Kubernetes

Data Infrastructure Insights

NetApp
February 10, 2026

目次

Kubernetes

Kubernetes クラスターの概要

Data Infrastructure Insights Kubernetes Explorer は、Kubernetes クラスターの全体的な健全性と使用状況を表示し、調査領域を簡単にドリルダウンできる強力なツールです。

ダッシュボード **> Kubernetes Explorer** をクリックすると、Kubernetes クラスターのリスト ページが開きます。この概要ページには、テナント上の Kubernetes クラスターの表が含まれています。

[Kubernetes リストページ]

クラスターリスト

クラスター リストには、テナント上の各クラスターの次の情報が表示されます。

- クラスター 名前。クラスター名をクリックすると、"詳細ページ"そのクラスターに対して。
- *彩度*のパーセンテージ。全体的な飽和度は、CPU、メモリ、またはストレージの飽和度のうち最も高い値です。
- クラスター内の ノード の数。この番号をクリックすると、ノード リスト ページが開きます。
- クラスター内の **Pod** の数。この番号をクリックすると、Pod リスト ページが開きます。
- クラスター内の*名前空間*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。
- クラスター内の ワークロード の数。この番号をクリックすると、ワークロード リスト ページが開きます。

フィルターの改良

フィルタリングを行う場合、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 を作成したり、「なし」オプションを選択してフィールド内の null 値をフィルター処理したりすることもできます。

[K8Sエクスプローラーでワイルドカードを使ってフィルタリングする]

ワイルドカードまたは式 (例: NOT、AND、「なし」など) に基づくフィルターは、フィルター フィールドに濃い青色で表示されます。リストから直接選択した項目は水色で表示されます。

[ワイルドカードと選択された項目を表示するフィルター]

Kubernetes フィルターはコンテキストに依存します。つまり、たとえば特定のノード ページにいる場合、pod_name フィルターによってそのノードに関連するポッドのみがリストされます。さらに、特定の名前空間にフィルターを適用すると、pod_name フィルターによって、そのノード上およびその名前空間内のポッドのみがリストされます。

ワイルドカードと式の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、ブール値では機能しないことに注意してください。

NetApp Kubernetes Monitoring Operator をインストールまたはアップグレードする前に

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 モニタリング オペレーター"。

コンポーネント	要件
Kubernetes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 は、 Arm64 アーキテクチャで実行されているノードをサポートしていません。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、これらのプラットフォームで次の Kubernetes ラベルを検索する Kubernetes ノード セレクターを指定することにより、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 以上: Kubernetes.io/os = linux Rancher + cattle.io (オーケストレーション/Kubernetes プラットフォームとして): 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 PATH に追加します。
接続	kubectl cli は、ターゲットの K8s クラスターと通信するように構成されており、 Data Infrastructure Insights環境にインターネット接続できます。インストール中にプロキシを使用している場合は、"プロキシサポートの設定"オペレータインストールのセクション。正確な監査とデータ レポートを行うには、ネットワーク タイム プロトコル (NTP) または簡易ネットワーク タイム プロトコル (SNTP) を使用してエージェント 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、"OpenShiftの手順"これらの前提条件が満たされていることを確認することに加えて。
APIトークン	Operator を再デプロイする場合 (つまり、更新または置換する場合)、新しい API トークンを作成する必要はなく、以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な事項

もしあなたがプロキシ、持っているカスタムリポジトリ、または使用中オープンシフト、以下のセクションを

注意深くお読みください。

こちらもご覧ください権限。

プロキシサポートの設定

NetApp Kubernetes Monitoring Operator をインストールするために、テナント上でプロキシを使用できる場所は 2 つあります。これらは同じプロキシ システムである場合もあれば、別のプロキシ システムである場合もあります。

- インストール コード スニペットの実行中（「curl」を使用）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 Kubernetes クラスターが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 Operating Monitor をインストールするには、まずプロキシがData Infrastructure Insights環境との良好な通信を許可するように構成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ー/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については、Operator をインストールする前に、*http_proxy/https_proxy* 環境変数を設定します。一部のプロキシ環境では、*no_proxy environment* 変数も設定する必要がある場合があります。

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. 現在のユーザーの *https_proxy* および/または *http_proxy* 環境変数を設定します。
   a. セットアップするプロキシに認証 (ユーザー名/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 （ユーザー名/パスワード）
がある場合は、次のコマンドを実行します。

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

Kubernetes クラスターがData Infrastructure Insights環境と通信するために使用するプロキシについては、これらの手順をすべて読んだ後、 NetApp Kubernetes Monitoring Operator をインストールしてください。

NetApp Kubernetes Monitoring Operator をデプロイする前に、operator-config.yaml の AgentConfiguration の proxy セクションを構成します。

```
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**Docker**リポジトリの使用

デフォルトでは、 NetApp Kubernetes Monitoring Operator は、 Data Infrastructure Insightsリポジトリからコンテナ イメージをプルします。監視のターゲットとして Kubernetes クラスタが使用されており、そのクラスタがカスタムまたはプライベート 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 イメージをプルするように構成されている場合は、 NetApp Kubernetes Monitoring Operator に必要なコンテナへのアクセスを構成する必要があります。

NetApp Monitoring Operator インストール タイルから「イメージ プル スニペット」を実行します。このコマンドは、 Data Infrastructure Insightsリポジトリにログインし、オペレーターのすべてのイメージ依存関係をプルし、 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、提供され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ているかについては、以下を参照してください。

コアオペレーター機能とKubernetesモニタリング

- netapp 監視
- kube-rbac-プロキシ
- kube-state-metrics
- テレグラフ
- ディストロレスルートユーザー

イベントログ

- 流暢なビット
- kubernetes イベント エクスポーター

ネットワークパフォーマンスとマップ

- ci-net-オブザーバー

企業ポリシーに従って、オペレーターの 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リポジトリのものと一致していることを確認します。

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

