData Infrastructure Insightsによってアクティブに監視されている必要があります。

問題	次の操作を実行します
E0901 15:21:39.962145 1 reflector.go:178]k8s.io/kube-state-metrics/internal/store/builder.go:352: List*v1.MutatingWebhookConfiguration:サーバはリクエストされたリソースE0901 15:21:43.168161を見つけることができませんでした。	これらのメッセージは、1.20より前のバージョンのKubernetesでkube-state-metricsバージョン2.0.0以上を実行している場合に発生する可能性があります。Kubernetes のバージョンを取得するには、次の Leubectl version_ kbe-state-metrics バージョンを取得します。 *kubectl* デプロイ */kube-state-metrics -o jsonpath='{.image}'* これらのメッセージが発生しないようにするには、 kube-state-metrics デプロイを修正して、次の Leases 設定を具体的に無効にしてください。 _hookates_web_volumeconfigurations resources= 証明リクエスト , configmaps,cronjobs,demonsets,horizontalscalers,ingleers,jobs,limitrange,scapers,networkpolicies , nodes,persistentvolumes,persistentvolumesalims,persistentvolumes,podeters, replicaSets,replicaSets,replicationcontrollers ,residetodポッド ,residetappeditors,appers,uns,uns,uns,uns,sets,uns,uns,uns,uns,uns,sets,uns,sets,uns,sets,uns,uns,sets,uns,uns,sets,uns,uns,uns,wodecodeclieticecodetics,sets,sets,sets,sets,uns,sets,uns,uns,sets,sets,sets,un 検証する Web フック設定 ' ボリュームの添付ファイル
Telegrafから次のようなエラーメッセージが表示されますが、Telegrafは起動して実行されます。10月11日14:23:41 IP-172-31-39-47 systemd[1]: InfluxDBにメトリックを報告するために、プラグイン駆動のサーバーエージェントを起動しました。10月11日14：23：41 IP-172-31-39-47 telegraf [1827]：time="2021-10-11T14：23：41Z" level=error msg="failed to create cache directory./etc/telegraf/.cache/snowflake、err:mkdir /etc/telegraf/.ca che: permission denied.ignored \n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" Oct 11 14:23:41 ip-172-31-39-47 telegrafaf [1827]無視されました。open /etc/telegraf/.cache/snowflake/ocsp_response_cache.json:該当するファイルまたはディレクトリがありません\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" Oct 11 14:23:41 ip-172-31-39-47 telegrZ [1827]: 2021-T1114:114:114Telegraf 1.19.3 を起動しています	これは問題と呼ばれています。"この GitHub の記事" 詳細については、を参照してください。Tegraf が起動して動作している限り、ユーザはこのエラーメッセージを無視できます。
Kubernetes で、 Telegraf ポッドが次のエラーを報告しています。 "Error in processing mountstats info: failed to open mountstats file: /hostfs /proc/1/mountstats 、 error: open /hostfs /proc/1/mountstats ： permission denied"	SELinuxを有効にして強制すると、TelegrafポッドがKubernetesノードの/proc/1/mountstatsファイルにアクセスできなくなる可能性があります。この制限を克服するには、agentconfigurationを編集し、runPrivileged設定を有効にします。詳細については、を参照して"OpenShift の手順"ください。

問題	次の操作を実行します
Kubernetes で、 Telegraf ReplicaSet ポッドから次のエラーが報告されています。 [プラグインの inputs.prometheus] エラー： Could not load keypair /etc/Kubernetes /pki/ etcd/server.crt ： /etc/Kubernetes /pki/ etcd/server.key ： open /etc/Kubernetes /pki/ etcd/server.key ：特定のディレクトリまたは crt ファイルをロードできませんでした	Telegraf ReplicaSet ポッドは、マスターまたは etcd 用に指定されたノード上で実行することを目的としています。これらのノードのいずれかで ReplicaSet ポッドが実行されていない場合は、これらのエラーが発生します。マスター / etcd ノードに汚染があるかどうかを確認します。その場合は、 Telegraf ReplicaSet 、テレグラム af-RS に必要な忍容を追加します。たとえば、 ReplicaSet…kubectl を編集して RS テレグラムを編集し、仕様に適切な公差を追加します。次に、 ReplicaSet ポッドを再起動します。
PSP/PSA環境があります。これはモニタリングオペレータに影響しますか？	KubernetesクラスタがPod Security Policy（PSP）またはPod Security Admission（PSA）を使用して実行されている場合は、最新のKubernetes Monitoring Operatorにアップグレードする必要があります。PSP/PSAをサポートしている現在のオペレータにアップグレードするには、次の手順に従います。アンインストール以前の監視演算子: kubectl delete agent-monitoring-cr-n NetApp kubectl delete ns NetApp -monitoring kubectl delete crd agents.monitoring.com kubectl delete clusterrole agent-manager-role agent-proxy-role agent-metrics-reader kubectl delete clusterrolebinding agent-manager-manager-rolebinding agent-manager-manager-rolebinding NetApp NetAppインストールモニタリングオペレータの最新バージョン。
Operatorを展開しようとして問題が発生しましたが、PSP/PSAを使用しています。	1.次のコマンドを使用してエージェントを編集します。kubectl -n <name-space> edit agent 2.「security-policy enabled」を「false」に設定します。これにより、PodセキュリティポリシーとPodセキュリティアドミッションが無効になり、オペレータが展開できるようになります。次のコマンドを使用して確認します。kubectl get psp（should show Pod Security Policy removed）kubectl get all -n <namespace>
grep -i psp（should show that nothing is found）	「ImagePullBackoff」エラーが発生しました
これらのエラーは、カスタムまたはプライベートのDockerリポジトリがあり、Kubernetes Monitoring Operatorを適切に認識するように設定していない場合に表示されることがあります。詳細はこちらカスタム/プライベートリポジトリの構成について	監視オペレータの配置に問題 を使用していますが、現在のドキュメントでは解決できません。

問題	次の操作を実行します

次のコマンドの出力をキャプチャまたはメモし、テクニカルサポートチームに連絡します。
 	
`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`
	Operator名前空間のNet-Observer（ワークロードマップ）ポッドがCrashLoopBackOffにある

これらのポッドは、Network ObservabilityのWorkload Mapデータコレクタに対応しています。以下を試してみてください:•いずれかのポッドのログをチェックして、カーネルの最小バージョンを確認してください。例：--- {"ci-tenant-id"："your-tenant-id"、"collector-cluster"："your-k8s-cluster-name"、"environment"："prod"、"level"："error"、"msg"："検証に失敗しました。理由:カーネルバージョン3.10.0が最小カーネルバージョン4.18.0よりも小さい、"time":"2022-11-09T08:23:08Z"}---•Net-observerポッドを使用するには、Linuxカーネルバージョンが4.18.0以上である必要があります。「uname -r」コマンドを使用してカーネルのバージョンを確認し、4.18.0以上であることを確認します	PodはOperatorネームスペース（デフォルト：netapp-monitoring）で実行されているが、Queriesのワークロードマップまたは Kubernetes指標のデータがUIに表示されない
K8Sクラスタのノードの時間設定を確認します。監査およびデータレポートを正確に作成するには、Network Time Protocol（NTP；ネットワークタイムプロトコル）またはSimple Network Time Protocol（SNTP；簡易ネットワークタイムプロトコル）を使用してAgentマシンの時刻を同期することを強く推奨します。	Operator名前空間の一部のnet-observerポッドがPending状態です
net-observerはデーモンセットであり、Kubernetesクラスタの各ノードでポッドを実行します。•保留状態のポッドをメモし、CPUまたはメモリのリソース問題 が発生しているかどうかを確認します。必要なメモリとCPUがノードにあることを確認します。	Kubernetes監視演算子をインストールした直後にログに次のようなメッセージが表示されます。[inputs.prometheus]プラグインエラー:\ http://kube-state-metricsへのHTTPリクエストの作成エラー。<namespace>.svc.cluster.local：8080/metrics：get\ http://kube-state-metrics <namespace>.svc.cluster.local：808080/metrics：dial tcp：lookup kube-state-metrics .<namespace>.svc.svc.cluster.local tc.local

問題	次の操作を実行します
このメッセージが表示されるのは、通常、_KSM_PODが起動する前に、新しいオペレータがインストールされ、_テレ グラム-RS_PODが稼働している場合のみです。これらのメッセージは、すべてのポッドが実行されると停止します。	クラスタに存在するKubernetes CronJobsについて収集された指標が表示されません。
Kubernetesのバージョンを確認します（例：`kubectl version`）。v1.20.x以下の場合、これは想定される制限です。Kubernetes Monitoring Operatorで導入されたkube-state-metricsリリースでは、v1.cronjobのみがサポートされます。Kubernetes 1.20.x以前では、cronjobリソースはv1beta.cronjobにあります。その結果、kube-state-metricsはcronjobリソースを見つけることができません。	オペレータのインストール後、telegraf-DSポッドがCrashLoopBackOffに入り、PODログに「su ：Authentication failure」と表示されます。
_AgentConfiguration_のtelegrafセクションを編集し、set_dockerMetricCollectionEnabled_をfalseに設定します。詳細については、オペレータのを参照して"設定オプション"ください。…spec:…telegraf:… -name：docker run-mode： - DaemonSet 置換： -key ：docker_unix_sock_placeholder 値：unix ：//run/docker.sock……	Telegrafログに次のようなエラーメッセージが繰り返し表示されます。[agent]出力への書き込み中にエラーが発生しました。http：Post "\https ：//<tenant_url>/rest/v1/lake/ingest/influxdb"：context deadline exceeded (Client. ヘッダー待機中にタイムアウトを超過しました）
_AgentConfiguration_およびincrease_outputTimeout_のtelegrafセクションを10秒に編集します。詳細については、オペレータのを参照して"設定オプション"ください。	一部のイベントログの_involvedobject_dataが見つかりません。
上記の手順を実行していることを確認して"権限"ください。	2つの監視オペレータポッド（netapp-ci-monitoring-operator-pod <pod>とmonitoring-operator-pod）が実行されているのはなぜ<pod>ですか？
2023年10月12日付けで、Data Infrastructure Insightsは、ユーザへのサービス向上のためにオペレータをリファクタリングしました。これらの変更を完全に採用するには古いオペレータを削除します。、とが必要です。新しいものを取り付ける	Kubernetesイベントが予期せずData Infrastructure Insightsに報告されなくなりました。
event-exporterポッドの名前を取得します。 `kubectl -n netapp-monitoring get pods`	grep event-exporter

