쿠버네티스
Data Infrastructure Insights

NetApp
February 11, 2026

목차

쿠버네티스

쿠버네티스 클러스터 개요

Data Infrastructure Insights Kubernetes Explorer는 Kubernetes 클러스터의 전반적인 상태와 사용량을 표시하는 강력한 도구로, 조사 영역을 쉽게 자세히 살펴볼 수 있습니다.

*대시보드 > Kubernetes Explorer*를 클릭하면 Kubernetes 클러스터 목록 페이지가 열립니다. 이 개요 페이지에는 테넌트의 Kubernetes 클러스터 표가 포함되어 있습니다.

[쿠버네티스 목록 페이지]

클러스터 목록

클러스터 목록에는 테넌트의 각 클러스터에 대한 다음 정보가 표시됩니다.

- 클러스터 이름. 클러스터 이름을 클릭하면 다음이 열립니다."상세 페이지" 해당 클러스터에 대해서요.
- 포화 백분율. 전반적인 포화도는 CPU, 메모리 또는 저장소 포화도 중 가장 높은 수준입니다.
- 클러스터의 노드 수. 이 숫자를 클릭하면 노드 목록 페이지가 열립니다.
- 클러스터의 **Pod** 수. 이 번호를 클릭하면 Pod 목록 페이지가 열립니다.
- 클러스터의 네임스페이스 수. 이 숫자를 클릭하면 네임스페이스 목록 페이지가 열립니다.
- 클러스터의 워크로드 수. 이 숫자를 클릭하면 작업량 목록 페이지가 열립니다.

필터 개선

필터링을 할 때 입력을 시작하면 현재 텍스트를 기반으로 *와일드카드 필터*를 만들 수 있는 옵션이 제공됩니다. 이 옵션을 선택하면 와일드카드 표현식과 일치하는 모든 결과가 반환됩니다. NOT 또는 AND를 사용하여 *표현식*을 만들 수도 있고, "없음" 옵션을 선택하여 필드에서 null 값을 필터링할 수도 있습니다.

[K8S Explorer에서 와일드카드로 필터링]

와일드카드나 표현식(예: NOT, AND, "없음" 등)을 기반으로 하는 필터는 필터 필드에 진한 파란색으로 표시됩니다. 목록에서 직접 선택한 항목은 밝은 파란색으로 표시됩니다.

[와일드카드 및 선택된 항목을 표시하는 필터]

Kubernetes 필터는 상황에 따라 다릅니다. 즉, 예를 들어 특정 노드 페이지에 있는 경우 pod_name 필터는 해당 노드와 관련된 Pod만 나열합니다. 게다가 특정 네임스페이스에 대한 필터를 적용하면 pod_name 필터는 해당 노드와 해당 네임스페이스에 있는 포드만 나열합니다.

와일드카드 및 표현식 필터링은 텍스트나 목록에서는 작동하지만 숫자, 날짜 또는 부울에서는 작동하지 않습니다.

NetApp Kubernetes Monitoring Operator를 설치하거나 업그레이드하기 전에

설치 또는 업그레이드하기 전에 이 정보를 읽으십시오."쿠버네티스 모니터링 운영자" .