```
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>
```

新しい docker リポジトリの場所を反映するように、operator-config.yaml の AgentConfiguration を編集します。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、*operator-config.yaml* の AgentConfiguration を編集して、*runPrivileged* 設定を有効に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

監視対象のクラスタに、ClusterRoleを持たないカスタムリソースが含まれている場合、"表示する集計"イベント ログでこれらのリソースを監視するには、オペレーターにこれらのリソースへのアクセスを手動で許可する必要があります。

1. インストール前に *operator-additional-permissions.yaml* を編集するか、インストール後にリソース *ClusterRole/<namespace>-additional-permissions* を編集します。
2. 動詞 ["get", "watch", "list"] を使用して、目的の apiGroup とリソースの新しいルールを作成します。参照\https://kubernetes.io/docs/reference/access-authn-authz/rbac/
3. 変更をクラスターに適用する

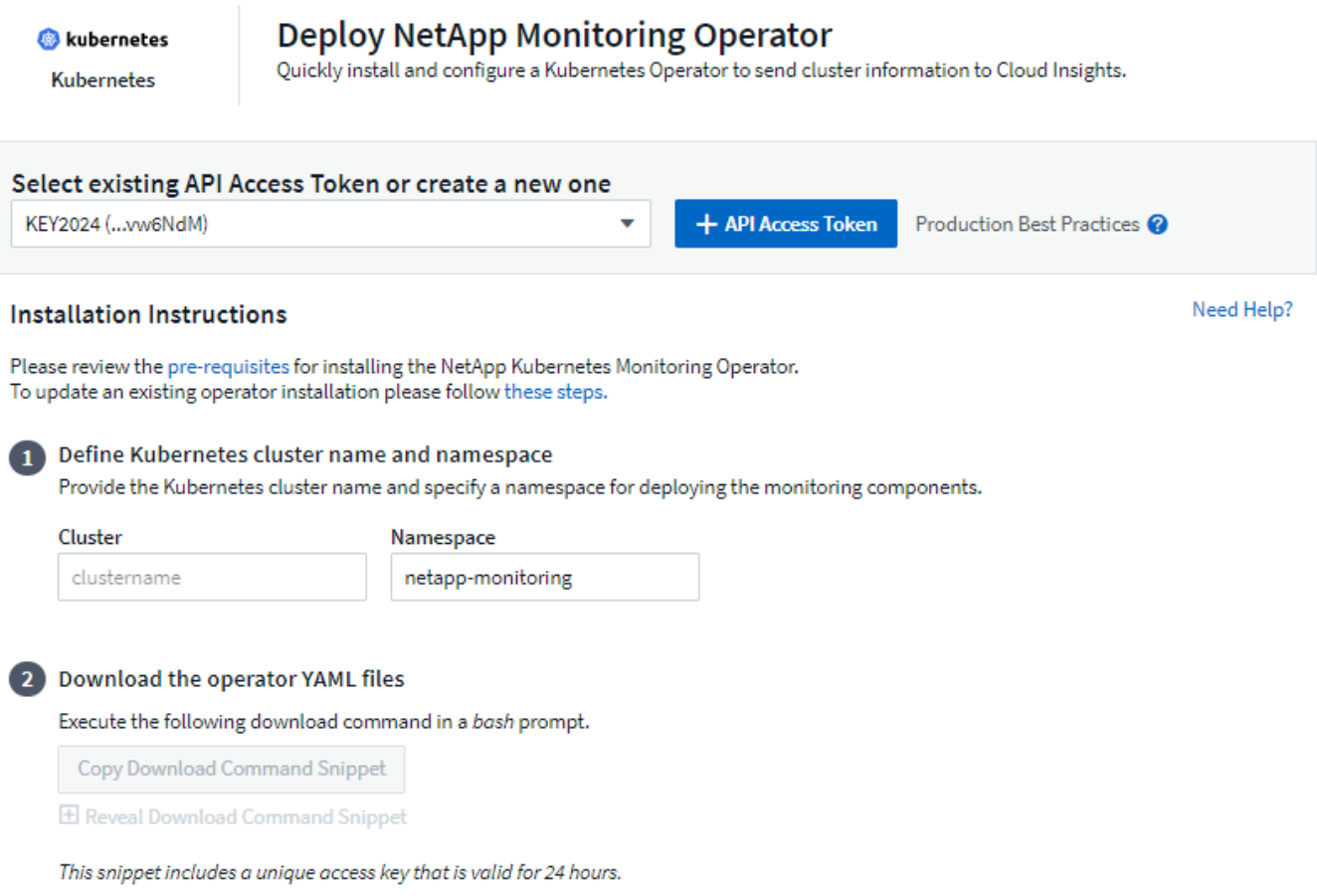
Kubernetes モニタリング オペレーターのインストールと構成

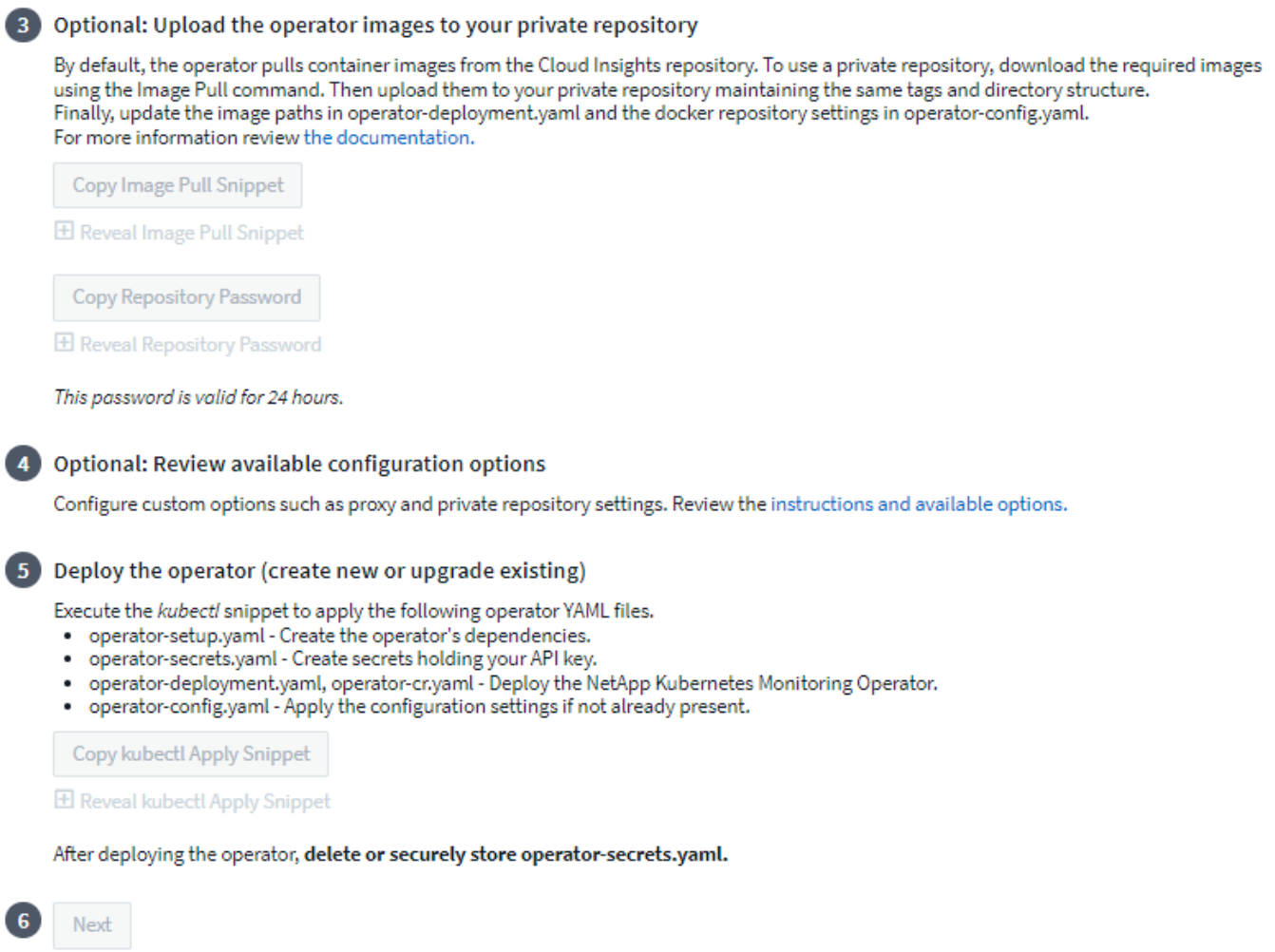
Data Infrastructure Insights は、 Kubernetes コレクション用の **Kubernetes Monitoring Operator** を提供します。新しいオペレーターをデプロイするには、**Kubernetes > Collectors > +Kubernetes Collector** に移動します。

Kubernetesモニタリングオペレーターをインストールする前に

参照"前提条件"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。

Kubernetes モニタリング オペレーターのインストール


kubernetes
Kubernetes
Deploy NetApp Monitoring Operator
Quickly install and configure a Kubernetes Operator to send cluster information to Cloud Insights.
Select existing API Access Token or create a new one
KEY2024 (...vw6NdM)
+ API Access Token
Production Best Practices
Installation Instructions
Need Help?
Please review the pre-requisites for installing the NetApp Kubernetes Monitoring Operator.
To update an existing operator installation please follow these steps.
1 Define Kubernetes cluster name and namespace
Provide the Kubernetes cluster name and specify a namespace for deploying the monitoring components.
Cluster
clustername
Namespace
netapp-monitoring
2 Download the operator YAML files
Execute the following download command in a bash prompt.
Copy Download Command Snippet
Reveal Download Command Snippet
This snippet includes a unique access key that is valid for 24 hours.



Kubernetes に **Kubernetes Monitoring Operator** エージェントをインストールする手順:

1. 一意のクラスター名と名前空間を入力します。もしあなたがアップグレード以前の Kubernetes Operator からの場合は、同じクラスター名と名前空間を使用します。
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 *bash* ウィンドウに貼り付けて実行します。 Operator インストール ファイルがダウンロードされます。スニペットには一意のキーがあり、24 時間有効であることに注意してください。
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、*bash* シェルに貼り付けて実行します。イメージをプルしたら、それをプライベート リポジトリにコピーします。必ず同じタグとフォルダー構造を維持してください。 *operator-deployment.yaml* 内のパスと、*operator-config.yaml* 内の 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 リポジトリ設定などの利用可能な構成オプションを確認します。詳細については、"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、ダウンロードして実行して、Operator をデプロイします。
7. インストールは自動的に進行します。完了したら、[次へ] ボタンをクリックします。
8. インストールが完了したら、[次へ] ボタンをクリックします。 *operator-secrets.yaml* ファイルも必ず削除するか、安全に保存してください。

カスタムリポジトリをお持ちの場合は、以下をお読みください。カスタム/プライベートDockerリポジトリを

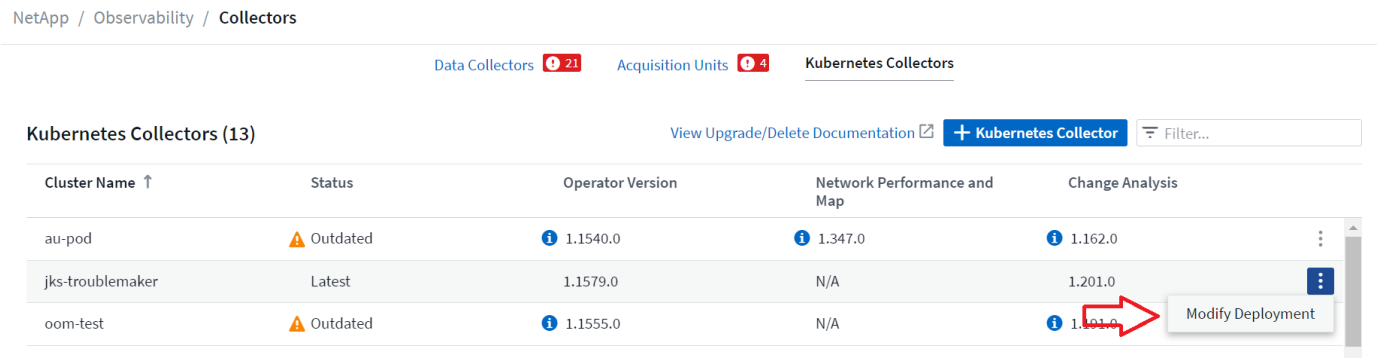
使用する 。

Kubernetes 監視コンポーネント

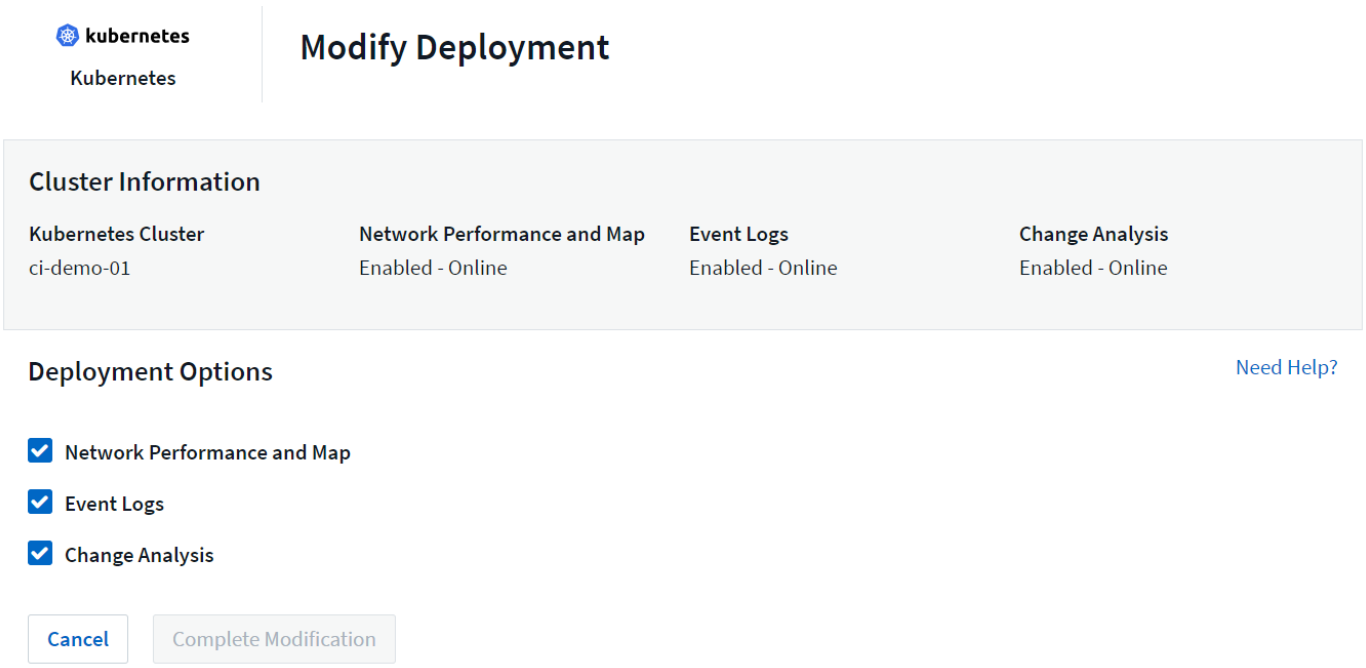
Data Infrastructure Insights Kubernetes モニタリングは、次の 4 つのモニタリング コンポーネントで構成されています。

- クラスターメトリック
- 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 コレクターに対してデフォルトで有効になっています。特定のコレクターに対して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、画面の右側にあるコレクターの「3 つのドット」メニューから *Modify Deployment* を選択して無効にすることができます。



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ー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モニタリングオペレーターへのアップグレード

DII プッシュボタンアップグレード

DII Kubernetes Collectors ページから Kubernetes Monitoring Operator をアップグレードできます。アップグレードするクラスターの横にあるメニューをクリックし、「アップグレード」を選択します。オペレーターはイメージ署名を検証し、現在のインストールのスナップショットを実行して、アップグレードを実行します。数分以内に、オペレーターのステータスが「アップグレード進行中」から「最新」へと進行していくのが確認できます。エラーが発生した場合は、詳細を表示するにはエラー ステータスを選択し、以下のプッシュ ボタン アップグレードのトラブルシューティング表を参照してください。

プライベートリポジトリによるプッシュボタンアップグレード

オペレーターがプライベート リポジトリを使用するように構成されている場合は、オペレーターの実行に必要なすべてのイメージとその署名がリポジトリで使用可能であることを確認してください。アップグレードプロセス中にイメージが不足しているためにエラーが発生した場合は、イメージをリポジトリに追加して、アップグレードを再試行してください。リポジトリにイメージ署名をアップロードするには、次のようにcosignツールを使用してください。3 オプション: 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 > イメージプルスニペットで指定されたすべてのイメージの署名をアップロードしてください。

```
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>
```

以前実行していたバージョンにロールバックする

プッシュボタン アップグレード機能を使用してアップグレードし、アップグレード後 7 日以内に現在のバージョンのオペレーターに問題が発生した場合は、アップグレード プロセス中に作成されたスナップショットを使用して、以前実行していたバージョンにダウングレードできます。ロールバックするクラスターの横にあるメニューをクリックし、[ロールバック] を選択します。

手動アップグレード

既存の Operator に *AgentConfiguration* が存在するかどうかを判断します（名前空間がデフォルトの *netapp-monitoring* でない場合は、適切な名前空間に置き換えます）：

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

AgentConfiguration が存在する場合：

- インストール既存のオペレータよりも最新のオペレータを優先します。
  - 必ず最新のコンテナイメージを取得するカスタム リポジトリを使用している場合。

AgentConfiguration が存在しない場合：

- Data Infrastructure Insightsによって認識されるクラスター名をメモします (名前空間がデフォルトの netapp-monitoring でない場合は、適切な名前空間に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（名前空間がデフォルトの netapp-
monitoring でない場合は、適切な名前空間に置き換えます）。
```

```
 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
>>既存のオペレータ。
* <<installing-the-kubernetes-monitoring-operator,インストール
>>最新のオペレーター。
```

- 同じクラスター名を使用します。
- 最新の Operator YAML ファイルをダウンロードした後、デプロイする前に、*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。
- 必ず最新のコンテナイメージを取得するカスタム リポジトリを使用している場合。

Kubernetes モニタリング オペレーターの停止と起動

Kubernetes モニタリング オペレーターを停止するには:

```
 kubectl -n netapp-monitoring scale deploy monitoring-operator
--replicas=0
Kubernetes モニタリング オペレーターを起動するには：
```

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

アンインストール

Kubernetesモニタリングオペレーターを削除するには

Kubernetes モニタリング オペレーターのデフォルトの名前空間は「netapp-monitoring」であることに注意してください。独自の名前空間を設定している場合は、これらのコマンドと後続のすべてのコマンドおよびファイルでその名前空間を置き換えます。

監視オペレーターの新しいバージョンは、次のコマンドで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>
```

監視オペレーターが専用のネームスペースにデプロイされている場合は、ネームスペースを削除します。

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

注：
最初のコマンドで「リソースが見つかりません」と返された場合は、次の手順に従って、監視オペレーターの古いバージョンをアンインストールしてください。

以下の各コマンドを順番に実行します。現在のインストールによっては、これらのコマンドの一部が「オブジェクトが見つかりません」というメッセージを返す場合があります。これらのメッセージは無視しても問題ありません。

```
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以前に作成されている場合:

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

Kube-state-metricsについて

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

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

オペレーターの設定/カスタマイズ

これらのセクションには、オペレーター構成のカスタマイズ、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 の操作に関する情報が含まれています。

設定オプション

最も頻繁に変更される設定は、*AgentConfiguration* カスタム リソースで構成できます。オペレーターをデプロイする前に、*operator-config.yaml* ファイル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の例が含まれています。リストを見る"利用可能な設定"オペレーターの最新バージ

ョン。

オペレーターをデプロイした後、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
```

デプロイされたオペレーターのバージョンが _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
```

「サーバーからのエラー (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ーをアップグレードする必要があります。

プロキシサポートの設定

Kubernetes モニタリング オペレーターをインストールするために、テナント上でプロキシを使用できる場所は 2 つあります。これらは同じプロキシ システムである場合もあれば、別のプロキシ システムである場合もあります。

- インストール コード スニペットの実行中（「curl」を使用）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 Kubernetes クラスターが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、まずプロキシがData Infrastructure Insights環境との良好な通信を許可するように構成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ー/VM からData Infrastructure Insightsにアクセスできる場合は、プロキシは適切に構成されている可能性があります。