問題	次の操作を実行します
awk '{print $1}'	sed 's/event-exporter./event-exporter/'` 「netapp-ci-event-exporter」または「event-exporter」のいずれかにする必要があります。次に、監視エージェントを編集し `kubectl -n netapp-monitoring edit agent`、前の手順で見つけた適切なイベントエクスポータポッド名を反映するようにlog_fileの値を設定します。具体的には、log_file は「/var/log/containers/netapp-ci-event-exporter.log」または「/var/log/containers/event-exporter 。**log**」のいずれかに設定する必要があります。 **….** **fluent-bit:** **…** **- name: event-exporter-ci** **substitutions:** **- key: LOG_FILE** **values:** **- /var/log/containers/netapp-ci-event-exporter**.log … …. あるいは、1つはまた、再インストールエージェントをすることができますアンインストール。
リソースが不足しているため、Kubernetes Monitoring Operatorによってデプロイされたポッドがクラッシュしています。	CPUやメモリの制限を必要に応じて増やすには、Kubernetes Monitoring Operatorを参照して"設定オプション"ください。
イメージがないか無効な設定が原因で、netapp-ci-kube-state-metricsポッドが起動しないか準備完了状態になりました。これでStatefulSetが停止し、設定の変更がnetapp-ci-kube-state-metricsポッドに適用されなくなりました。	StatefulSetはステートに"切断"あります。設定の問題を修正したら、netapp-ci-kube-state-metricsポッドをバウンスします。
NetApp-ci-kube-state-metricsポッドがKubernetes Operatorのアップグレード実行後に起動せず、ErrImagePullがスローされる（イメージをプルできない）。	ポッドを手動でリセットしてみてください。
Kubernetesクラスタの[Log Analysis]で、「Event discarded as being older then maxEventAgeSeconds」というメッセージが確認されています。	Operator_agentconfiguration_を変更し、*event-exporter-maxEventAgeSeconds*（60秒）、*event-exporter-kubeQPS*（100）、および_event-exporter-kubeBurst_（500）を増やします。これらの設定オプションの詳細については、ページを参照して"設定オプション"ください。

問題	次の操作を実行します
Telegrafが警告するか、ロック可能なメモリが不足しているためにクラッシュします。	基盤となるオペレーティングシステム/ノードでTelegrafのロック可能メモリの制限を増やしてみてください。制限値を増やすことができない場合は'NKMOエージェントの構成を変更して'_unprotected_to_true_に設定しますこれにより、Telegrafはロックされたメモリページを予約しないように指示します。復号化されたシークレットがディスクにスワップアウトされる可能性があるため、セキュリティリスクが発生する可能性がありますが、ロックされたメモリを予約できない環境では実行できます。_unprotected_configurationオプションの詳細については、ページを参照してください"設定オプション"。
Telegrafから次のような警告メッセージが表示されます。[inputs.diskio]「vdc」のディスク名を収集できません：/dev/vdcの読み取り中にエラーが発生しました：該当するファイルまたはディレクトリがありません_	Kubernetes Monitoring Operatorの場合、これらの警告メッセージは問題なく無視してかまいません。 または、AgentConfigurationでtelegrafセクションを編集し、_runDsPrivileged_をtrueに設定します。詳細については、を参照して"オペレータの設定オプション"ください。

問題	次の操作を実行します

Fluent-bitポッドが次のエラーで失敗しています。[2024/10/16 14:16:23][error][/src/fluent-bit/plugins/in tail/tail_fs_inotify.c:360 errno=24]開いているファイルが多すぎます[2024/10/16 14:16:23][error] failed initialize initialization failed.	クラスタの_fsnotify_settingsを変更してみます。
	
	`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>`
	
	Fluent-bitを再起動します。
	
	注：これらの設定をノードの再起動後も維持するには、_/etc/sysctl.conf_に次の行を追加する必要があります。
	
	`fs.inotify.max_user_instances=<something larger than current setting>`
	`fs.inotify.max_user_watches=<something larger than current setting>`

詳細については、のページまたはを"Data Collector サポートマトリックス"参照して"サポート"ください。

Kubernetes監視オペレータの設定オプション

"Kubernetes監視オペレータ"構成はカスタマイズできます。

次の表に、_AgentConfiguration_ファイルに使用できるオプションを示します。

コンポーネント	オプション	製品説明
エージェント		オペレータがインストールできるすべてのコンポーネントに共通の設定オプション。これらは「グローバル」オプションと見なすことができます。
	dockerrepo	Data Infrastructure Insights Dockerリポジトリと比較して、お客様のプライベートDockerリポジトリからイメージを取得するためのdockerRepoオーバーライド。デフォルトはData Infrastructure Insights Dockerリポジトリ
	dockerImagePullSecretの略	オプション:顧客のプライベートリポジトリのシークレット
	クラスタ名	すべてのお客様のクラスタ間でクラスタを一意に識別するフリーテキストフィールド。これは、Data Infrastructure Insightsのテナント全体で一意である必要があります。デフォルトでは、UIで[Cluster Name]フィールドに入力します
	プロキシ形式:プロキシ:サーバー:ポート:ユーザー名:パスワード:noProxy:isTelegrafProxyEnabled:isAuProxyEnabled:isFluentbitProxyEnabled:isCollectorProxyEnabled:	プロキシを設定する場合はオプションです。これは通常、顧客の法人代理店です。
テレグラフ		オペレータのTelegrafインストールをカスタマイズできる設定オプション
	collectionInterval	指標収集間隔（秒）（最大=60秒）
	dsCpuLimit	Telegraf DSのCPU制限
	dsMemLimit	Telegraf DSのメモリ制限
	dsCpuRequest	Telegraf DSのCPU要求
	dsMemRequest	Telegraf DSのメモリ要求
	rsCpuLimit	Telegraf RSのCPU制限
	rsMemLimit	Telegraf RSのメモリ制限
	rsCpuRequest	Telegraf RSのCPU要求
	rsMemRequest	テレグラフRSのメモリ要求
	runPrivileged	特権モードでtelegrafデーモンセットの_telegraf-mountstats-poller_containerを実行します。KubernetesノードでSELinuxが有効になっている場合は、このパラメータをtrueに設定します。
	runDsPrivileged	特権モードでtelegraf DaemonSetのtelegrafコンテナを実行するには、runDsPrivilegedをtrueに設定します。
	バッチサイズ	を参照し "Telegraf設定ドキュメント"
	BufferLimit	を参照し "Telegraf設定ドキュメント"

コンポーネント	オプション	製品説明
	RoundIntervalの略	を参照し "Telegraf設定ドキュメント"
	collectionJitter	を参照し "Telegraf設定ドキュメント"
	精度	を参照し "Telegraf設定ドキュメント"
	flushInterval（フラッシュ間隔）	を参照し "Telegraf設定ドキュメント"
	FlushJitter（フラッシュジッタ	を参照し "Telegraf設定ドキュメント"
	outputTimeout	を参照し "Telegraf設定ドキュメント"
	dsTolerations	Telegraf-DS追加の許容値。
	rsTolerations	Telegraf-RS追加許容値。
	skipProcessorsAfterAggregators	を参照し "Telegraf設定ドキュメント"
	保護なし	これを見てください"既知のTelegraf問題" 。setting_unprotected_は、Kubernetes Monitoring Operatorにフラグを指定してTelegrafを実行するように指示し `--unprotected`ます。
kube-state-metricsの略		Operatorのkube状態メトリックのインストールをカスタマイズできる設定オプション
	cpuLimit	kube-state-metricsデプロイメントのCPU制限
	memLimit	kube-state-metrics展開のメモリ制限
	cpuRequest	kube state metrics deploymentのCPU要求
	MemRequestの略	KUBE状態メトリクス展開のためのMEM要求
	リソース	キャプチャするリソースのカンマ区切りリスト。例：cronjobs、daemonsets、deployment、ingresses、jobs、namespace、nodes、persistentvolumes、pods、ReplicaSets、resourcequotas、services、statefulsets
	許容範囲	kube-state-metrics追加の許容値。
	ラベル	kube-state-metricsでキャプチャするリソースのカンマ区切りリスト例：cronjobs=**[]**、**daemonsets=[]**、deployments=**[]**、**ingresses=[]**、jobs=**[]**、**namespaces=[]**、nodes=**[]**、**persistentvolumes=[]**、pods=**[]**、**replicaresets=[]**、**[]**、**[]**、[*]
ログ		オペレータのログ収集とインストールをカスタマイズできる設定オプション
	readFromHead	true / false。fluentビットがheadからログを読み取る必要があります
	タイムアウト	タイムアウト（秒）
	DNSMode（DNSMode）	TCP / UDP、DNSのモード

コンポーネント	オプション	製品説明
	Fluent-bit-tolerationsの略	FLUENT-BIT-DSの追加許容値。
	event-exporter-tolerationsの略	イベントエクスポータの追加許容値。
	event-exporter-maxEventAgeSeconds	イベントエクスポータの最大イベント経過時間。を参照し https://github.com/jkroepke/resmoio-kubernetes-event-exporter
ワークロードマップ		作業負荷マップの収集とオペレータのインストールをカスタマイズできる設定オプション。
	cpuLimit	ネットオブザーバーDSのCPU制限
	memLimit	ネットオブザーバDSのメモリ制限
	cpuRequest	ネットオブザーバーDSのCPU要求
	MemRequestの略	ネットオブザーバーDSのMEM要求
	metricAggregationInterval	メトリック集約間隔（秒単位）
	bpfPollIntervalの略	BPFポーリング間隔（秒単位）
	enableDNSLookup	trueまたはfalse、DNSルックアップを有効にします
	L4 -公差	NET-OBSERVER-L4-DS追加許容値。
	runPrivileged	true/false - KubernetesノードでSELinuxが有効になっている場合は、runPrivilegedをtrueに設定します。
変更管理		Kubernetes Change Management and Analysisの構成オプション
	cpuLimit	change-observer-watch-rsのCPU制限
	memLimit	change-observer-watch-rsのメモリ制限
	cpuRequest	change-observer-watch-rsのCPU要求
	MemRequestの略	change-observer-watch-rsのMEM要求
	failureDeclarationIntervalMins	ワークロードの導入に失敗した場合に障害が発生したとマークされる間隔（分）
	deployAggrIntervalSeconds	ワークロード導入を実行中のイベントが送信される頻度
	nonWorkloadAggrIntervalSeconds	ワークロード以外の導入環境を組み合わせて送信する頻度
	termsToRedact	値が編集される環境名およびデータマップで使用される一連の正規表現例：「pwd」、「password」、「token」、「apiKey」、「api-key」、「jwt」
	AdditionalKindsToWatch	コレクターが監視するデフォルトの種類のセットから、監視する追加の種類のコンマ区切りリスト
	kindsToIgnoreFromWatch	コレクタが監視するデフォルトの種類のセットから、監視対象から無視する種類のコンマ区切りのリスト