요소	요구 사항
쿠버네티스 버전	Kubernetes v1.20 이상.
쿠버네티스 배포판	AWS Elastic Kubernetes Service(EKS) Azure Kubernetes Service(AKS) Google Kubernetes Engine(GKE) Red Hat OpenShift Rancher Kubernetes Engine(RKE) VMware Tanzu
리눅스 운영체제	Data Infrastructure Insights Arm64 아키텍처로 실행되는 노드를 지원하지 않습니다. 네트워크 모니터링: Linux 커널 버전 4.18.0 이상을 실행해야 합니다. Photon OS는 지원되지 않습니다.
라벨	Data Infrastructure Insights 다음 플랫폼에서 Kubernetes 레이블을 찾는 Kubernetes 노드 선택기를 지정하여 Linux를 실행하는 Kubernetes 노드의 모니터링을 지원합니다. Kubernetes v1.20 이상: Kubernetes.io/os = linux Rancher + cattle.io(오케스트레이션/Kubernetes 플랫폼): cattle.io/os = linux
명령	curl 및 kubectl 명령을 사용할 수 있어야 합니다. 최상의 결과를 얻으려면 이러한 명령을 PATH에 추가하세요.
연결성	kubectl cli는 대상 K8s 클러스터와 통신하도록 구성되었으며, Data Infrastructure Insights 환경에 인터넷으로 연결됩니다. 설치 중에 프록시 뒤에 있는 경우 다음 지침을 따르세요."프록시 지원 구성" 운영자 설치 섹션입니다. 정확한 감사 및 데이터 보고를 위해 NTP(Network Time Protocol) 또는 SNTP(Simple Network Time Protocol)를 사용하여 에이전트 컴퓨터의 시간을 동기화하세요.
다른	OpenShift 4.6 이상을 실행 중인 경우 다음을 따라야 합니다."OpenShift 지침" 이러한 전제 조건이 충족되는지 확인하는 것 외에도.
API 토큰	Operator를 다시 배포하는 경우(즉, 업데이트하거나 교체하는 경우) 새로운 API 토큰을 만들 필요가 없습니다. 이전 토큰을 다시 사용할 수 있습니다.

시작하기 전에 주의해야 할 중요한 사항

당신이 ~로 달리고 있다면대리 , 가지고있다사용자 정의 저장소 , 또는 사용 중오픈시프트 다음 섹션을 주의 깊게 읽어보세요.

또한 읽어보세요권한 .

프록시 지원 구성

테넌트에서 프록시를 사용하여 NetApp Kubernetes Monitoring Operator를 설치할 수 있는 두 곳이 있습니다. 이는 동일하거나 별도의 프록시 시스템일 수 있습니다.

- 설치 코드 조각을 실행하는 동안 필요한 프록시("curl" 사용)는 조각이 실행되는 시스템을 Data Infrastructure Insights 환경에 연결합니다.

- 대상 Kubernetes 클러스터가 Data Infrastructure Insights 환경과 통신하는 데 필요한 프록시

이 두 가지 중 하나 또는 둘 다에 프록시를 사용하는 경우 NetApp Kubernetes Operating Monitor를 설치하려면 먼저 프록시가 Data Infrastructure Insights 환경과의 원활한 통신을 허용하도록 구성되어 있는지 확인해야 합니다. 예를 들어, Operator를 설치하려는 서버/VM에서 Data Infrastructure Insights 에 액세스하고 Data Infrastructure Insights 에서 바이너리를 다운로드할 수 있어야 합니다.

NetApp Kubernetes Operating Monitor를 설치하는 데 사용되는 프록시의 경우, Operator를 설치하기 전에 *http_proxy/https_proxy* 환경 변수를 설정하세요. 일부 프록시 환경에서는 _no_proxy 환경 변수도 설정해야 할 수 있습니다.

변수를 설정하려면 NetApp Kubernetes Monitoring Operator를 설치하기 전에 시스템에서 다음 단계를 수행하세요.

1. 현재 사용자에 대해 *https_proxy* 및/또는 *http_proxy* 환경 변수를 설정합니다.
   a. 설정 중인 프록시에 인증(사용자 이름/비밀번호)이 없는 경우 다음 명령을 실행합니다.

```
 export https_proxy=<proxy_server>:<proxy_port>
.. 설정 중인 프록시에 인증(사용자 이름/비밀번호)이 있는 경우 다음 명령을 실행하세요.
```

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

Kubernetes 클러스터가 Data Infrastructure Insights 환경과 통신하는 데 사용되는 프록시의 경우, 이 지침을 모두 읽은 후 NetApp Kubernetes Monitoring Operator를 설치하세요.

NetApp Kubernetes Monitoring Operator를 배포하기 전에 operator-config.yaml에서 AgentConfiguration의 프록시 섹션을 구성합니다.