Kubernetes オペレーティング モニター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* 環境変数を設定します。一部のプロキシ環境では、*no_proxy environment* 変数も設定する必要がある場合があります。

変数を設定するには、Kubernetes モニタリング オペレーター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. 現在のユーザーの *https_proxy* および/または *http_proxy* 環境変数を設定します。
   a. セットアップするプロキシに認証 (ユーザー名/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 （ユーザー名/パスワード）がある場合は、次のコマンドを実行します。

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

Kubernetes クラスターがData Infrastructure Insights環境と通信するために使用するプロキシについては、これらの手順をすべて読んだ後、Kubernetes Monitoring Operator をインストールしてください。

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**Docker**リポジトリの使用

デフォルトでは、Kubernetes Monitoring Operator はData Infrastructure Insightsリポジトリからコンテナ イメージをプルします。監視のターゲットとして Kubernetes クラスターが使用されており、そのクラスターがカスタムまたはプライベート Docker リポジトリまたはコンテナー レジストリからのみコンテナー イメージをプルするように構成されている場合は、Kubernetes 監視オペレーターに必要なコンテナーへのアクセスを構成する必要があります。

NetApp Monitoring Operator インストール タイルから「イメージ プル スニペット」を実行します。このコマンドは、 Data Infrastructure Insightsリポジトリにログインし、オペレーターのすべてのイメージ依存関係をプルし、 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、提供され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ているかについては、以下を参照してください。

コアオペレーター機能とKubernetesモニタリング

- netapp 監視
- ci-kube-rbac-プロキシ
- ci-ksm
- ci-telegraf
- ディストロレスルートユーザー

イベントログ

- ci-fluent-bit
- ci-kubernetes-イベントエクスポーター

ネットワークパフォーマンスとマップ

- ci-net-オブザーバー

企業ポリシーに従って、オペレーターの 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リポジトリのものと一致していることを確認します。

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

```
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>
```

新しい docker リポジトリの場所を反映するように、*operator-config.yaml* の *AgentConfiguration* を編集します。プライベートリポジトリ用の新しい 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
```

長期パスワード用の **API** アクセストークン

一部の環境（プロキシリポジトリなど）では、Data Infrastructure Insights docker リポジトリの長期パスワードが必要です。インストール時に UI で提供されるパスワードは 24 時間のみ有効です。それを使用する代わりに、API アクセストークンを docker リポジトリパスワードとして使用できます。このパスワードは、API アクセストークンが有効である限り有効です。この特定の目的のために新しい API アクセストークンを生成することも、既存のものを使用することもできます。

"こちらをお読みください"新しいAPIアクセストークンを作成する手順については、こちらをご覧ください。

ダウンロードした *operator-secrets.yaml* ファイルから既存の API アクセス トークンを抽出するには、ユーザーは以下を実行できます：

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

実行中のオペレーターインストールから既存のAPIアクセストークンを抽出するには、ユーザーは以下を実行できます：

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

OpenShift の手順

OpenShift 4.6 以降で実行している場合は、*operator-config.yaml* の *AgentConfiguration* を編集して *runPrivileged* 設定を有効に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。

寛容と汚点

netapp-ci-telegraf-ds、*netapp-ci-fluent-bit-ds*、および *netapp-ci-net-observer-l4-ds* DaemonSets は、すべてのノードでデータを正しく収集するために、クラスター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。オペレーターは、いくつかのよく知られた 汚染 を許容するように設定されています。ノードにカスタムテイントを設定して、ポッドがすべてのノードで実行されないようにしている場合は、それらのテイントに対して*許容*を作成できます。"_AgentConfiguration_内" 。クラスター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、オペレーター ポッドをスケジュールして実行できるように、オペレーター デプロイメントに必要な許容値も追加する必要があります。

Kubernetesについて詳しく知る"汚名と寛容"。

戻る"* NetApp Kubernetes Monitoring Operator のインストール* ページ"

秘密についてのメモ

Kubernetes モニタリング オペレーターがクラスター全体でシークレットを表示する権限を削除するには、インストール前に *operator-setup.yaml* ファイルから次のリソースを削除します。

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

アップグレードの場合は、クラスター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* または *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。この行における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 モニタリング オペレーター イメージ署名の検証

オペレータのイメージとそれが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したり、Kubernetes アドミッション コントローラーを構成したりすることができます。詳細については、"Kubernetesドキュメント" 。

イメージ署名の検証に使用される公開鍵は、モニタリング オペレーターのインストール タイルの「オプション: オペレーター イメージをプライベート リポジトリにアップロード > イメージ署名公開鍵」で入手できます。

イメージ署名を手動で検証するには、次の手順を実行します。

1. 画像プルスニペットをコピーして実行する
2. リポジトリパスワードをコピーしてプロンプトが表示されたら入力します
3. イメージ署名公開鍵（例では dii-image-signing.pub）を保存します。
4. cosign を使用してイメージを検証します。共同署名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 モニタリング オペレーターの設定中に問題が発生した場合に試すことは次のとおりです。

問題：	これを試してください:
Kubernetes 永続ボリュームと対応するバックエンド ストレージ デバイス間のハイパーリンク/接続が表示されません。私の Kubernetes 永続ボリュームは、ストレージ サーバーのホスト名を使用して構成されています。	手順に従って既存の Telegraf エージェントをアンインストールし、最新の Telegraf 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 以降を使用している必要があり、Kubernetes クラスター 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：	これを試してください:
ログには次のようなメッセージが表示されます: E0901 15:21:39.962145 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: *v1.MutatingWebhookConfiguration の一覧を取得できませんでした: サーバーは要求されたリソースを見つけることができませんでした E0901 15:21:43.168161 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: *v1.Lease の一覧を取得できませんでした: サーバーは要求されたリソースを見つけることができませんでした (get leases.coordination.k8s.io) など。	これらのメッセージは、Kubernetes バージョン 1.20 未満で kube-state-metrics バージョン 2.0.0 以上を実行している場合に表示されることがあります。 Kubernetes のバージョンを取得するには: *kubectl version* kube-state-metrics のバージョンを取得するには: *kubectl get deploy/kube-state-metrics -o jsonpath='{..image}'* これらのメッセージが表示されないようにするには、ユーザーは kube-state-metrics デプロイメントを変更して、次のリースを無効にすることができます: *mutatingwebhookconfigurations validatingwebhookconfigurations volumeattachments resources* 具体的には、次の CLI 引数を使用できます: resources=certificatesigningrequests,configmaps,cronjobs,daemonsets, deployments,endpoints,horizontalpodautoscalers,ingresses,jobs,limitranges, namespaces,networkpolicies,nodes,persistentvolumeclaims,persistentvolumes, poddisruptionbudgets,pods,replicasets,replicationcontrollers,resourcequotas, secrets、services 、statefulsets、storageclasses デフォルトのリソースリストは次のとおりです: "certificatesigningrequests、configmaps、cronjobs、daemonsets、deployments、endpoints、horizontalpodautoscalers、ingresses、jobs、leases、limitranges 、mutatingwebhookconfigurations、namespaces、networkpolicies、nodes、persistentvolumeclaims、persistentvolumes、poddisruptionbudgets、pods、replicasets、replicationcontrollers、resourcequotas、secrets 、services、statefulsets、storageclasses、validatingwebhookconfigurations、volumeattachments"
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: Oct 11 14:23:41 ip-172-31-39-47 systemd[1]: Started The plugin-driven server agent for reporting metrics into InfluxDB. 10月11日 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="キャッシュディレクトリの作成に失敗しました。 /etc/telegraf/.cache/snowflake、err: mkdir /etc/telegraf/.ca che: 権限が拒否されました。無視されました\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" Oct 11 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="開けませんでした。無視されました。open /etc/telegraf/.cache/snowflake/ocsp_response_cache.json: そのようなファイルまたはディレクトリはありません\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" Oct 11 14:23:41 ip-172-31-39-47 telegraf[1827]: 2021-10-11T14:23:41Z I! Telegraf 1.19.3 の起動	これは既知の問題です。参照"このGitHubの記事"詳細についてはこちらをご覧ください。 Telegraf が稼働している限り、ユーザーはこれらのエラー メッセージを無視できます。

問題：	これを試してください:
Kubernetes では、Telegraf ポッドが次のエラーを報告しています: 「mountstats 情報の処理中にエラーが発生しました: mountstats ファイルを開けませんでした: /hostfs/proc/1/mountstats、エラー: open /hostfs/proc/1/mountstats: 権限が拒否されました」	SELinux が有効になっていて強制されている場合、Telegraf ポッドが Kubernetes ノード上の /proc/1/mountstats ファイルにアクセスできない可能性があります。この制限を克服するには、エージェント構成を編集し、runPrivileged 設定を有効にします。詳細については、OpenShift の手順を参照してください。
Kubernetes では、Telegraf ReplicaSet ポッドが次のエラーを報告しています: [inputs.prometheus] プラグインのエラー: キーペア /etc/kubernetes/pki/etcd/server.crt をロードできませんでした:/etc/kubernetes/pki/etcd/server.key: open /etc/kubernetes/pki/etcd/server.crt: そのようなファイルまたはディレクトリはありません	Telegraf ReplicaSet ポッドは、マスターまたは etcd として指定されたノード上で実行されることを目的としています。 ReplicaSet ポッドがこれらのノードのいずれかで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 taint があるかどうかを確認します。そうなる場合は、Telegraf ReplicaSet (telegraf-rs) に必要な許容範囲を追加します。たとえば、ReplicaSet を編集します… kubectl edit rs telegraf-rs …そして、適切な許容値を仕様に追加します。次に、ReplicaSet ポッドを再起動します。
PSP/PSA環境があります。これは監視オペレーターに影響しますか?	Kubernetes クラスターが Pod Security Policy (PSP) または Pod Security Admission (PSA) を適用した状態で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 をサポートする現在の Operator にアップグレードするには、次の手順に従います。1. アンインストール以前の監視オペレーター: 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. インストール監視オペレータの最新バージョン。
PSP/PSA を使用しているのですが、Operator を展開しようとして問題が発生しました。	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」としてマークします。これにより、ポッド セキュリティ ポリシーとポッド セキュリティ アドミッションが無効になり、オペレーターがデプロイ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 (Pod Security Policy が削除されたことが表示されます) kubectl get all -n <namespace>
grep -i psp (何も見つからないことが表示されます)	「ImagePullBackoff」エラーが発生
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 モニタリング オペレーターがそれを適切に認識するようにまだ構成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。	モニタリング オペレーターのデプロイメントで問題が発生していますが、現在のドキュメントでは解決できません。

問題：	これを試してください:
次のコマンドの出力をキャプチャまたはメモして、テクニカル サポート チームに連絡してください。 ``` 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) ポッドは CrashLoopBackOff にあります
これらのポッドは、ネットワーク可観測性のワークロード マップ データ コレクターに対応します。以下を試してください: • いずれかのポッドのログをチェックして、最小カーネル バージョンを確認します。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 は、最小カーネル バージョン 4.18.0 より小さいです","time":"2022-11-09T08:23:08Z"} ---- • Net-observer ポッドでは、Linux カーネル バージョンが少なくとも 4.18.0 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。	ポッドはオペレーター名前空間（デフォルト：netapp-monitoring）で実行されていますが、ワークロードマップのUIやクエリのKubernetesメトリックにデータが表示されません。
K8S クラスターのノード上の時刻設定を確認します。正確な監査とデータ レポートを実現するために、ネットワーク タイム プロトコル (NTP) または簡易ネットワーク タイム プロトコル (SNTP) を使用してエージェント マシンの時刻を同期することを強くお勧めします。	オペレーター名前空間内の一部のネットオブザーバーポッドが保留状態になっています
Net-observer は DaemonSet であり、k8s クラスターの各ノードでポッドを実行します。 • 保留中の状態のポッドに注意し、CPU またはメモリのリソースの問題が発生しているかどうかを確認します。ノードで必要なメモリと CPU が使用可能であることを確認します。	Kubernetes モニタリング オペレーターをインストールした直後、ログに次の内容が表示されます: [inputs.prometheus] プラグインでエラーが発生しました: http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics への HTTP リクエストの作成エラー: http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics を取得: tcp をダイヤル: kube-state-metrics.<namespace>.svc.cluster.local を検索: そのようなホストはありません

問題：	これを試してください:
このメッセージは通常、新しいオペレータがインストールされ、*ksm* ポッドが起動する前に *telegraf-rs* ポッドが起動している場合にのみ表示されます。すべてのポッドが実行されると同時に、これらのメッセージは停止します。	クラスター内に存在する Kubernetes CronJobs に対して収集されているメトリックが表示されません。
Kubernetesのバージョンを確認してください（つまり `kubectl version`）。 v1.20.x 以下の場合、これは予想される制限です。 Kubernetes Monitoring Operator とともにデプロイされた kube-state-metrics リリースは、v1.CronJob のみをサポートします。 Kubernetes 1.20.x 以下では、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。	オペレーターをインストールすると、telegraf-ds ポッドは CrashLoopBackOff 状態になり、ポッド ログに「su: 認証失敗」と表示されます。
AgentConfiguration の telegraf セクションを編集し、*dockerMetricCollectionEnabled* を false に設定します。詳細については、オペレーターの"設定オプション"を参照してください。… spec：… telegraf：…-name：docker run-mode：- DaemonSet substitutions：- key：DOCKER_UNIX_SOCK_PLACEHOLDER value：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: E! [エージェント] 出力への書き込みエラー: http: Post "https://<tenant_url>/rest/v1/lake/ingest/influxdb": コンテキストの期限が切れました (ヘッダーの待機中に Client.Timeout を超えました)
AgentConfiguration の telegraf セクションを編集し、*outputTimeout* を 10 秒に増やします。詳細については、オペレーターの"設定オプション"。	一部のイベント ログの *involvedobject* データが見つかりません。
必ず、"権限"上記のセクション。	netapp-ci-monitoring-operator-<pod> と monitoring-operator-<pod> という名前の 2 つの監視オペレータ ポッドが実行されているのはなぜですか?
2023年10月12日現在、 Data Infrastructure Insights は、ユーザーへのサービス向上のため、オペレーターをリファクタリングしました。これらの変更を完全に適用するには、古い演算子を削除するそして新しいものをインストールする。	Kubernetes イベントが予期せずData Infrastructure Insightsへのレポートを停止しました。
イベント エクスポーター ポッドの名前を取得します。 `kubectl -n netapp-monitoring get pods`	grep event-exporter

問題：	これを試してください:
awk '{print $1}'	sed 's/event-exporter./event-exporter/'` 「netapp-ci-event-exporter」または「event-exporter」のいずれかである必要があります。次に、監視エージェントを編集します `kubectl -n netapp-monitoring edit agent`、LOG_FILE の値を、前の手順で見つかった適切なイベント エクスポーター ポッド名を反映するように設定します。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、uninstallそして再インストールエージェント。
Kubernetes モニタリング オペレーターによってデプロイされたポッドが、リソース不足のためにクラッシュしているのがわかります。	Kubernetesモニタリングオペレーターを参照してください"設定オプション"必要に応じて CPU および/またはメモリの制限を増やします。
イメージが欠落しているか、構成が無効であるため、netapp-ci-kube-state-metrics ポッドの起動または準備ができませんでした。現在、StatefulSet はスタックしており、構成の変更が netapp-ci-kube-state-metrics ポッドに適用されていません。	StatefulSetは"壊れた"州。構成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 ポッドは、Kubernetes Operator のアップグレードを実行した後に起動に失敗し、ErrImagePull (イメージのプルに失敗します) をスローします。	ポッドを手動でリセットしてみてください。
ログ分析の Kubernetes クラスターで、「イベントは maxEventAgeSeconds より古いため破棄されました」というメッセージが表示されています。	Operator の *agentconfiguration* を変更し、*event-exporter-maxEventAgeSeconds* (つまり 60 秒)、*event-exporter-kubeQPS* (つまり 100)、および *event-exporter-kubeBurst* (つまり 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、"設定オプション"ページ。

問題：	これを試してください:
Telegraf は、ロック可能なメモリが不足しているために警告を発したりクラッシュしたりします。	基盤となるオペレーティング システム/ノードで Telegraf のロック可能なメモリの制限を増やしてみてください。制限を増やすことができない場合は、NKMO エージェント構成を変更し、*unprotected* を *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されます。復号化された秘密がディスクにスワップアウトされる可能性があるため、セキュリティ上のリスクが生じ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の実行が可能になります。*unprotected* 設定オプションの詳細については、"設定オプション"ページ。
Telegraf から次のような警告メッセージが表示されます: *W! [inputs.diskio] "vdc"* のディスク名を収集できません*: /dev/vdc* の読み取りエラー*:* そのようなファイルまたはディレクトリはありません	Kubernetes Monitoring Operator の場合、これらの警告メッセージは無害であり、無視しても問題ありません。または、AgentConfiguration の 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、"オペレータの設定オプション"を参照してください。

問題：	これを試してください:

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] 入力 tail.0 の初期化に失敗しました [2024/10/16 14:16:23] [error] [engine] 入力の初期化に失敗しました	クラスター内の *fsnotify* 設定を変更してみます。
	
	`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。
	
	注意: 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>`

Telegraf DS ポッドは、TLS 証明書を検証できないために kubernetes 入力プラグインが HTTP リクエストを実行できないことに関連するエラーを報告しています。例: E! [inputs.kubernetes] プラグインのエラー: HTTPリクエストの送信中にエラーが発生しました"<a href="https://<kubelet_IP>:10250/stats/summary":" class="bare">https://<kubelet_IP>:10250/stats/summary":得る"<a href="https://<kubelet_IP>:10250/stats/summary":" class="bare">https://<kubelet_IP>:10250/stats/summary":tls: 証明書の検証に失敗しました: x509: IP SANが含まれていないため、<kubelet_IP>の証明書を検証できません	これは、kubelet が自己署名証明書を使用している場合、および/または指定された証明書の証明書の *Subject Alternative Name* リストに <kubelet_IP> が含まれていない場合に発生します。これを解決するには、ユーザーは"エージェント構成"、*telegraf:insecureK8sSkipVerify* を *true* に設定します。これにより、Telegraf 入力プラグインが検証をスキップするように設定されます。あるいは、ユーザーはkubeletを次のように設定することができます。"サーバーTLSブートストラップ"これにより、「certificates.k8s.io」API からの証明書要求がトリガーされます。

問題：	これを試してください:
Fluent-bit ポッドで次のエラーが発生し、ポッドを起動できません：026/01/12 20:20:32] [error] [sqldb] error=unable to open database file [2026/01/12 20:20:32] [error] [input:tail:tail.0] db: could not create 'in_tail_files' table [2026/01/12 20:20:32] [error] [input:tail:tail.0] could not open/create database [2026/01/12 20:20:32] [error] failed initialize input tail.0 [2026/01/12 20:20:32] [error] [engine] input initialization failed	DBファイルが存在するホストディレクトリに適切な読み取り / 書き込み権限があることを確認してください。具体的には、ホストディレクトリは非ルートユーザーに読み取り / 書き込み権限を付与する必要があります。デフォルトのDBファイルの場所は、fluent-bit-dbFile *agentconfiguration* オプションで上書きされない限り、/var/log/ です。SELinuxが有効になっている場合は、fluent-bit-seLinuxOptionsType agentconfigurationオプションを 'spc_t' に設定してみてください。

追加情報は以下からご覧いただけます。"サポート"ページまたは"データコレクターサポートマトリックス"。

Kubernetes モニタリング オペレーターの設定オプション

その"Kubernetes モニタリング オペレーター" AgentConfiguration ファイルを通じて、幅広いカスタマイズオプションが提供されます。リソース制限、収集間隔、プロキシ設定、許容値、コンポーネント固有の設定を構成して、Kubernetes 環境の監視を最適化できます。これらのオプションを使用して、Telegraf、kube-state-metrics、ログ収集、ワークロードマッピング、変更管理、その他の監視コンポーネントをカスタマイズできます。

サンプル **AgentConfiguration** ファイル

以下に、各オプションの説明を含むサンプルの *AgentConfiguration* ファイルを示します。