コンポーネント	オプション	製品説明
	logRecordAggrIntervalSeconds	コレクタからCIにログレコードを送信する頻度
	ウォッチトレランス	change-observer-watch-ds追加の許容値。省略された単一行形式のみ。例：「｛key：taint1、operator：exists、effect：NoSchedule｝、｛key：taint2、operator：exists、effect：NoExecute｝」

サンプルの**AgentConfiguration**ファイル

以下は、Sample_AgentConfiguration_ファイルです。

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

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

    # # Proxy settings. The proxy that the operator should use to send
metrics to Data Infrastructure Insights.
    # # Please see documentation here: https://docs.netapp.com/us-
en/cloudinsights/task_config_telegraf_agent_k8s.html#configuring-proxy-
support
    # proxy:
    #   server:
    #   port:
    #   noproxy:
    #   username:
    #   password:
    #   isTelegrafProxyEnabled:
```

```
    #   isFluentbitProxyEnabled:
    #   isCollectorsProxyEnabled:

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

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

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

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

```
(precision).
    # precision: '0s'
    # # Time telegraf will wait between writing outputs (flush_interval).
Max=collectionInterval
    # flushInterval: '60s'
    # # Each output waits a random amount of time between the scheduled
write time and that time + flush_jitter before writing outputs
(flush_jitter).
    # flushJitter: '0s'
    # # Timeout for writing to outputs (timeout).
    # outputTimeout: '5s'

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

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

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

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


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

```
    # unprotected: 'false'

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

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

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

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

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

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

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

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

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

    # # Comma-separated list of resources to enable.
    # # See resources in https://github.com/kubernetes/kube-state-
```

```
metrics/blob/main/docs/cli-arguments.md
    # resources:
'cronjobs,daemonsets,deployments,ingresses,jobs,namespaces,nodes,persisten
tvolumeclaims,persistentvolumes,pods,replicasets,resourcequotas,services,s
tatefulsets'

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

```
otal,kube_pod_status_scheduled_time,kube_pod_status_unschedulable,kube_pod
_spec_volumes_persistentvolumeclaims_readonly,kube_pod_container_resource_
requests_cpu_cores,kube_pod_container_resource_requests_memory_bytes,kube_
pod_container_resource_requests_storage_bytes,kube_pod_container_resource_
requests_ephemeral_storage_bytes,kube_pod_container_resource_limits_cpu_co
res,kube_pod_container_resource_limits_memory_bytes,kube_pod_container_res
ource_limits_storage_bytes,kube_pod_container_resource_limits_ephemeral_st
orage_bytes,kube_pod_init_container_resource_limits_cpu_cores,kube_pod_ini
t_container_resource_limits_memory_bytes,kube_pod_init_container_resource_
limits_storage_bytes,kube_pod_init_container_resource_limits_ephemeral_sto
rage_bytes,kube_pod_init_container_resource_requests_cpu_cores,kube_pod_in
it_container_resource_requests_memory_bytes,kube_pod_init_container_resour
ce_requests_storage_bytes,kube_pod_init_container_resource_requests_epheme
ral_storage_bytes,kube_replicaset_status_replicas,kube_replicaset_status_r
eady_replicas,kube_replicaset_status_observed_generation,kube_replicaset_s
pec_replicas,kube_replicaset_metadata_generation,kube_replicaset_labels,ku
be_replicaset_created,kube_replicaset_owner,kube_resourcequota,kube_resour
cequota_created,kube_service_info,kube_service_labels,kube_service_created
,kube_service_spec_type,kube_statefulset_status_replicas,kube_statefulset_
status_replicas_current,kube_statefulset_status_replicas_ready,kube_statef
ulset_status_replicas_updated,kube_statefulset_status_observed_generation,
kube_statefulset_replicas,kube_statefulset_metadata_generation,kube_statef
ulset_created,kube_statefulset_labels,kube_statefulset_status_current_revi
sion,kube_statefulset_status_update_revision,kube_node_status_capacity,kub
e_node_status_allocatable,kube_node_status_condition,kube_pod_container_re
source_requests,kube_pod_container_resource_limits,kube_pod_init_container
_resource_limits,kube_pod_init_container_resource_requests'

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

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

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

```
    # shards: '2'

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

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

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

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

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

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

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

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

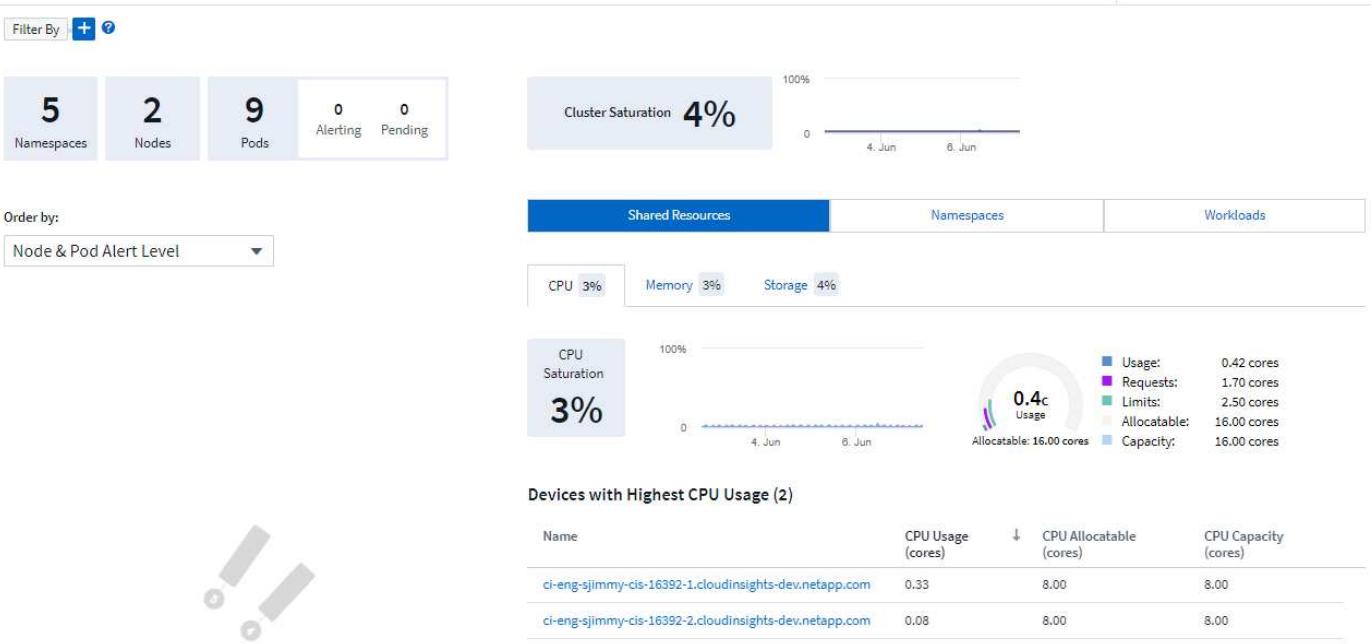
```
    # event-exporter-kubeQPS: 20
    # event-exporter-kubeBurst: 100

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

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

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

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

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

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

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

  # change-management:
    # # change-observer-watch-rs CPU/Mem limits and requests.
```

```
    # # See https://kubernetes.io/docs/concepts/configuration/manage-
resources-containers/
    # cpuLimit: '1'
    # memLimit: '1Gi'
    # cpuRequest: '500m'
    # memRequest: '500Mi'

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

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

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

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

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

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

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

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

    # # change-observer-watch-ds additional tolerations. Use the following
```

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

Kubernetes クラスタの詳細ページ

Kubernetesクラスタの詳細ページには、Kubernetesクラスタの概要が表示されます。



ネームスペース、ノード、およびポッドの数

ページ上部のカウントには、クラスタ内のネームスペース、ノード、ポッドの総数、および現在アラート中および保留中のPopdの数が表示されます。

共有リソースと飽和

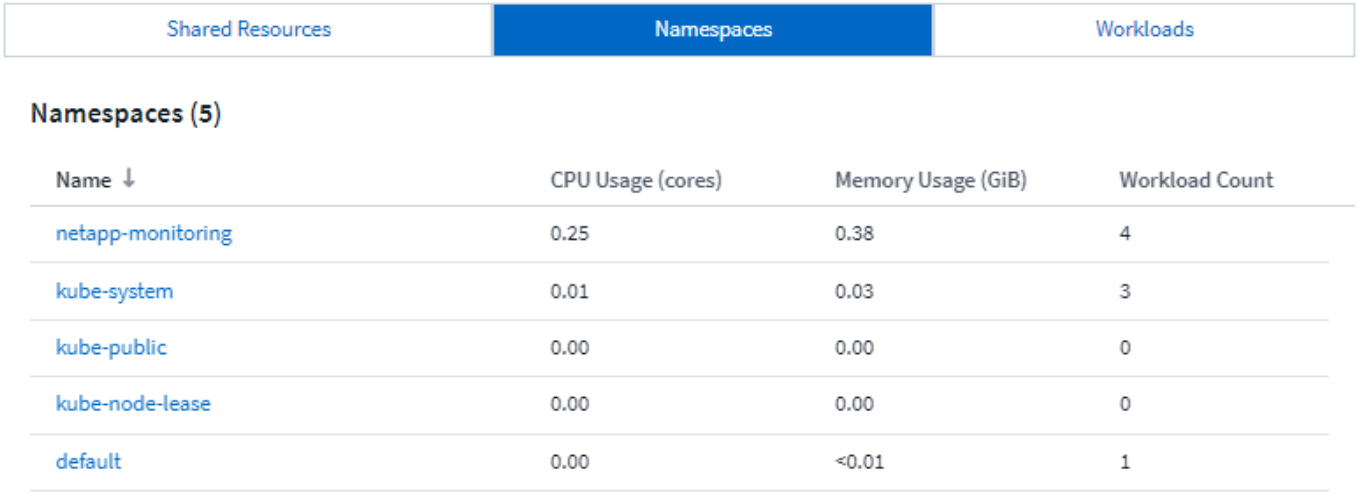
詳細ページの右上には、クラスタが現在の割合で飽和状態になっているかどうかのグラフと、その期間の最新の傾向が表示されます。クラスタの飽和は、各時点でのCPU、メモリ、またはストレージの飽和状態の最大値です。

その下には、CPU、メモリ、ストレージのタブがデフォルトで*共有リソース*の使用状況として表示されます。各タブには、時間の経過に伴う飽和度と傾向が表示され、使用状況の詳細も表示されます。ストレージの場合、この値はバックエンドとファイルシステムの飽和度の大きい方で、それぞれ独立して計算されます。