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

사용자 정의 또는 개인 **Docker** 저장소 사용

기본적으로 NetApp Kubernetes Monitoring Operator는 Data Infrastructure Insights 저장소에서 컨테이너 이미지를 가져옵니다. 모니터링 대상으로 Kubernetes 클러스터를 사용하고 해당 클러스터가 사용자 지정 또는 개인 Docker 저장소나 컨테이너 레지스트리에서만 컨테이너 이미지를 가져오도록 구성된 경우 NetApp Kubernetes Monitoring Operator에 필요한 컨테이너에 대한 액세스를 구성해야 합니다.

NetApp Monitoring Operator 설치 타일에서 "이미지 풀 스니펫"을 실행합니다. 이 명령은 Data Infrastructure Insights 저장소에 로그인하고, 운영자에 대한 모든 이미지 종속성을 끌어오고, Data Infrastructure Insights 저장소에서 로그아웃합니다. 메시지가 표시되면 제공된 저장소 임시 비밀번호를 입력하세요. 이 명령은 옵션 기능을 포함하여 운영자가 사용하는 모든 이미지를 다운로드합니다. 이 이미지가 어떤 기능에 사용되는지 아래에서 확인하세요.

핵심 운영자 기능 및 Kubernetes 모니터링

- 넷앱 모니터링
- kube-rbac-프록시
- kube-state-metrics
- 텔레그라프
- distroless-root-user

이벤트 로그

- 유창한 비트
- 쿠버네티스 이벤트 내보내기

네트워크 성능 및 맵

- ci-net-observer

회사 정책에 따라 운영자 Docker 이미지를 개인/로컬/엔터프라이즈 Docker 저장소에 푸시합니다. 저장소에 있는 이미지 태그와 해당 이미지의 디렉토리 경로가 Data Infrastructure Insights 저장소의 이미지 태그와 디렉토리 경로와 일치하는지 확인하세요.

operator-deployment.yaml에서 monitoring-operator 배포를 편집하고 모든 이미지 참조를 수정하여 개인 Docker 저장소를 사용합니다.

```
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.yaml의 AgentConfiguration을 편집하여 새로운 docker repo 위치를 반영합니다. 개인 저장소에 대한 새로운 imagePullSecret을 생성하세요. 자세한 내용은 _https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/_를 참조하세요.

```
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 지침

OpenShift 4.6 이상을 사용하는 경우 *operator-config.yaml_*에서 *AgentConfiguration*을 편집하여 *_runPrivileged* 설정을 활성화해야 합니다.

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

Openshift는 일부 Kubernetes 구성 요소에 대한 액세스를 차단할 수 있는 추가 보안 수준을 구현할 수 있습니다.

권한

모니터링하는 클러스터에 ClusterRole이 없는 사용자 지정 리소스가 포함되어 있는 경우"보기 위한 집계" 이벤트 로그를 통해 이러한 리소스를 모니터링하려면 운영자에게 이러한 리소스에 대한 액세스 권한을 수동으로 부여해야 합니다.

1. 설치 전 _operator-additional-permissions.yaml_을 편집하거나 설치 후 리소스 _ClusterRole/<namespace>-additional-permissions_를 편집합니다.
2. ["get", "watch", "list"] 동사를 사용하여 원하는 apiGroups 및 리소스에 대한 새 규칙을 만듭니다. https://kubernetes.io/docs/reference/access-authn-authz/rbac/ 참조하세요.
3. 클러스터에 변경 사항을 적용합니다.