```
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
```

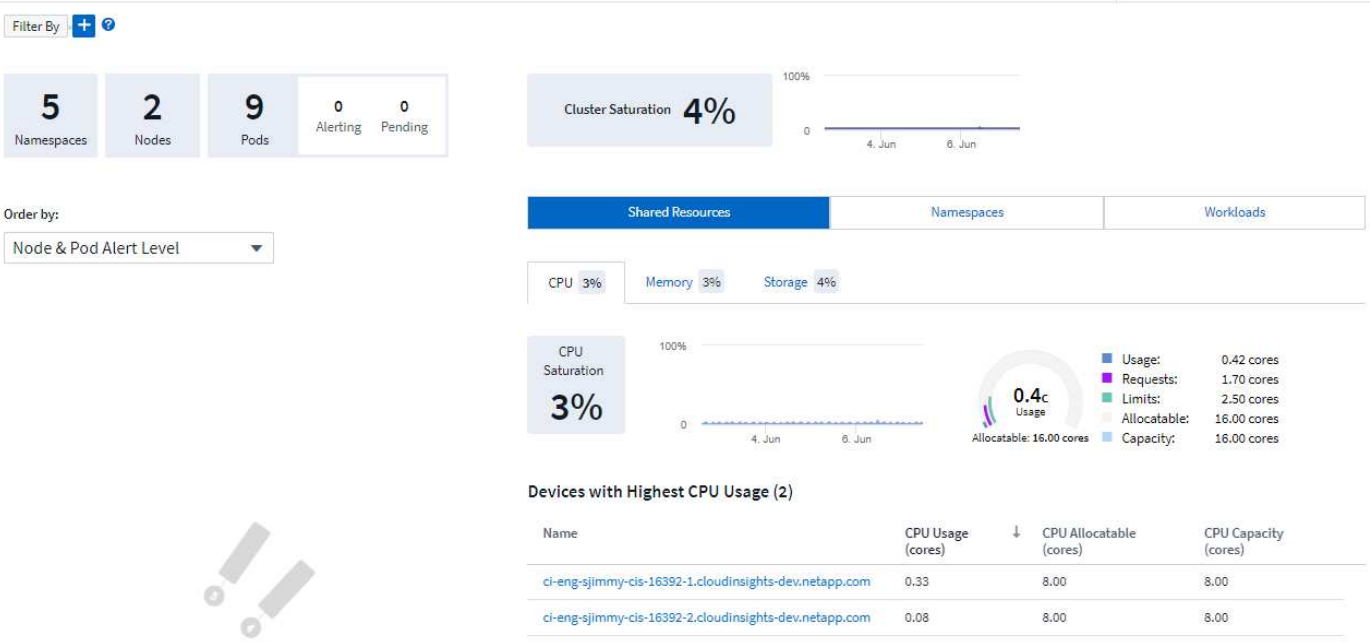
```
    ##
    # enableDNSLookup: 'true'