使用率が最も高いデバイスが下部の表に表示されます。リンクをクリックすると、これらのデバイスを確認できます。

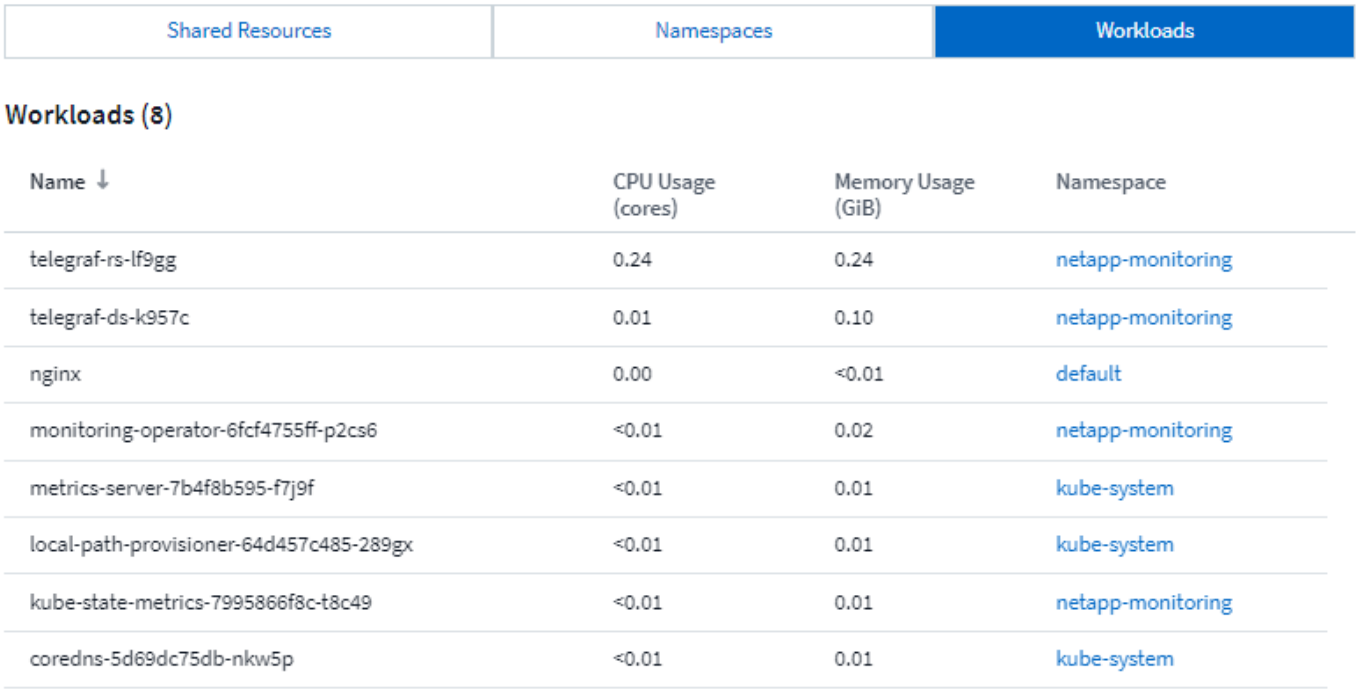
ネームスペース

[ネームスペース]タブには、Kubernetes環境内のすべてのネームスペースのリストが表示されます。このタブには、CPUとメモリの使用状況、および各ネームスペース内のワークロードの数が表示されます。名前のリンクをクリックして、各ネームスペースを確認します。



ワークロード

同様に、各ネームスペースのワークロードのリストがワークロードタブに表示され、CPUとメモリの使用量も表示されます。ネームスペースリンクをクリックすると、ドリルでそれぞれが実行されます。



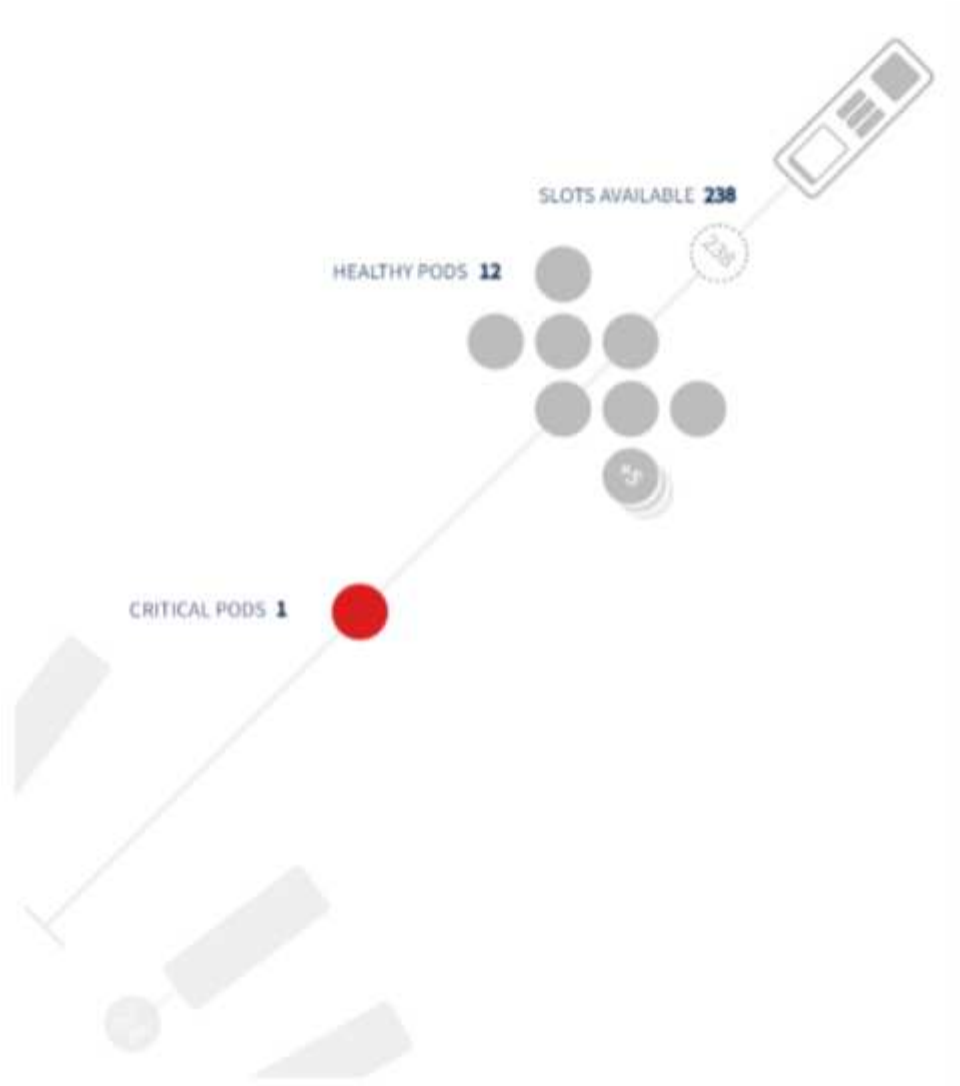
クラスタ「ホイール」

クラスタの「ホイール」セクションでは、ノードとポッドの健全性を一目で確認できます。詳細については、ドリルで確認できます。クラスタのノード数がページのこの領域に表示される数を超えている場合は、使用可能なボタンを使用してホイールを回すことができます。

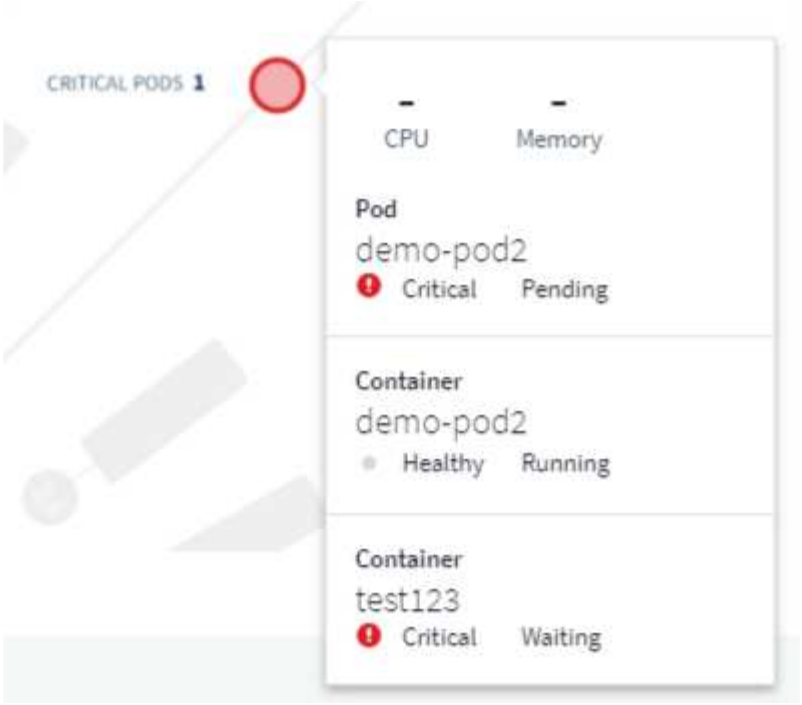
アラートのポッドまたはノードは赤で表示されます。「警告」の領域はオレンジで表示されます。スケジュールされていないポッド（未接続）は、クラスタ「 Wheel 」の下部コーナーに表示されます。

ポッド（円）またはノード（バー）にカーソルを合わせると、ノードのビューが拡張されます。

そのビューでポッドまたはノードをクリックすると、展開されたノードビューが拡大表示されます。



ここから、エレメントにカーソルを合わせると、そのエレメントの詳細を表示できます。たとえば、この例で重要なポッドにカーソルを合わせると、そのポッドに関する詳細が表示されます。



ファイルシステム、メモリ、および CPU の情報を表示するには、 Node 要素にカーソルを合わせます。

ゲージに関する注意

メモリと CPU のゲージには、 *allocatable capacity* と _total の容量 _ に関連して _Used_in が表示されるので、 3 色が表示されます。