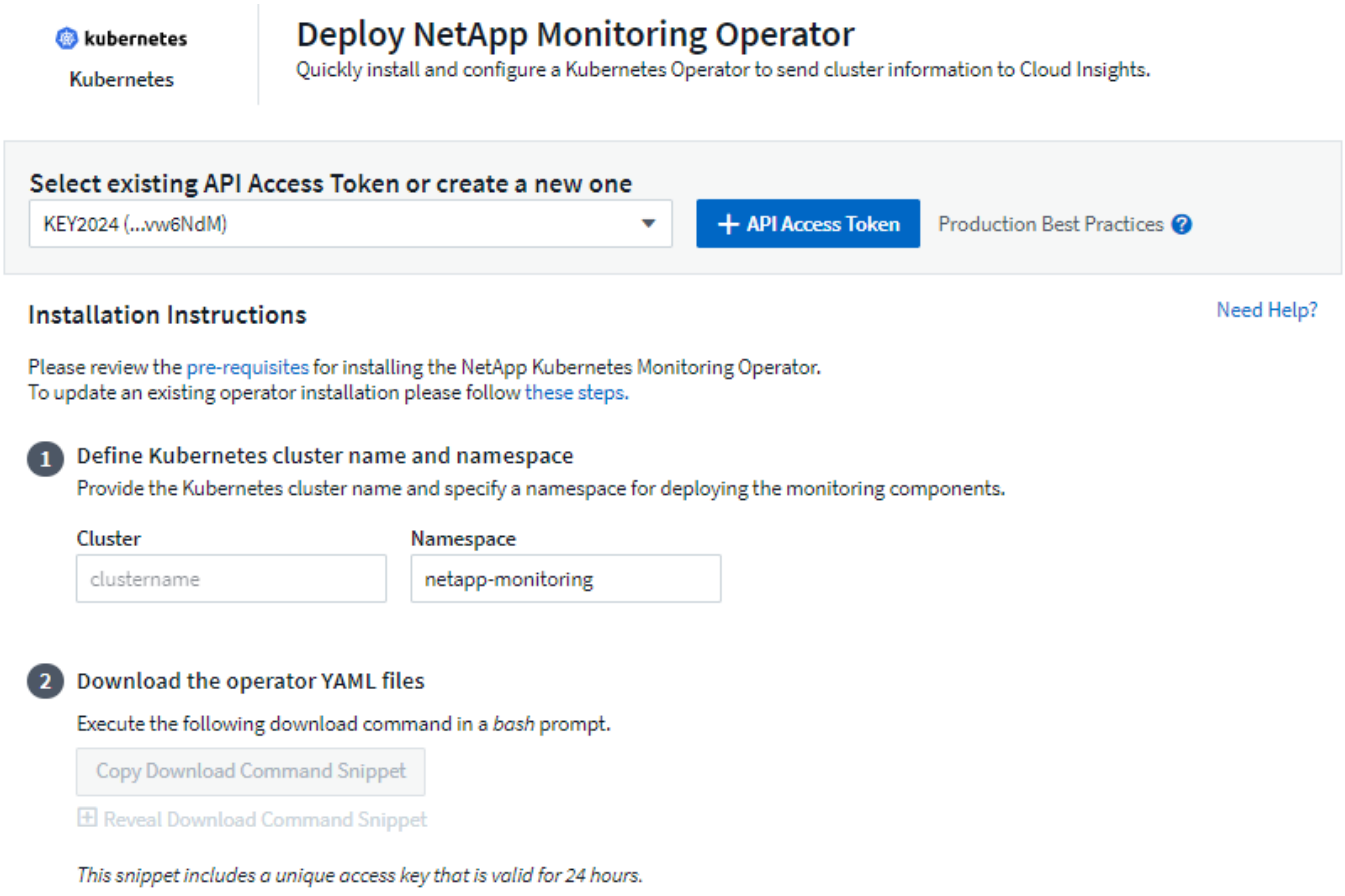
Kubernetes 모니터링 운영자 설치 및 구성

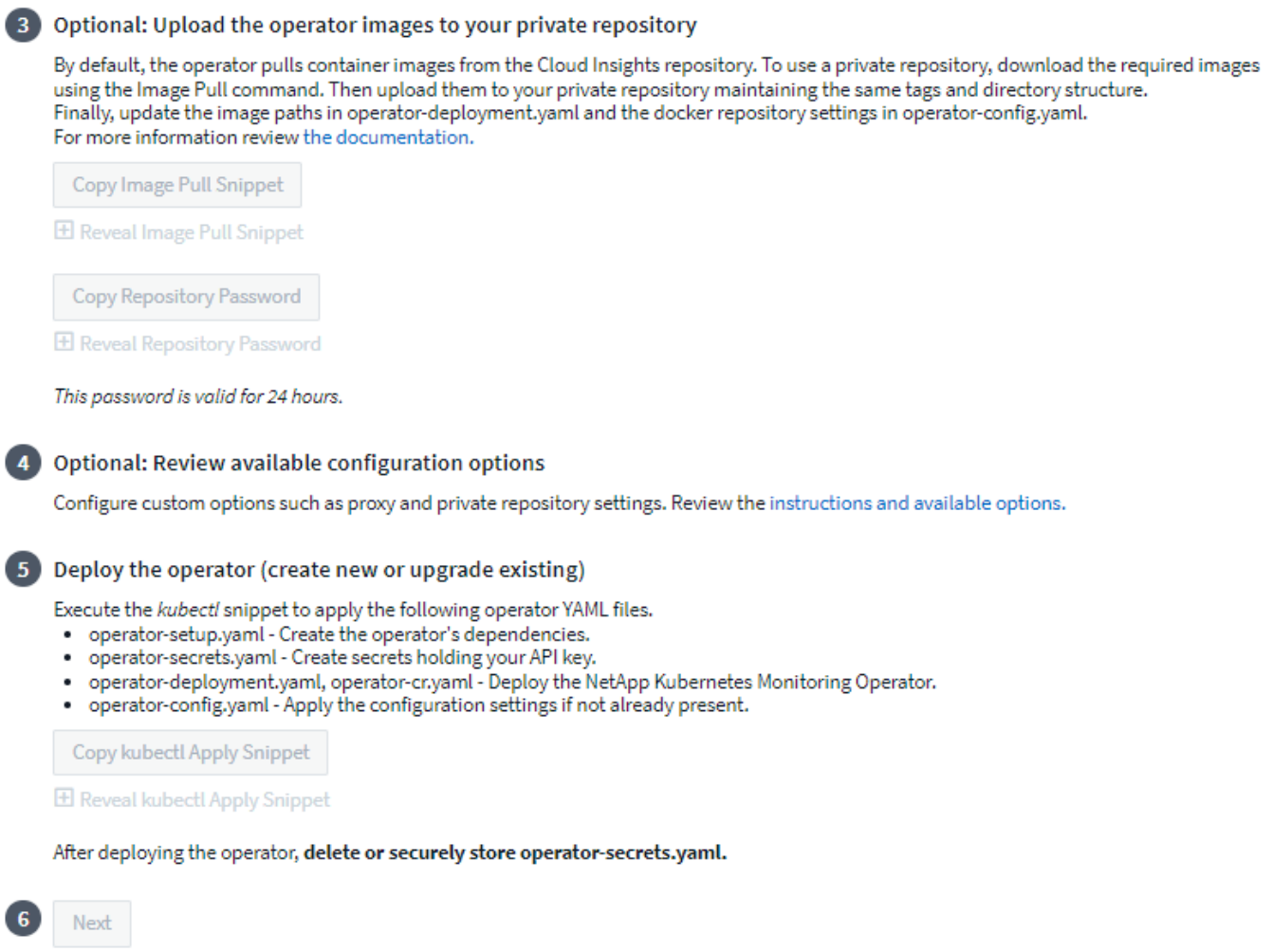
Data Infrastructure Insights Kubernetes 컬렉션을 위한 *Kubernetes Monitoring Operator*를 제공합니다. 새로운 운영자를 배포하려면 *Kubernetes > Collectors > +Kubernetes Collector*로 이동합니다.

Kubernetes Monitoring Operator를 설치하기 전에

를 참조하십시오"필수 조건" Kubernetes Monitoring Operator를 설치하거나 업그레이드하기 전에 설명서를 참조하세요.