    ##
    ## 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: ''
```

Kubernetes クラスターの詳細ページ

Kubernetes クラスターの詳細ページには、Kubernetes クラスターの詳細な概要が表示されます。



名前空間、ノード、ポッドの数

ページ上部のカウントには、クラスター内の名前空間、ノード、ポッドの合計数と、現在アラートおよび保留中の popd の数が表示されます。

共有リソースと飽和

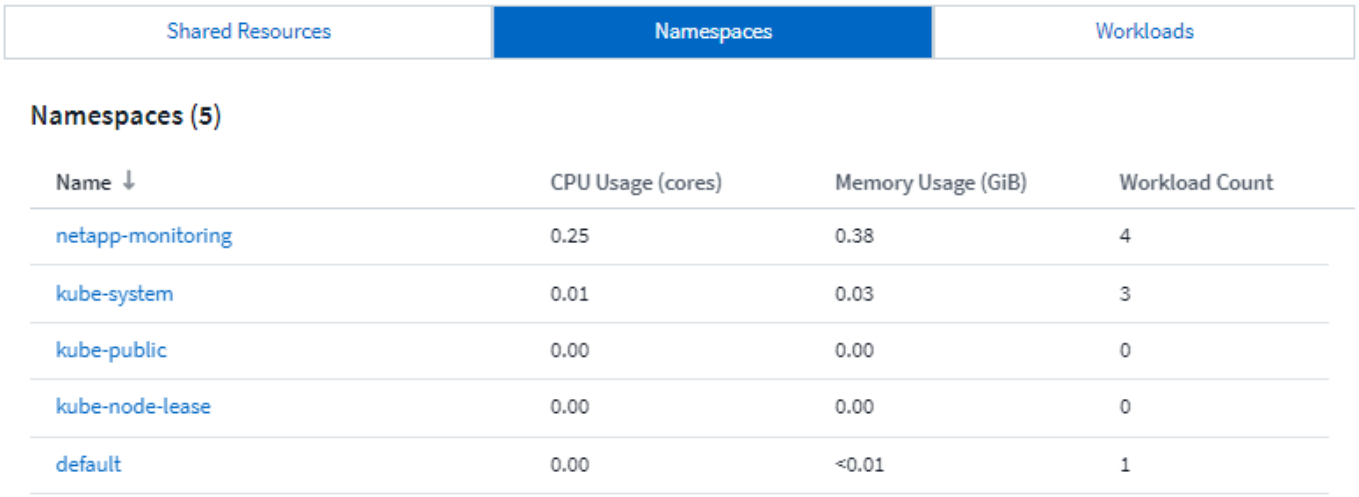
詳細ページの右上には、現在のクラスターの飽和度がパーセンテージで表示され、時間の経過に伴う最近の傾向を示すグラフも表示されます。クラスターの飽和度は、各時点での CPU、メモリ、またはストレージの飽和度のうち最も高い値です。

その下には、デフォルトで 共有リソース の使用状況が表示され、CPU、メモリ、ストレージのタブがあります。各タブには、飽和率と時間の経過に伴う傾向、および追加の使用状況の詳細が表示されます。ストレージの場合、表示される値は、個別に計算されるバックエンドとファイルシステムの飽和度のうち大きい方になります。

使用頻度が最も高いデバイスは下部の表に表示されます。これらのデバイスを調べるには、いずれかのリンクをクリックしてください。

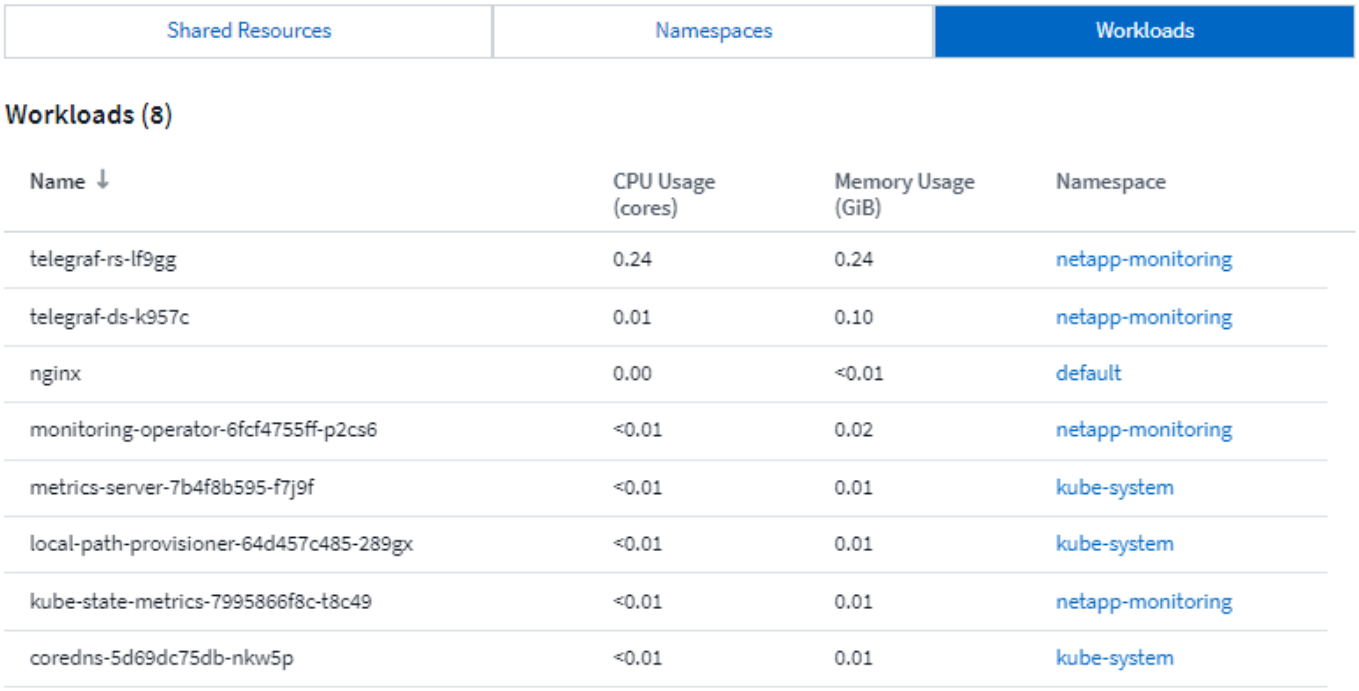
ネームスペース

「名前空間」タブには、Kubernetes 環境内のすべての名前空間のリストが表示され、CPU とメモリの使用率、および各名前空間のワークロード数が表示されます。名前リンクをクリックして、各名前空間を調べます。



ワークロード

同様に、[ワークロード] タブには、各名前空間のワークロードのリストが表示され、CPU とメモリの使用状況も表示されます。名前空間リンクをクリックすると、それぞれの詳細が表示されます。



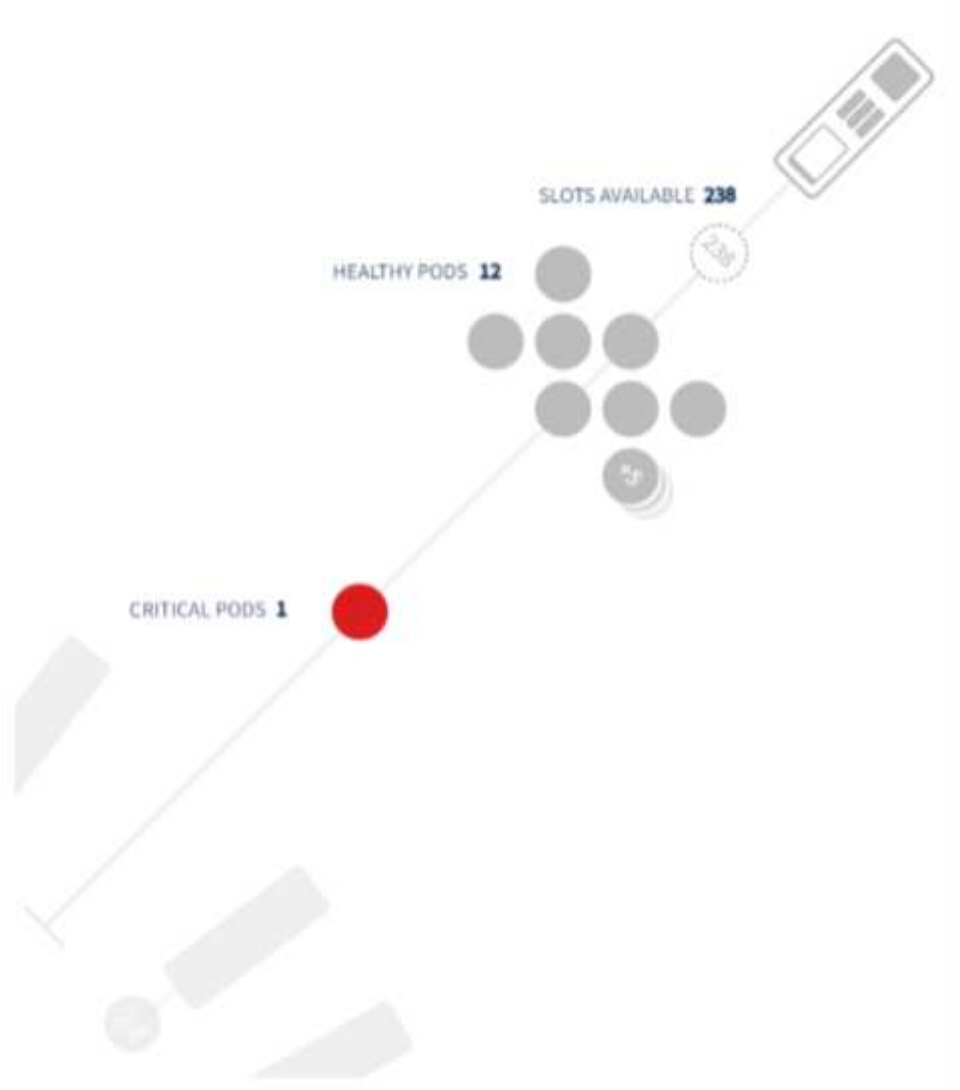
クラスター「ホイール」

クラスターの「ホイール」セクションでは、ノードとポッドの健全性が一目でわかり、さらに詳しい情報にドリルダウンできます。クラスターにページのこの領域に表示できる数を超えるノードが含まれている場合は、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。

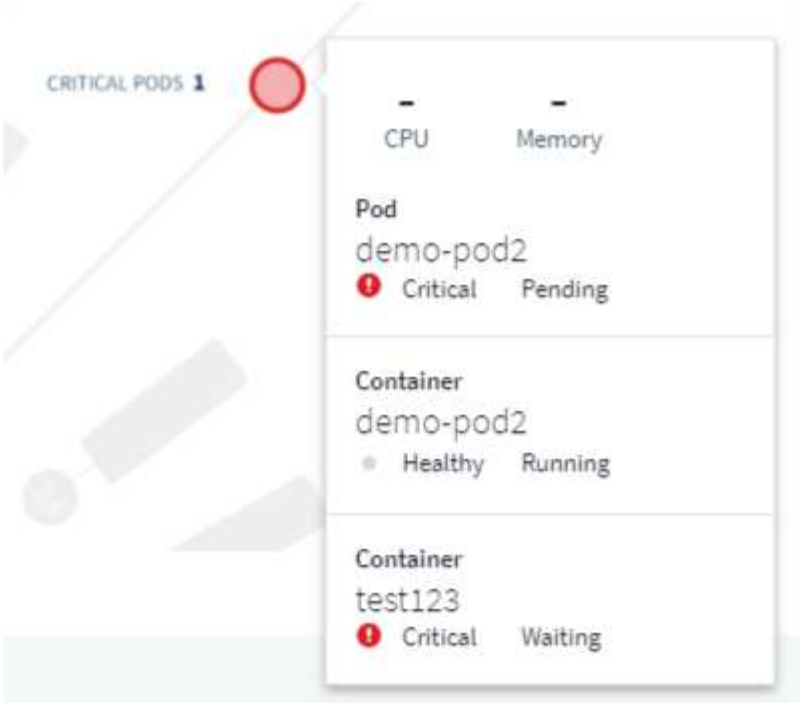
アラートを発しているポッドまたはノードは赤で表示されます。 「警告」エリアはオレンジ色で表示されます。スケジュールされていない (つまり、接続されていない) ポッドは、クラスターの「ホイール」の下隅に表示されます。

ポッド (円) またはノード (バー) の上にマウスを移動すると、ノードのビューが拡張されます。

そのビュー内のポッドまたはノードをクリックすると、拡張されたノード ビューが拡大表示されます。



ここから、要素の上にマウスを置くと、その要素の詳細が表示されます。たとえば、この例の重要なポッドにマウスを移動すると、そのポッドの詳細が表示されます。



ノード要素にマウスを移動すると、ファイルシステム、メモリ、CPU の情報を表示できます。

ゲージに関する注意

メモリ ゲージと CPU ゲージは、割り当て可能な容量 と 合計容量 の両方に関連して 使用済み を表示するため、3 色で表示されます。