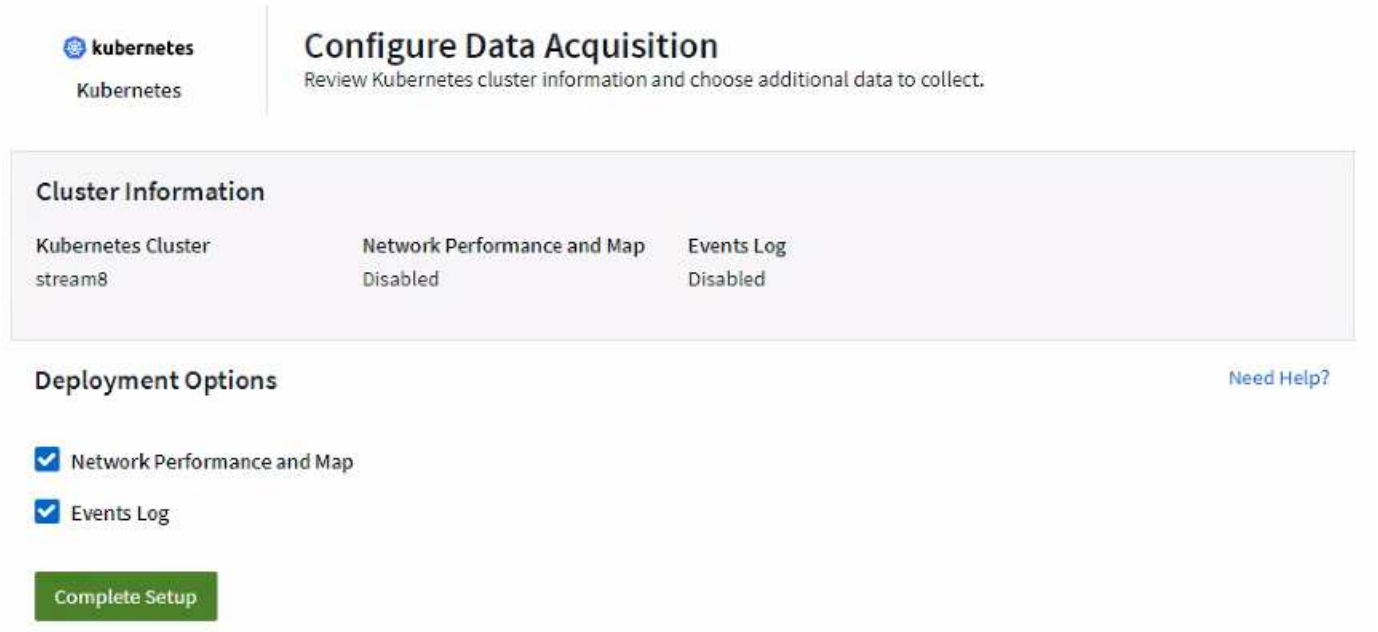
Kubernetes Network Performance Monitoring and Mapの略

KubernetesのNetwork Performance Monitoring and Map機能は、サービス（ワークロードとも呼ばれます）間の依存関係をマッピングすることでトラブルシューティングを簡易化し、ネットワークパフォーマンスのレイテンシや異常をリアルタイムで可視化して、ユーザに影響を与える前にパフォーマンスの問題を特定します。この機能は、Kubernetesのトラフィックフローを分析、監査することで全体的なコストを削減するのに役立ちます。

主な機能:•ワークロードマップはKubernetesワークロードの依存関係とフローを示し、ネットワークとパフォーマンスの問題を強調します。•Kubernetesポッド、ワークロード、ノード間のネットワークトラフィックを監視し、トラフィックとレイテンシの問題の原因を特定します。•入力、出力、リージョン間、ゾーン間のネットワークトラフィックを分析することで、全体的なコストを削減します。

前提条件

Kubernetes Network Performance Monitoring and Mapを使用する前に、このオプションを有効にするようにを設定しておく必要があります"NetApp Kubernetes Monitoring Operator"。オペレータの配備中に、[ネットワークパフォーマンスとマップ]チェックボックスをオンにして有効にします。このオプションを有効にするには、Kubernetesランディングページに移動して[Modify Deployment]を選択します。



モニタ

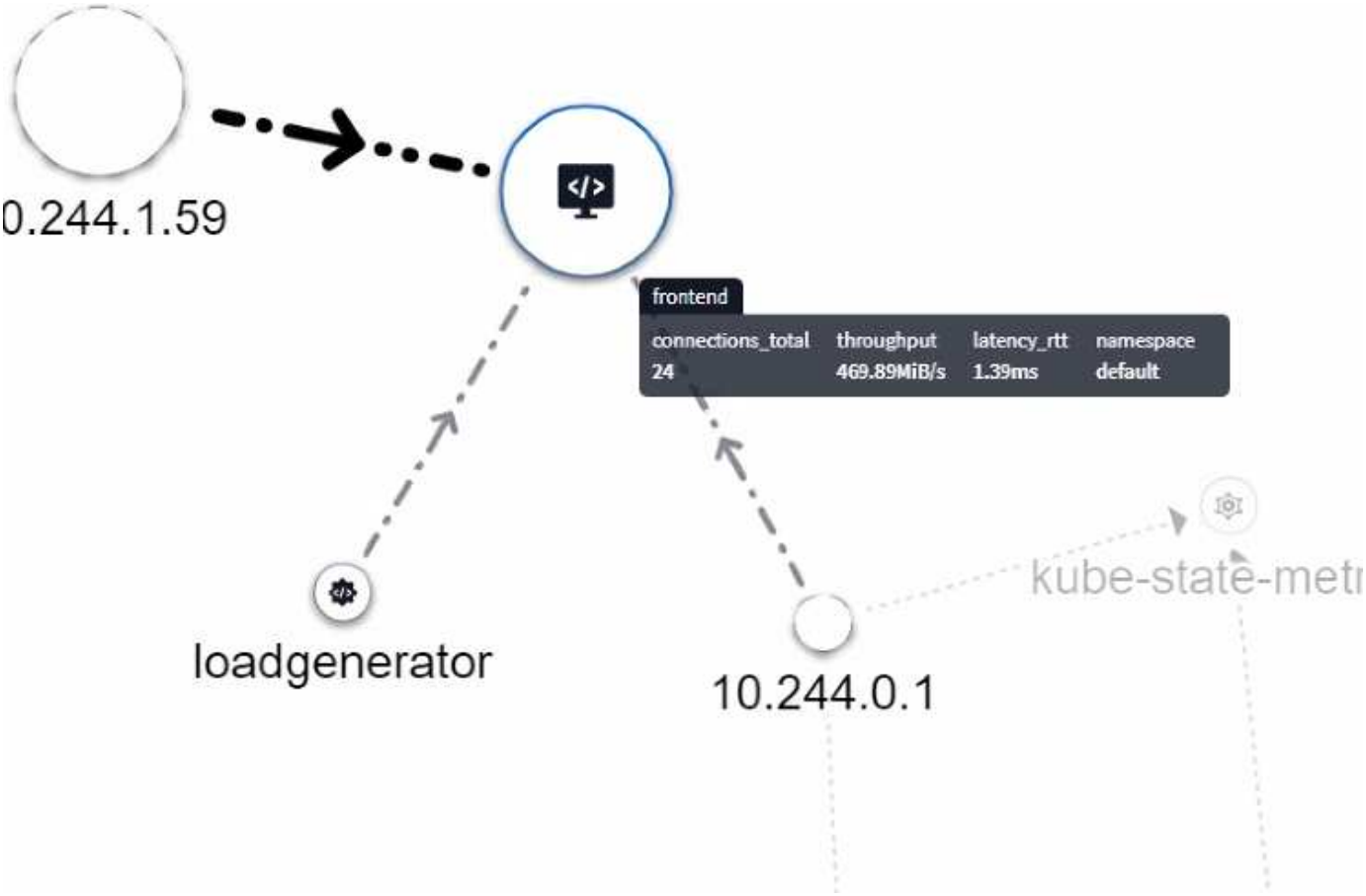
ワークロードマップは、を使用し"カンシ"て情報を取得します。Data Infrastructure Insightsには、多数のデフォルトKubernetesモニタが用意されています（デフォルトでは、これらのモニタは_Paused_になっている場合があります）。必要なモニタを_Resume_(つまり有効化)することも、ワークロードマップでも使用されるKubernetesオブジェクト用のカスタムモニタを作成することもできます。

Data Infrastructure Insightsの指標アラートは、以下のいずれかのオブジェクトタイプに対して作成できます。データがデフォルトのオブジェクトタイプでグループ化されていることを確認します。

- kubernetes.workload
- kubernetes.daemonset
- kubernetes.deployment
- kubernetes.cronjob
- kutability.job
- kubernetes.replicaset
- kubernetes.statefulset
- Kubernetesポッド
- kubernetes.network_traffic_l4

地図

マップには、サービス/ワークロードとそれらの相互関係が表示されます。矢印は交通の方向を示しています。ワークロードにカーソルを合わせると、そのワークロードの概要情報が表示されます（次の例を参照）。



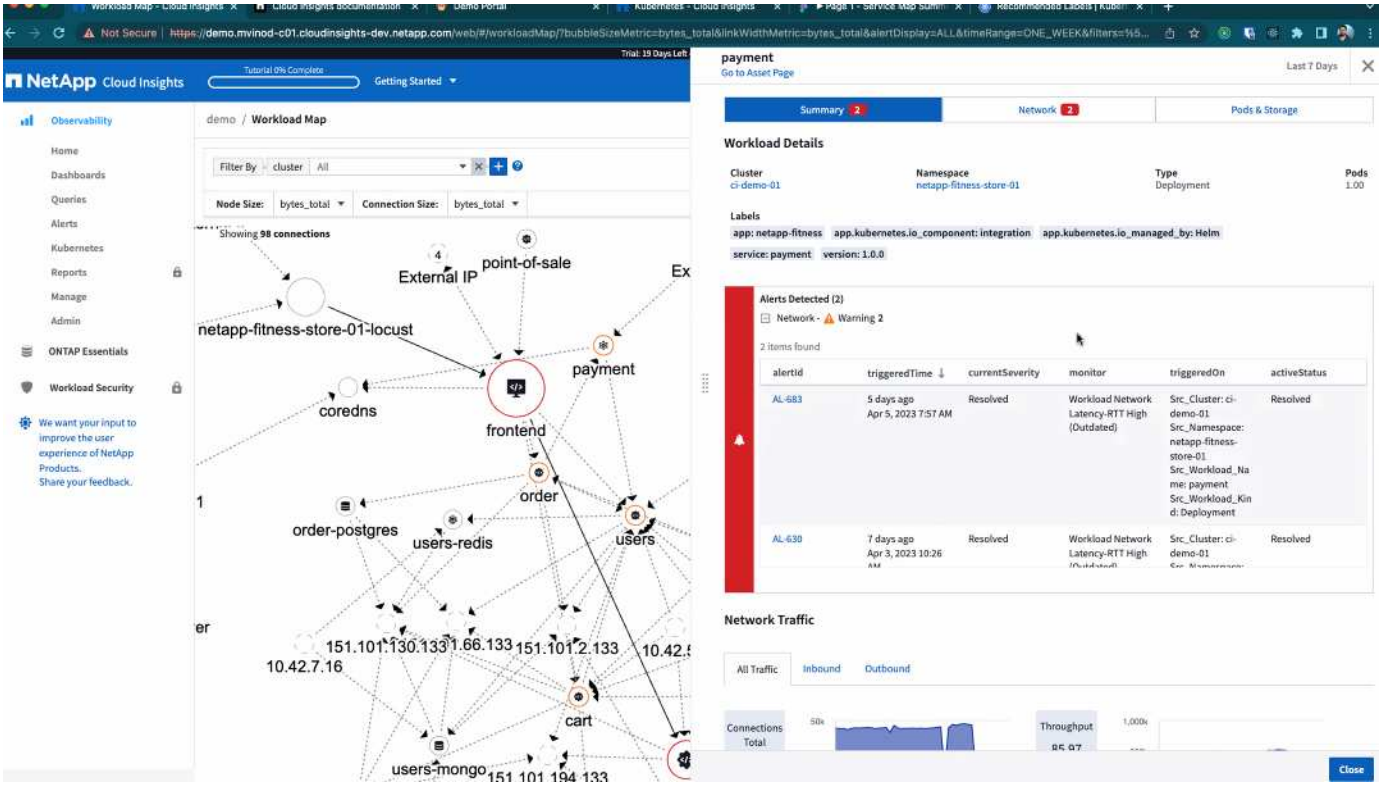
円の中のアイコンは、さまざまなサービスタイプを表します。アイコンは、基になるオブジェクトにがある場合にのみ表示されラベルます。