Kubernetes 모니터링 운영자 설치





Kubernetes에 **Kubernetes Monitoring Operator** 에이전트를 설치하는 단계:

1. 고유한 클러스터 이름과 네임스페이스를 입력하세요. 만약 당신이라면업그레이드 이전 Kubernetes Operator에서 동일한 클러스터 이름과 네임스페이스를 사용합니다.
2. 이를 입력하면 다운로드 명령 스니펫을 클립보드에 복사할 수 있습니다.
3. 스니펫을 *bash* 창에 붙여넣고 실행합니다. Operator 설치 파일이 다운로드됩니다. 스니펫에는 고유 키가 있으며 24시간 동안 유효합니다.
4. 사용자 정의 또는 개인 저장소가 있는 경우 선택 사항인 이미지 풀 스니펫을 복사하여 *bash* 셸에 붙여넣고 실행합니다. 이미지를 가져온 후 개인 저장소에 복사하세요. 동일한 태그와 폴더 구조를 유지하세요. _operator-deployment.yaml_의 경로와 _operator-config.yaml_의 docker 저장소 설정을 업데이트합니다.
5. 원하는 경우 프록시나 개인 저장소 설정 등 사용 가능한 구성 옵션을 검토하세요. 더 자세히 읽어보세요"구성 옵션" .
6. 준비가 되면 kubectl Apply 스니펫을 복사하고, 다운로드하고, 실행하여 Operator를 배포합니다.
7. 설치가 자동으로 진행됩니다. 완료되면 다음 버튼을 클릭하세요.
8. 설치가 완료되면 다음 버튼을 클릭하세요. *operator-secrets.yaml* 파일도 삭제하거나 안전하게 저장하세요.

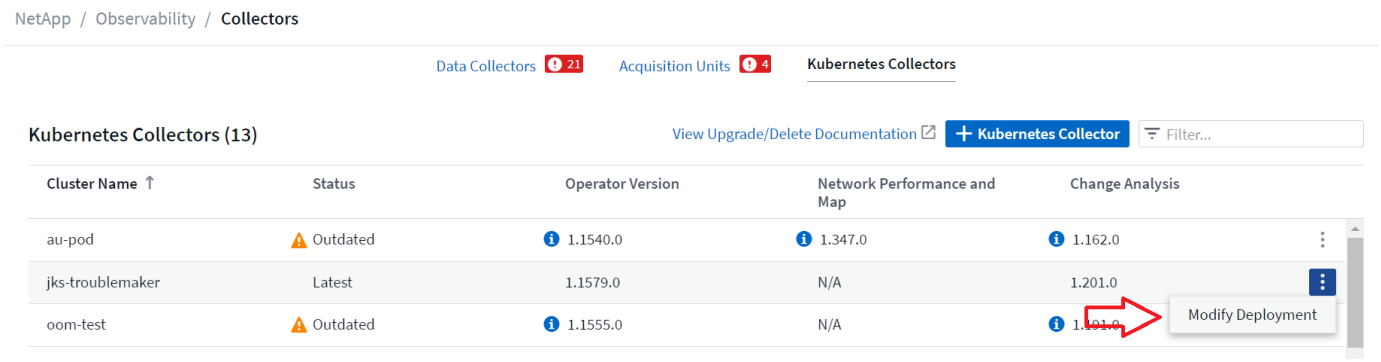
사용자 정의 저장소가 있는 경우 다음을 읽어보세요.사용자 정의/개인 Docker 저장소 사용 .