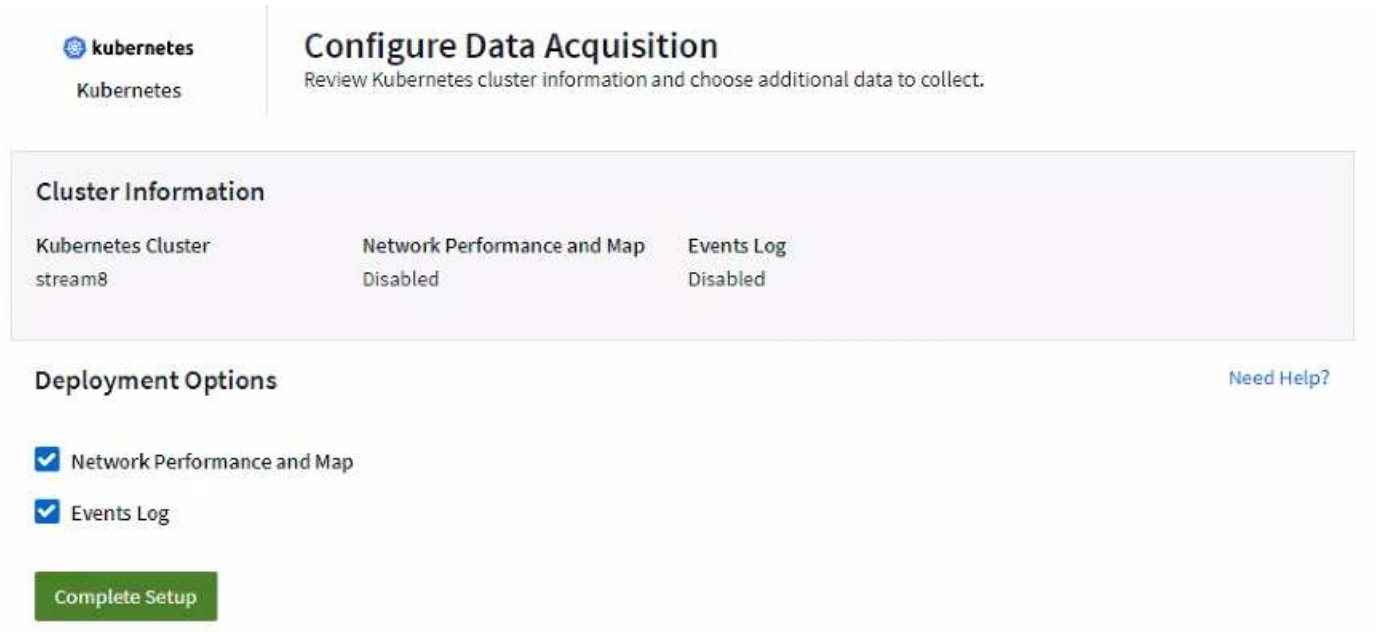
Kubernetes ネットワークパフォーマンス監視とマップ

Kubernetes ネットワーク パフォーマンス モニタリングおよびマップ機能は、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素化し、ネットワーク パフォーマンスの遅延と異常をリアルタイムで可視化して、ユーザーに影響が及ぶ前にパフォーマンスの問題を特定します。この機能により、組織は Kubernetes トラフィック フローを分析および監査して全体的なコストを削減できます。

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表示します。 • 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。 • イングレス、エグレス、クロスリージョン、クロスゾーンのネットワーク 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ネットワークパフォーマンスモニタリングとマップを使用する前に、"NetApp Kubernetes 監視オペレーター"このオプションを有効にします。オペレーターのデプロイ中に、「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選択して有効にします。 Kubernetes ランディング ページに移動して「デプロイメントの変更」を選択することで、このオプションを有効にすることもできます。



モニター

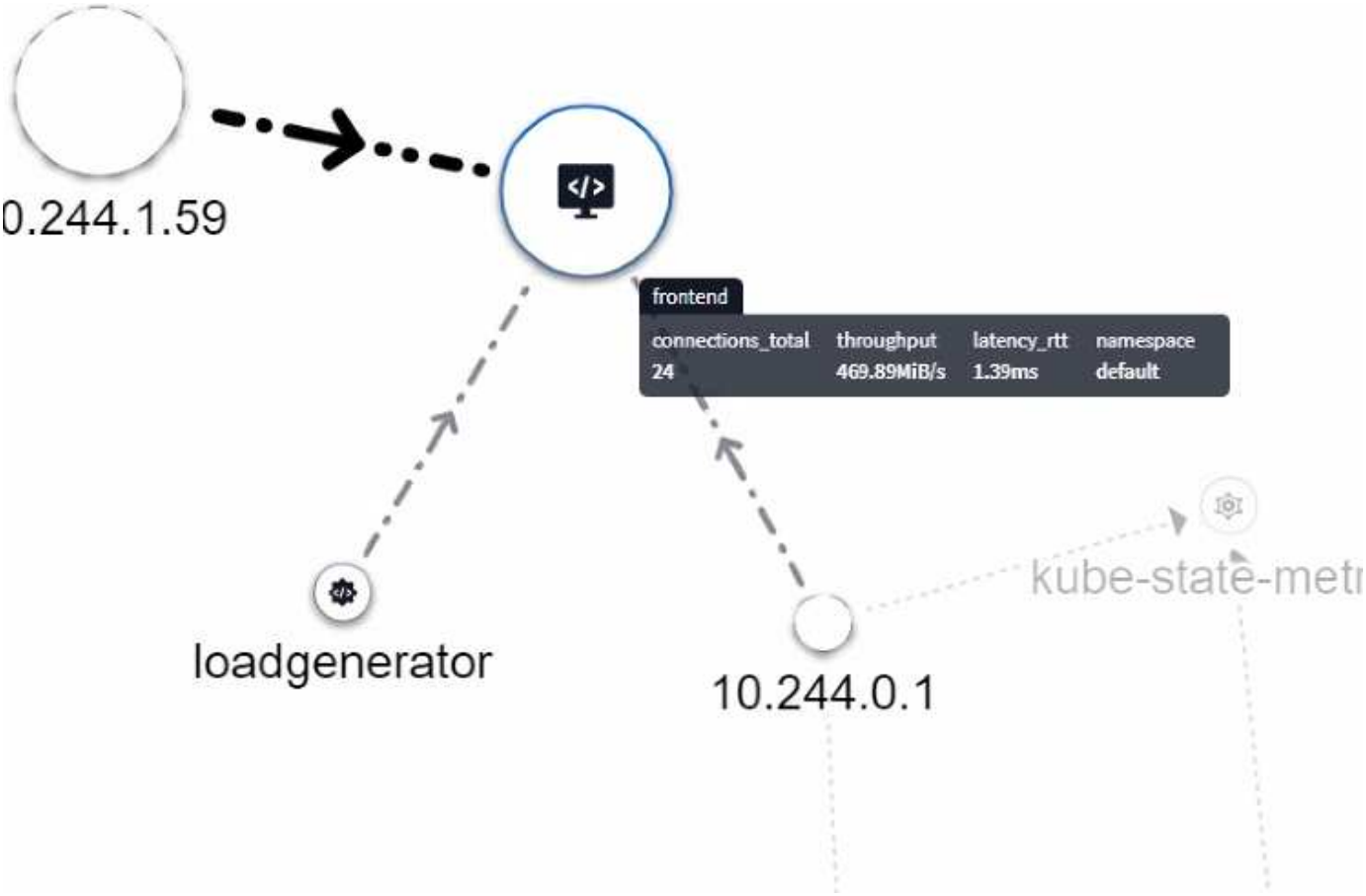
ワークロードマップは"監視"情報を導き出す。 Data Infrastructure Insights は、いくつかのデフォルトの Kubernetes モニターを提供します (これらはデフォルトで 一時停止 されている可能性があることに注意してください)。必要なモニターを 再開 (つまり有効化) することも、ワークロード マップでも使用される Kubernetes オブジェクト用のカスタム モニターを作成することもできます。

以下のいずれかのオブジェクト タイプに対して、 Data Infrastructure Insightsメトリック アラートを作成できます。データがデフォルトのオブジェクト タイプ別にグループ化されていることを確認します。

- kubernetes.ワークロード
- kubernetes.daemonset
- kubernetes.デプロイメント
- kubernetes.cronジョブ
- kubernetes.ジョブ
- kubernetes.レプリカセット
- 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ます。ワークロードの上にマウスを置くと、次の例のように、そのワークロードの概要情報が表示されます。



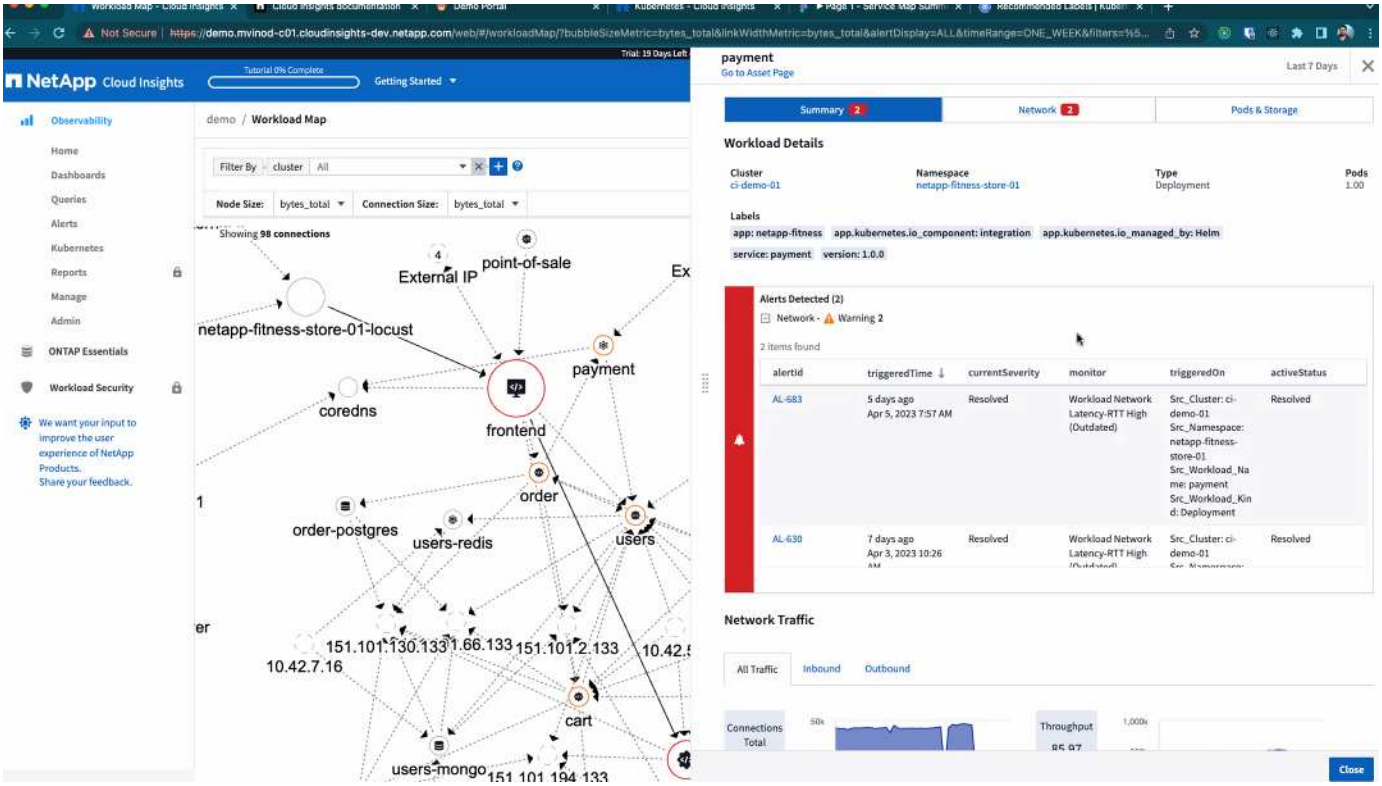
円内のアイコンは、さまざまなサービス タイプを表します。アイコンは、基礎となるオブジェクトがラベル。

各円のサイズはノードのサイズを示します。これらのサイズは相対的なものであり、ブラウザのズーム レベルや画面サイズによって実際の円のサイズが変わる可能性があることに注意してください。同様に、トラフィック ライン スタイルを使用すると、接続サイズが一目でわかります。太い実線はトラフィック量が多く、細い点線はトラフィック量が少ないことを示します。