各円のサイズはノードサイズを示します。これらのサイズは相対的なものであることに注意してください。ブラウザのズームレベルまたは画面サイズは、実際の円のサイズに影響を与える可能性があります。同様に、トラフィックラインスタイルでは、接続サイズが一目でわかるようになっています。太字の実線は交通量が多く、点線は交通量が少ないです。

円の中の数字は、サービスによって現在処理されている外部接続の数です。

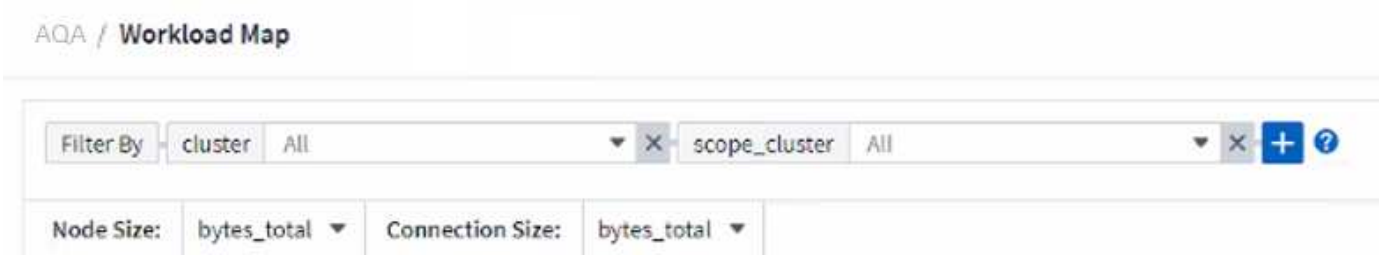
ワークロードの詳細とアラート

色の円は、ワークロードに関する警告レベルまたは重大レベルのアラートを示します。円にカーソルを合わせると問題 の概要が表示されます。円をクリックすると、より詳細なスライドアウトパネルが開きます。

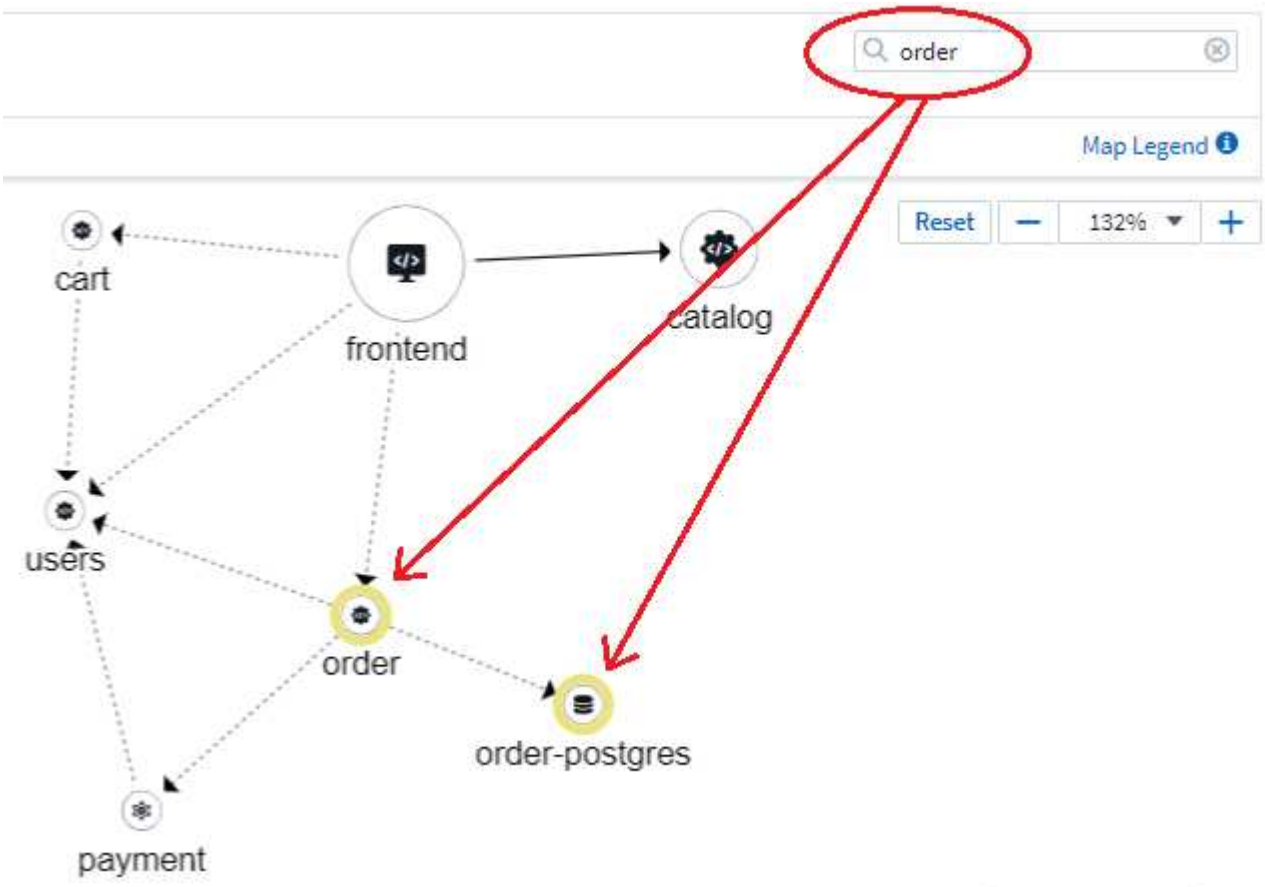


検索とフィルタ

Data Infrastructure Insightsの他の機能と同様に、必要なオブジェクトやワークロードの属性に絞り込むフィルタを簡単に設定できます。



同様に、_Find_フィールドに文字列を入力すると、一致するワークロードがハイライトされます。



ワークロードラベル

表示されるワークロードのタイプ（円のアイコンなど）をマップで識別するには、ワークロードラベルが必要です。ラベルは次のように導出されます。

- 一般的な用語で実行されているサービス/アプリケーションの名前
- ソースがポッドの場合：
  - ラベルはポッドのワークロードラベルから取得されます
  - ワークロードの想定されるラベル：app.kubernetes.io/component
  - ラベル名参照： https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels/
  - 推奨ラベル：
    - フロントエンド

- バックエンド
- データベース
- キャッシュ
- キュー
- カフカ

- ソースがKubernetesクラスタの外部にある場合は、次の手順を実行します。
  - Data Infrastructure Insightsは、DNS解決名を解析してサービスタイプを抽出しようとします。

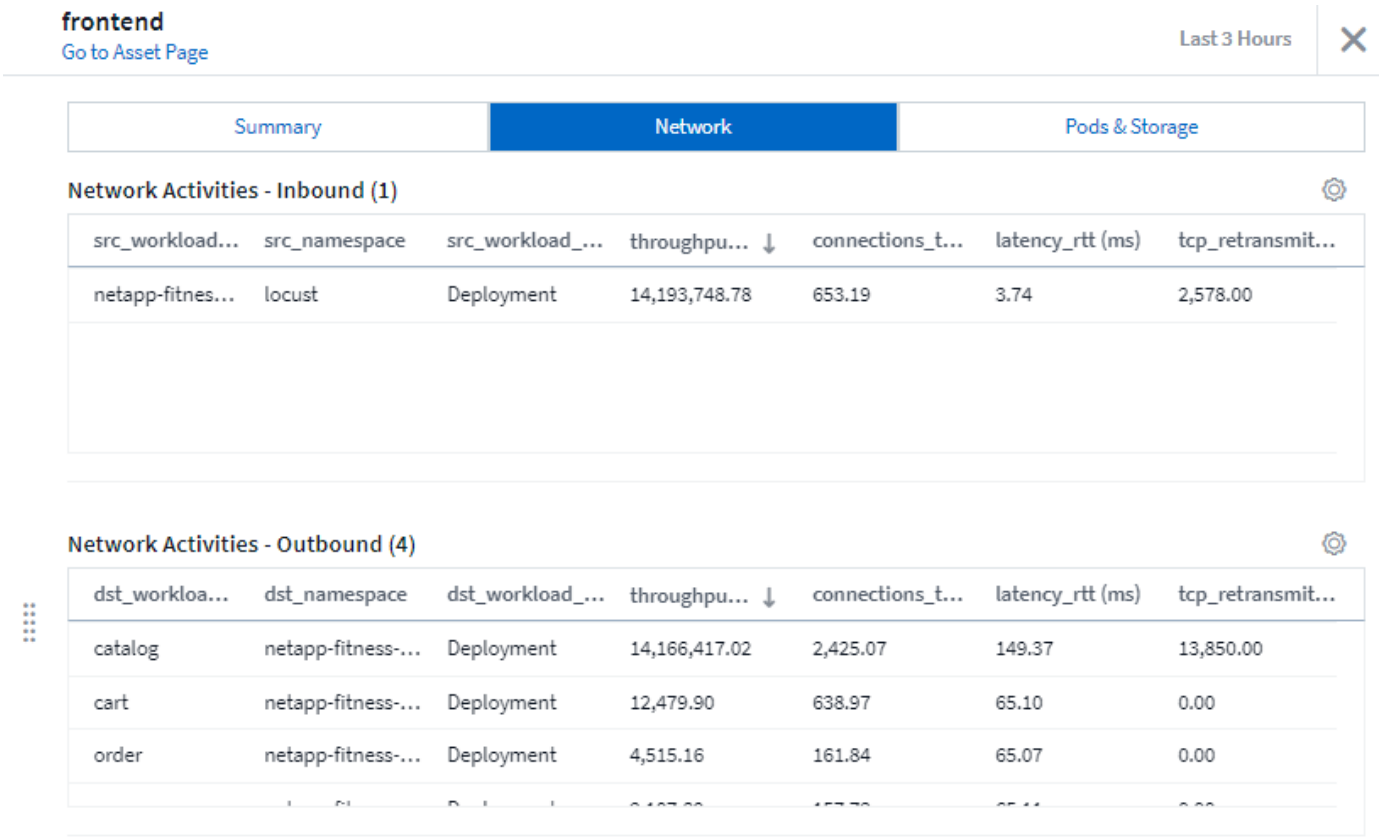
    たとえば、DNS解決名が_s3.eu-north-1.amazonaws.comの場合、解決された名前はサービスタイプとしてget_s3_に解析されます。