円内の数字は、現在サービスによって処理されている外部接続の数です。

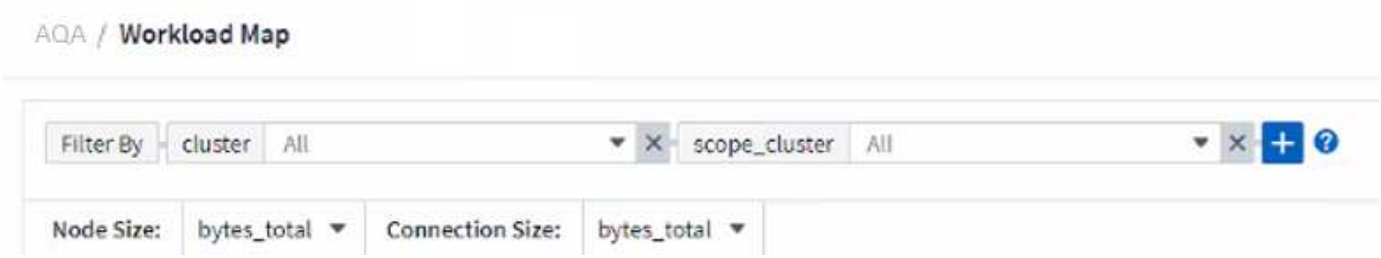
ワークロードの詳細とアラート

色で表示される円は、ワークロードの警告レベルまたは重大レベルのアラートを示します。問題の概要を表示するには円の上にマウスを移動するか、円をクリックして詳細を示すスライドアウト パネルを開きます。

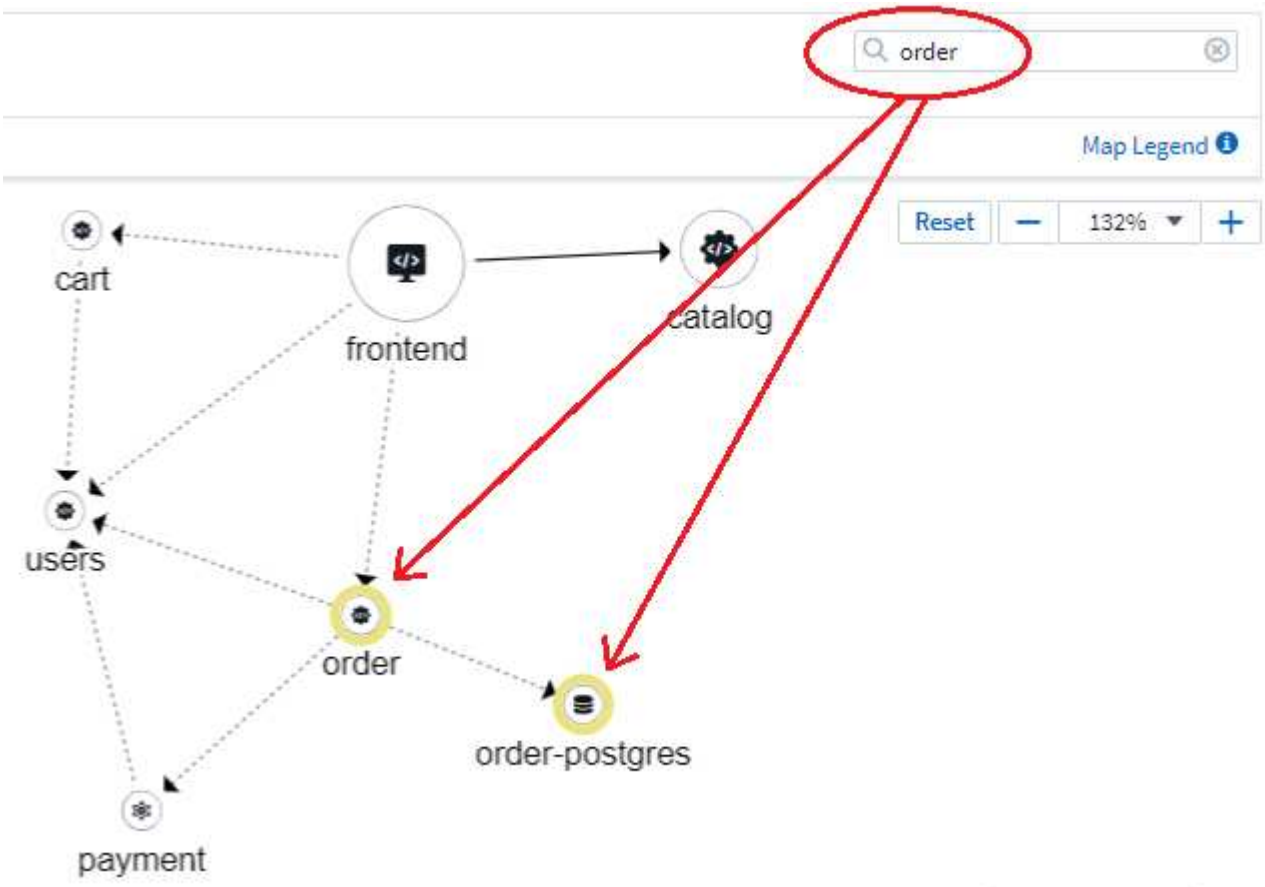


検索とフィルタリング

他のData Infrastructure Insights機能と同様に、必要な特定のオブジェクトまたはワークロード属性に焦点を絞るためのフィルターを簡単に設定できます。



同様に、[検索] 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強調表示されます。



ワークロードラベル

マップに表示されるワークロードの種類 (円形アイコンなど) を識別する場合は、ワークロード ラベルが必要です。ラベルは次のように生成されます。

- 実行中のサービス/アプリケーションの一般的な名前
- 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派生されます
  - ワークロードに期待されるラベル: 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

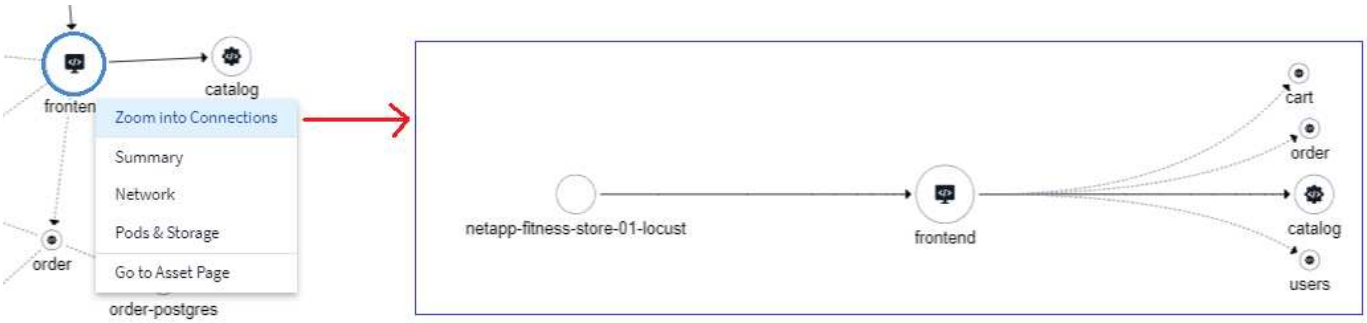
- バックエンド
- データベース
- キャッシュ
- キュー
- カフカ

- ソースが Kubernetes クラスターの外部にある場合:
  - Data Infrastructure Insights は、 DNS 解決された名前を解析してサービス タイプを抽出しようとします。

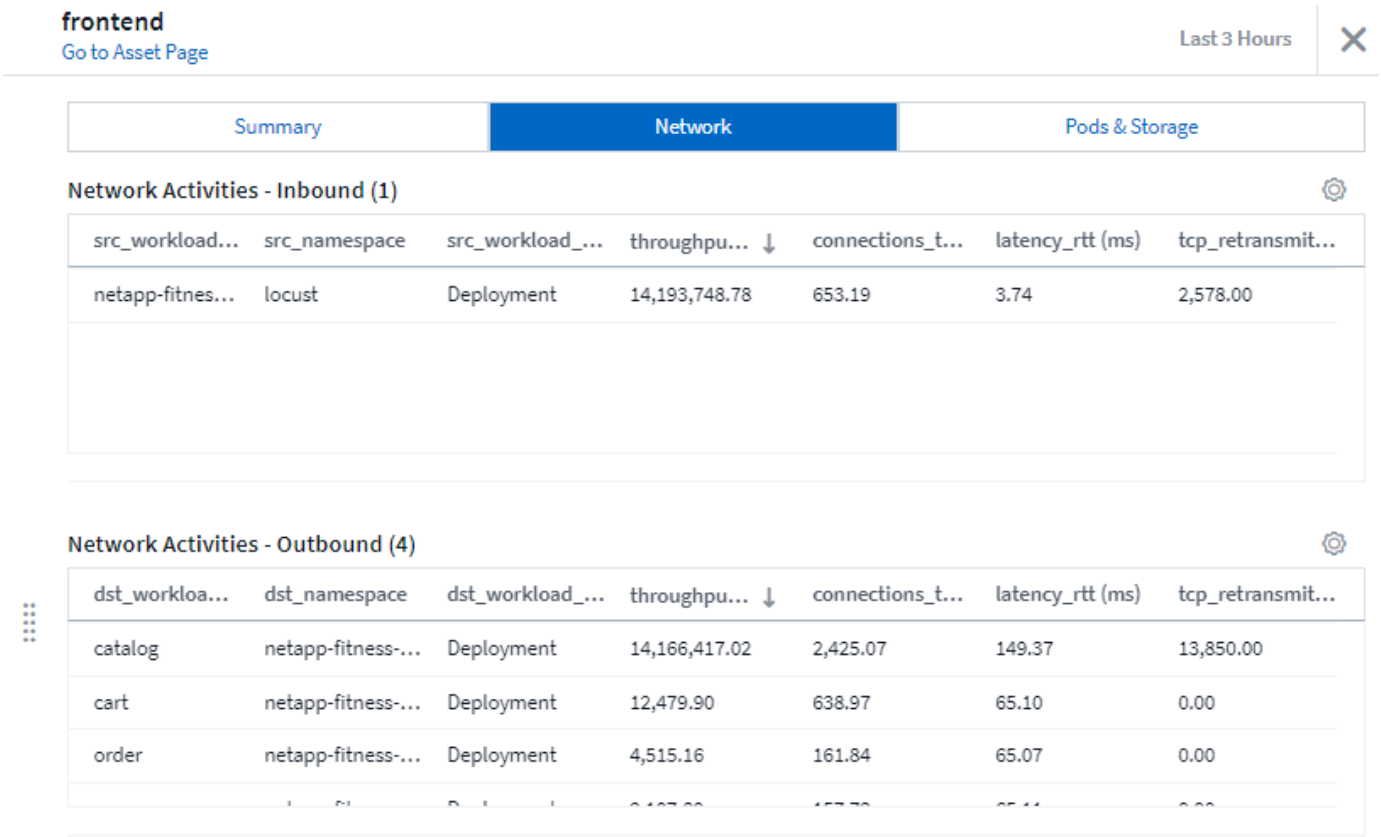
    たとえば、DNS 解決名が *s3.eu-north-1.amazonaws.com* の場合、解決された名前は解析され、サービス タイプとして *s3* が取得されます。

深く潜る

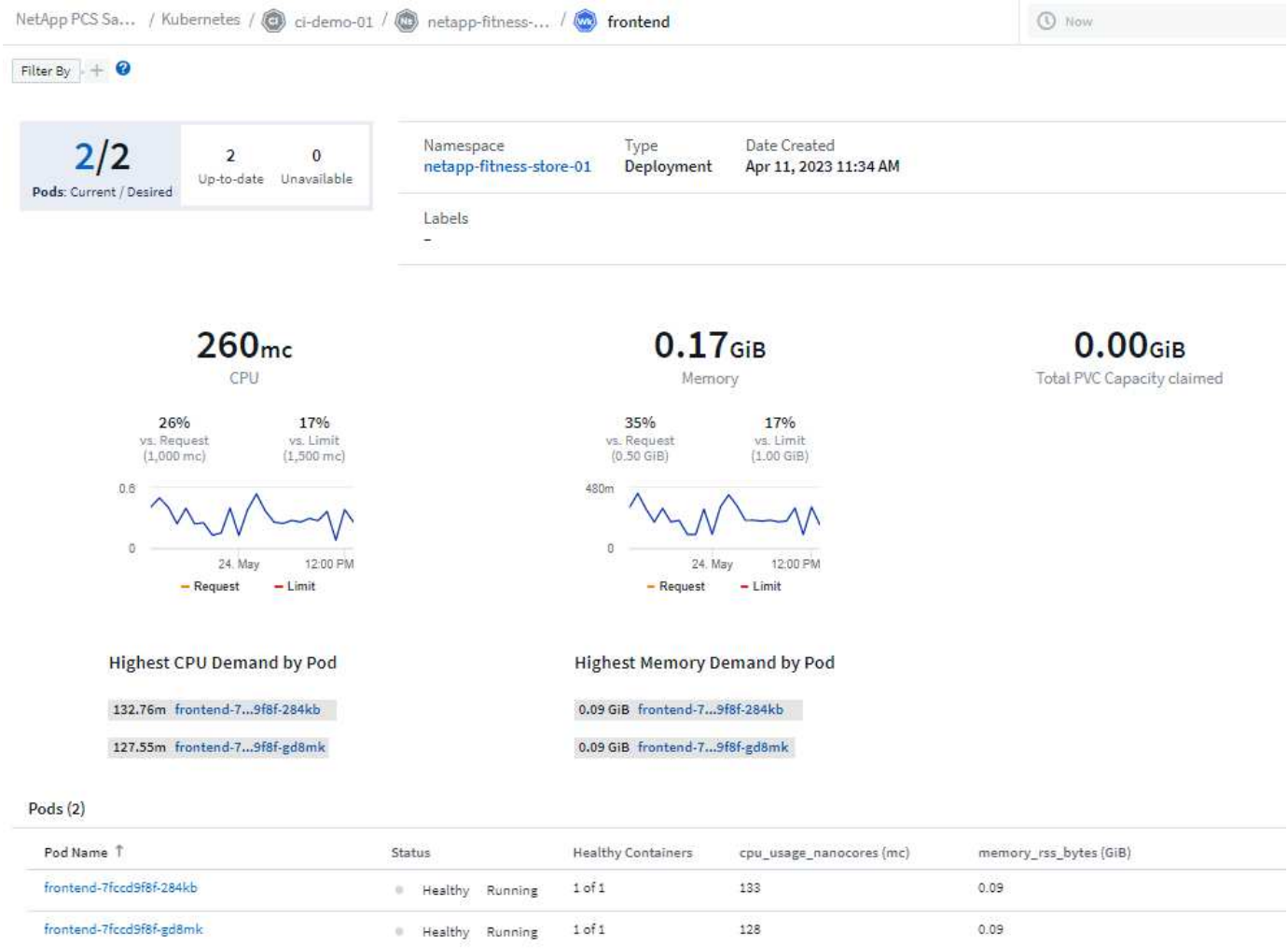
ワークロードを右クリックすると、さらに詳しく調べるための追加オプションが表示されます。たとえば、ここからズームインして、そのワークロードの接続を表示できます。



または、詳細スライドアウト パネルを開いて、[概要]、[ネットワーク]、または [ポッドとストレージ] タブを直接表示することもできます。



最後に、[アセット ページに移動] 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。



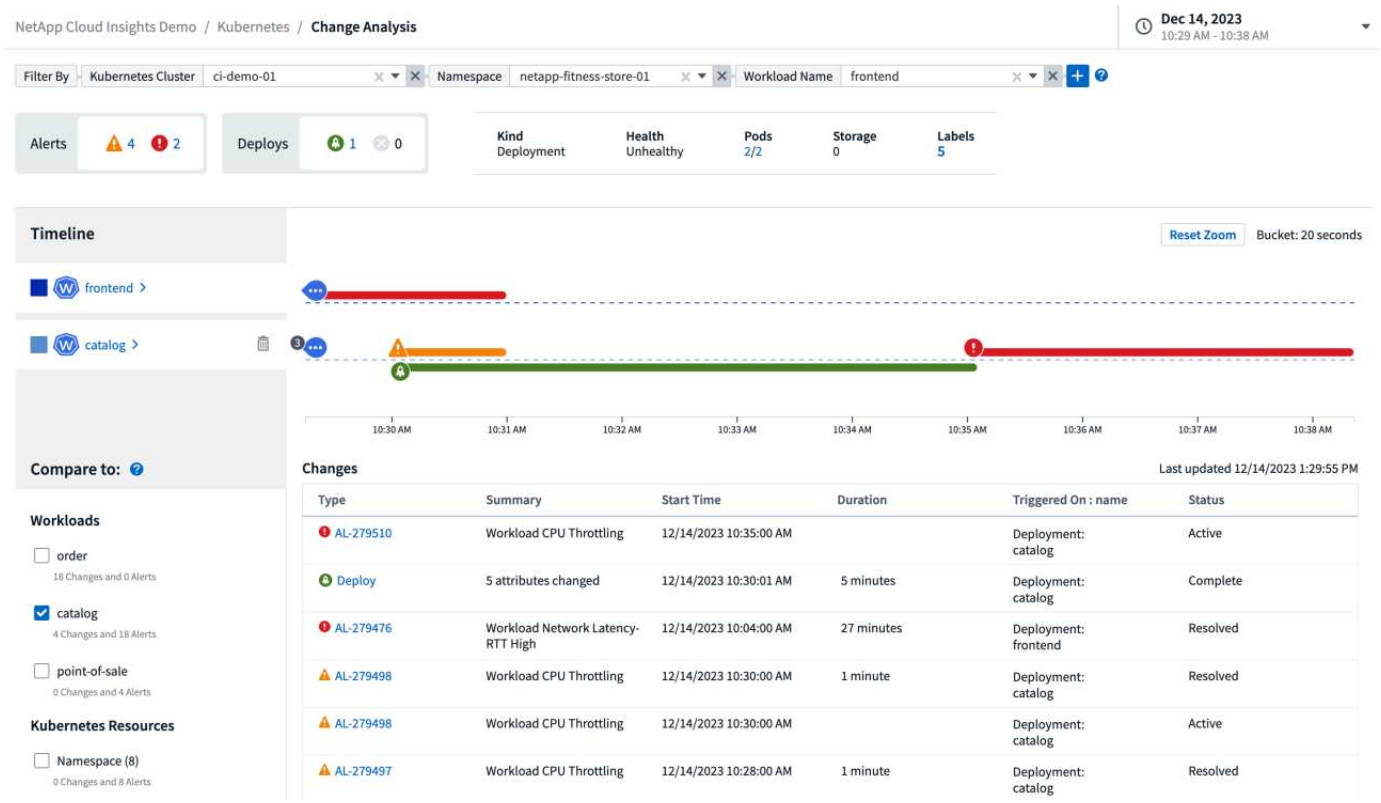
Kubernetes 変更分析

Kubernetes Change Analytics は、K8s 環境への最近の変更をワンストップで表示します。アラートと展開ステータスを簡単に確認できます。 Change Analytics を使用すると、すべてのデプロイメントと構成の変更を追跡し、それを K8s サービス、インフラストラクチャ、クラスターの健全性およびパフォーマンスと相関させることができます。

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、誤った構成変更により停止が発生する可能性があります。 Change Analytics は、ワークロードと構成の変更の健全性を表示して相関関係を調べるための単一のペインを提供することで、この作業を支援します。これは、動的な Kubernetes 環境のトラブルシューティングに役立つ可能性があります。

Kubernetes 変更分析を表示するには、**Kubernetes >** 変更分析 に移動します。



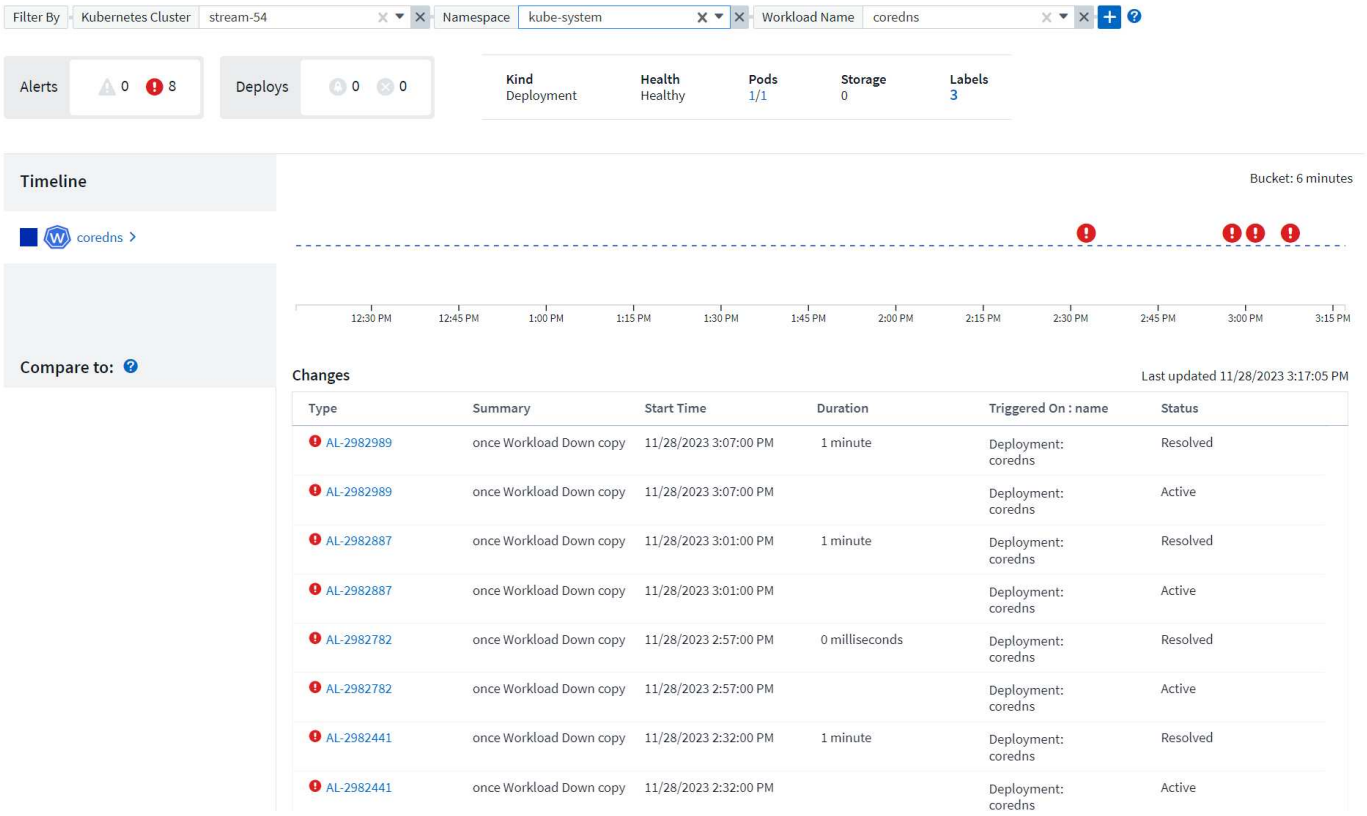
。

ページは、現在選択されているData Infrastructure Insightsの時間範囲に基づいて自動的に更新されます。時間範囲が狭いほど、画面の更新頻度が高くなります。

フィルタ

Data Infrastructure Insightsのすべての機能と同様に、変更リストのフィルタリングは直感的に行えます。ページの上部で、Kubernetes クラスター、名前空間、またはワークロードの値を入力または選択するか、[{+]] ボタンを選択して独自のフィルターを追加します。

特定のクラスター、名前空間、ワークロード (および設定したその他のフィルター) にフィルターすると、そのクラスターのその名前空間内のそのワークロードのデプロイメントとアラートのタイムラインが表示されます。グラフをクリックしてドラッグすると、さらに拡大して、より具体的な時間範囲に焦点を絞ることができます。

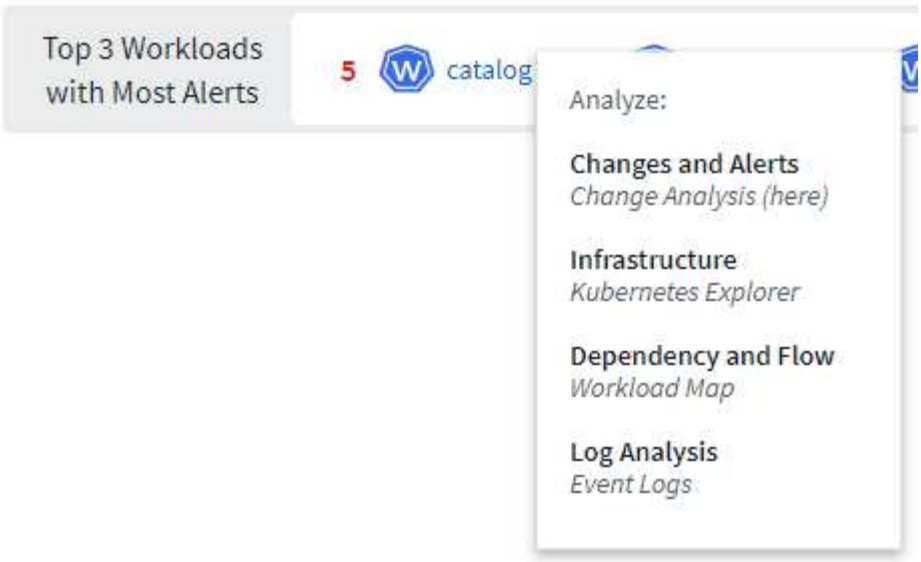


クイックステータス

フィルタリング領域の下には、いくつかの高レベルインジケーターがあります。左側にはアラートの数（警告と重大）が表示されます。この数には、_アクティブ_アラートと_解決済み_アラートが含まれます。 _アクティブ_アラートのみを表示するには、「ステータス」のフィルターを設定し、「アクティブ」を選択します。

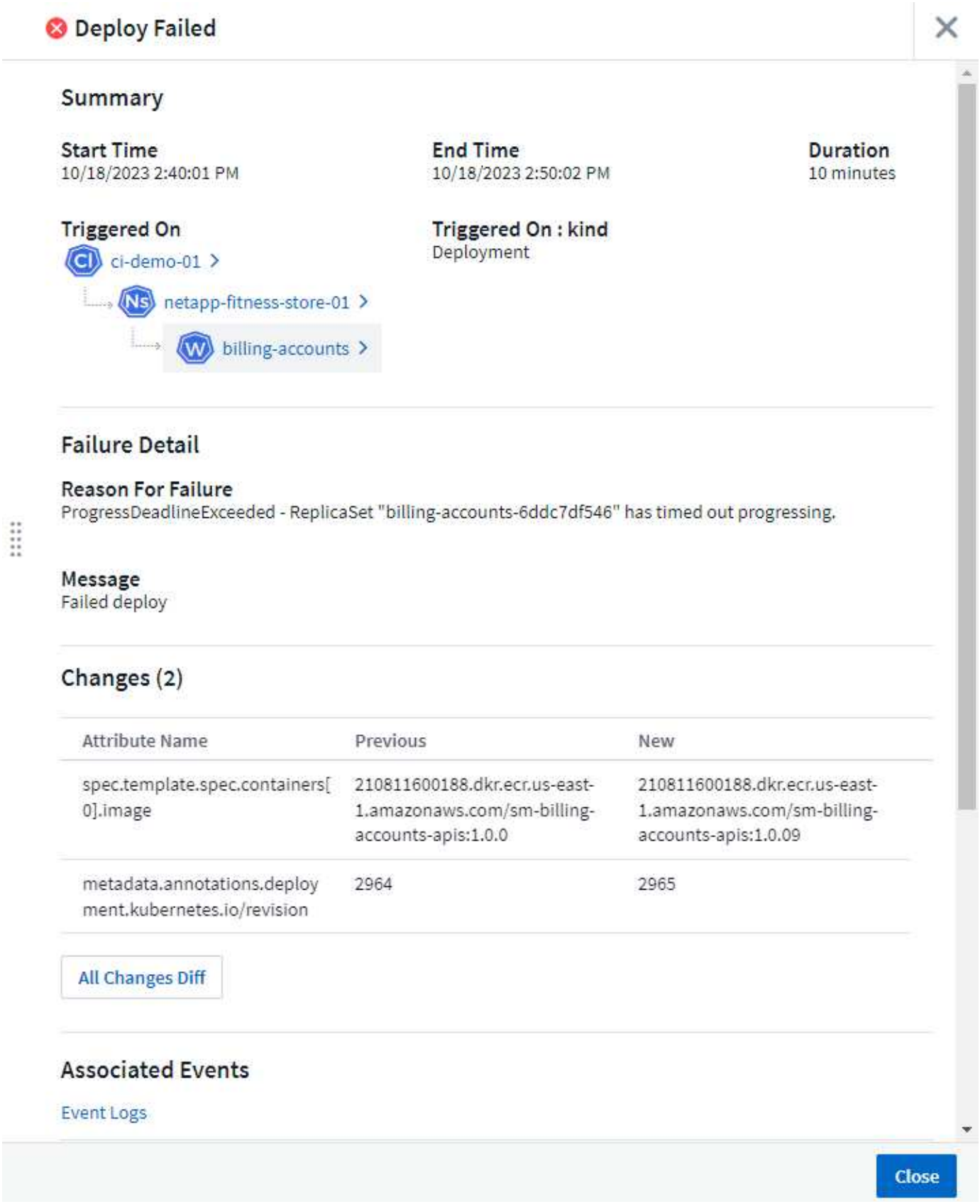
展開ステータスもここに表示されます。ここでも、デフォルトでは、開始、完了、および 失敗 のデプロイメントの数が表示されます。失敗したデプロイメントのみを表示するには、「ステータス」のフィルターを設定し、「失敗」を選択します。

次に、アラートが最も多い上位 3 つのワークロードを示します。各ワークロードの横にある赤い数字は、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 リンクをクリックして、インフラストラクチャ (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) を調べます。



詳細パネル

リスト内の変更を選択すると、変更の詳細を説明するパネルが開きます。たとえば、失敗したデプロイを選択すると、開始時刻と終了時刻、期間、デプロイがトリガーされた場所を含むデプロイの概要と、それらのリソースを調べ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、および関連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ーやアラートの視覚的なタイムラインを示すグラフなど、アラートに関する詳細が表示されます。