深海に潜る

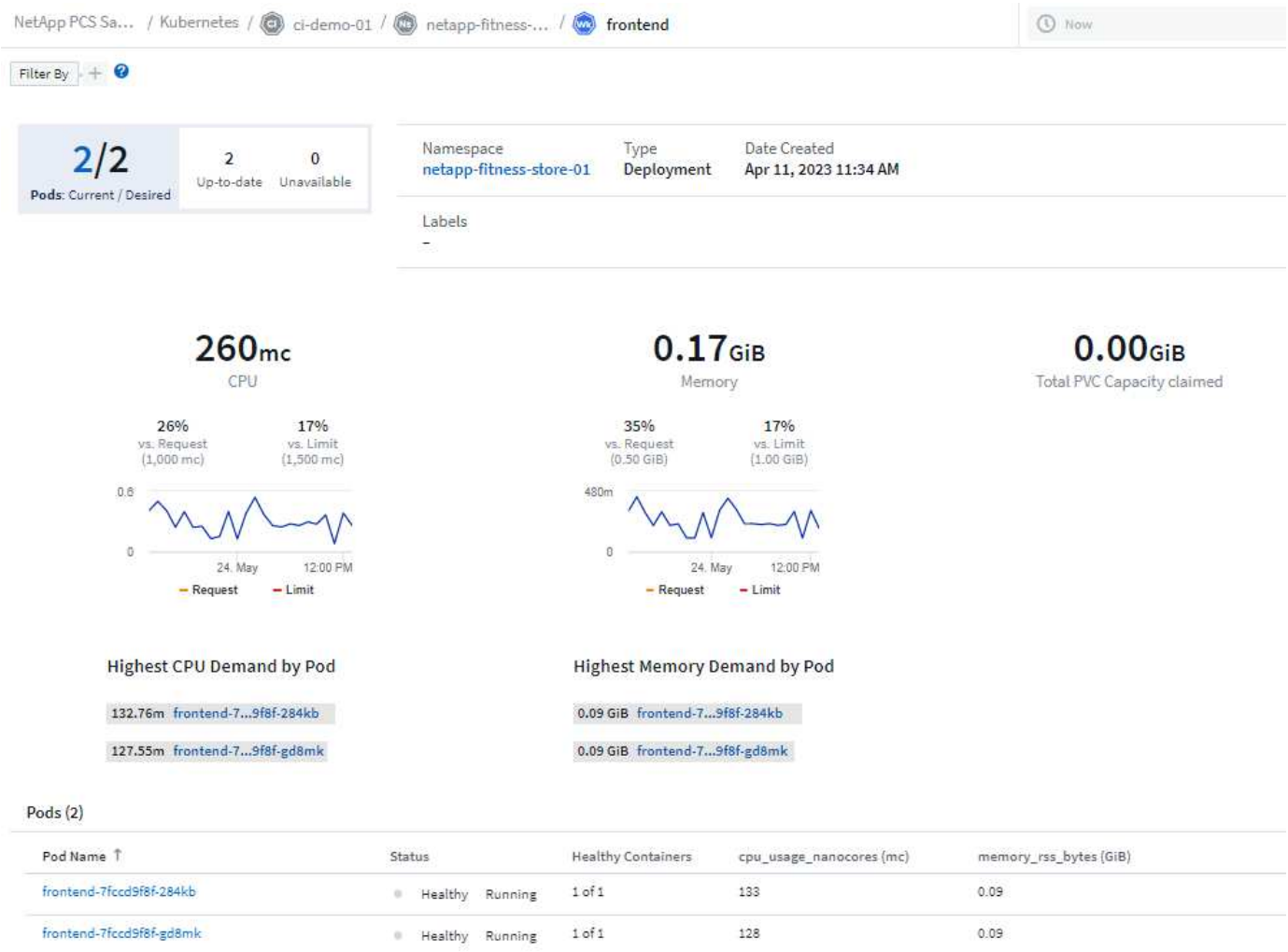
ワークロードを右クリックすると、さらに詳しく調べるための追加のオプションが表示されます。たとえば、ここからズームインして、そのワークロードの接続を表示できます。

を右クリックすると、ワークロードの接続が表示されます"]

または、詳細スライドアウトパネルを開いて、*Summary*、*Network*、または_Pod & Storage_タブを直接表示することもできます。



最後に、[*Go to Asset Page*]を選択すると、ワークロードの詳細なアセットランディングページが開きます。



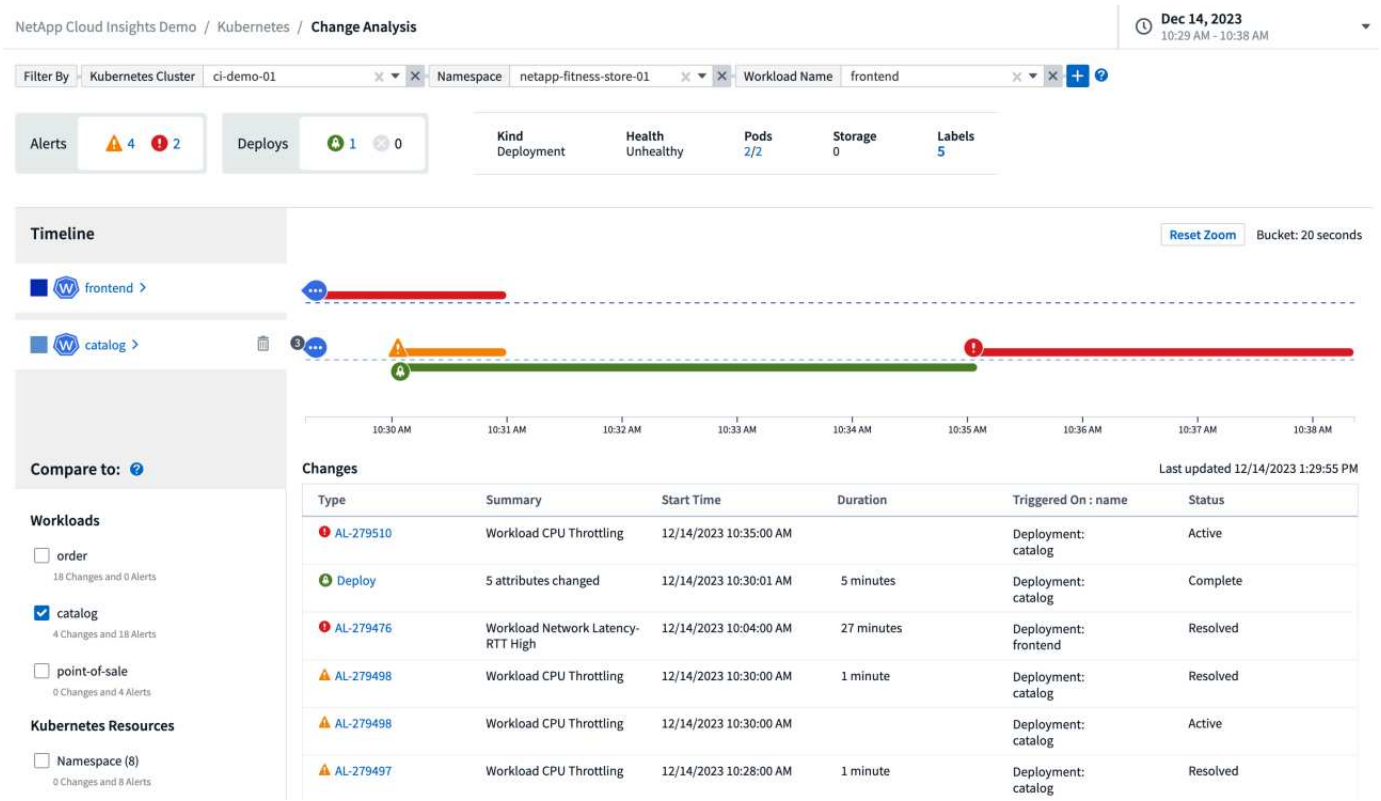
Kubernetesの変更分析

Kubernetes Change Analyticsを使用すると、Kubernetes環境に対する最近の変更をオールインワンビューで確認できます。アラートと導入ステータスをすぐに確認できます。変更分析を使用すると、導入と設定の変更をすべて追跡し、Kubernetesのサービス、インフラ、クラスタの健全性とパフォーマンスに関連付けることができます。

変更分析はどのように役立ちますか？

- マルチテナントKubernetes環境では、設定ミスが原因でシステムが停止する可能性があります。Change Analyticsでは、ワークロードの健全性と構成の変更を1つのペインで表示して関連付けることができます。これは、動的なKubernetes環境のトラブルシューティングに役立ちます。

Kubernetes Change Analyticsを表示するには、* Kubernetes > Change Analysis *に移動します。



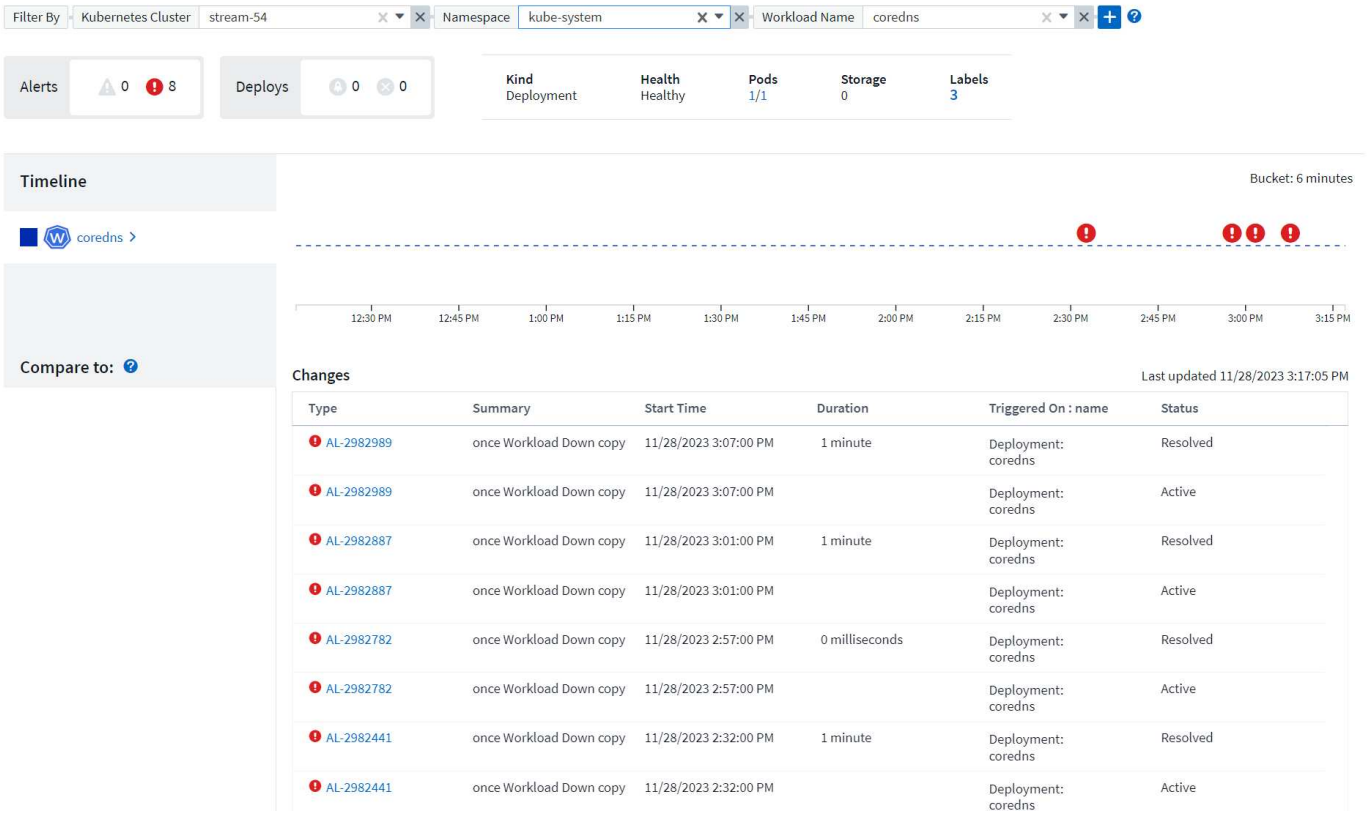
です。

このページは、現在選択されているData Infrastructure Insightsの期間に基づいて自動的に更新されます。時間範囲が小さいほど、画面の更新頻度が高くなります。

フィルタリング

Data Infrastructure Insightsのすべての機能と同様に、変更リストは直感的にフィルタリングできます。ページ上部で、Kubernetesクラスタ、ネームスペース、ワークロードの値を入力または選択したり、[+]ボタンを選択して独自のフィルタを追加したりできます。

特定のクラスタ、ネームスペース、ワークロードにフィルタを適用して（設定した他のフィルタと一緒に）、そのクラスタ上のそのネームスペース内のそのワークロードに対する導入とアラートのタイムラインが表示されます。さらに拡大するには、グラフをクリックしてドラッグし、より具体的な時間範囲にフォーカスします。

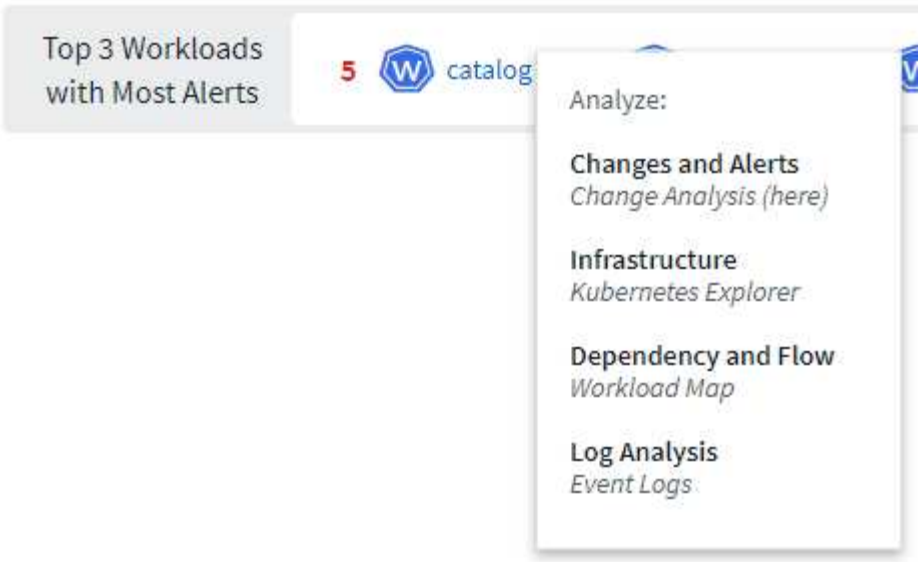


クイックステータス

フィルタリングエリアの下には、いくつかの高レベルインジケータがあります。左側にはアラートの数（WarningおよびCritical）が表示されます。この数には、_Active_alertsと_Resolved_alertsが含まれます。Only_Active_alertsを表示するには、「Status」のフィルタを設定し、「Active」を選択します。

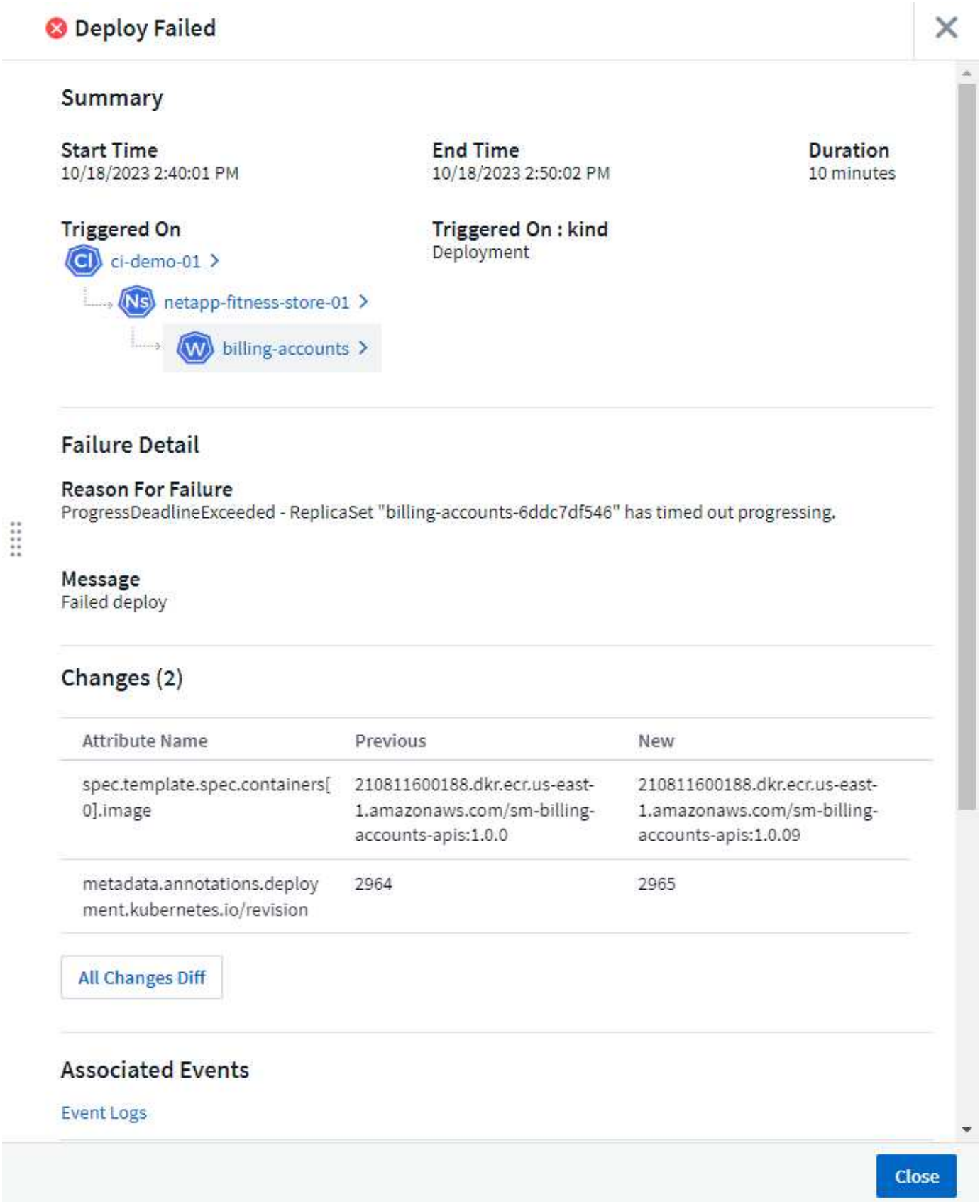
導入ステータスもここに表示されます。繰り返しになりますが、デフォルトでは、*Started*、*complete*、および_Failed_deploymentsの数が表示されます。Failed_Deploymentsのみを表示するには、[Status]のフィルタを設定し、[Failed]を選択します。

次にアラートが最も多い上位3つのワークロードが表示されます。各ワークロードの横にある赤の数字は、そのワークロードに関連するアラートの数を示します。ワークロードのリンクをクリックして、インフラ（Kubernetes Explorer）、依存関係（ワークロードマップ）、またはログ分析（イベントログ）を確認します。



詳細パネル

リストで変更を選択すると、変更を詳細に説明するパネルが開きます。たとえば、失敗したDeployを選択すると、Deployの概要、開始時刻と終了時刻、期間、および導入がトリガーされた場所、およびそれらのリソースを確認するためのリンクが表示されます。また、失敗の理由、関連する変更、関連するイベントも表示されます。



同様にアラートを選択すると、アラートをトリガーしたモニタやアラートのタイムラインを示すグラフなど、アラートの詳細が表示されます。

著作権に関する情報

商標に関する情報