Kubernetes 모니터링 구성 요소

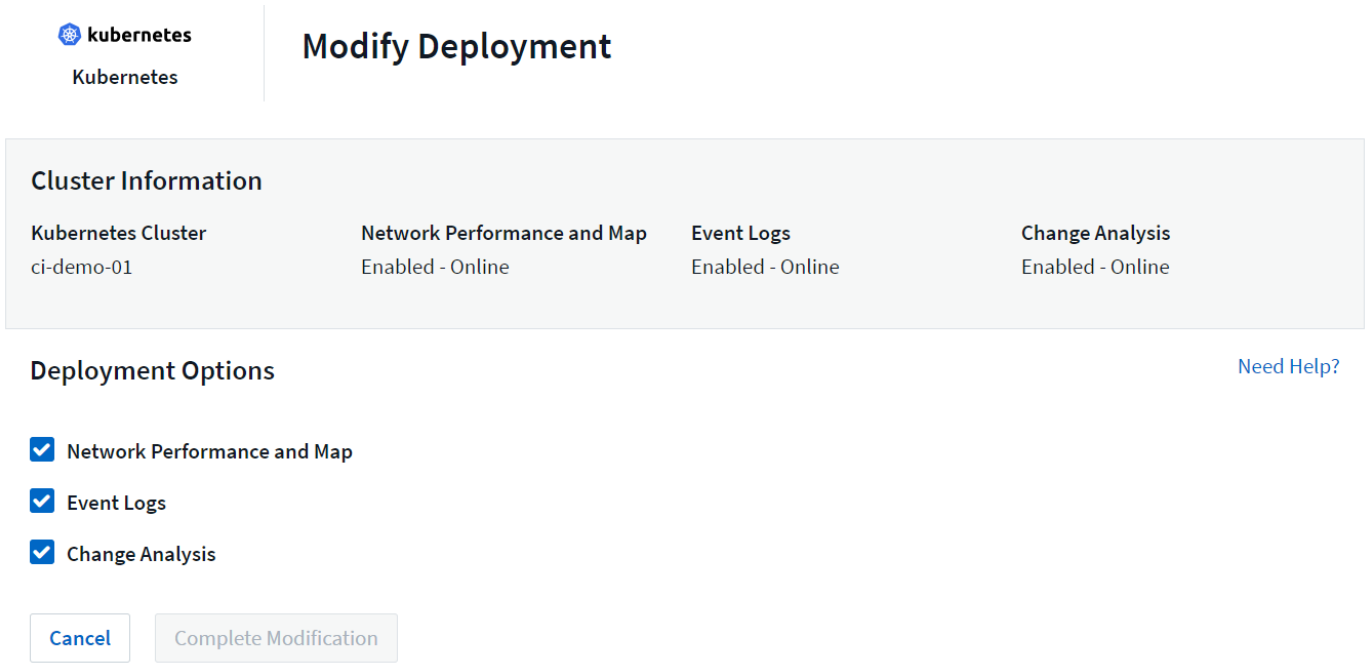
Data Infrastructure Insights Kubernetes Monitoring은 네 가지 모니터링 구성 요소로 구성됩니다.

- 클러스터 메트릭
- 네트워크 성능 및 맵(선택 사항)
- 이벤트 로그(선택 사항)
- 변경 분석(선택 사항)

위의 선택적 구성 요소는 각 Kubernetes 수집기에서 기본적으로 활성화됩니다. 특정 수집기에 대한 구성 요소가 필요하지 않다고 판단되면 *Kubernetes > 수집기*로 이동하여 화면 오른쪽에 있는 수집기의 "세 개의 점" 메뉴에서 _배포 수정_을 선택하여 해당 구성 요소를 비활성화할 수 있습니다.



화면에는 각 구성 요소의 현재 상태가 표시되며 필요에 따라 해당 수집기의 구성 요소를 비활성화하거나 활성화할 수 있습니다.



최신 Kubernetes Monitoring Operator로 업그레이드

DII 푸시 버튼 업그레이드

DII Kubernetes Collectors 페이지를 통해 Kubernetes Monitoring Operator를 업그레이드할 수 있습니다. 업그레이드하려는 클러스터 옆에 있는 메뉴를 클릭하고 _업그레이드_를 선택하세요. 운영자는 이미지 서명을 확인하고, 현재 설치의 스냅샷을 촬영한 후 업그레이드를 수행합니다. 몇 분 안에 운영자 상태가 업그레이드 진행 중에서 최신으로 바뀌는 것을 볼 수 있습니다. 오류가 발생하면 오류 상태를 선택하여 자세한 내용을 확인하고 아래의 푸시 버튼 업그레이드 문제 해결 표를 참조하세요.

개인 저장소를 사용한 푸시 버튼 업그레이드

운영자가 개인 저장소를 사용하도록 구성된 경우 운영자를 실행하는 데 필요한 모든 이미지와 해당 서명이 저장소에서 사용 가능한지 확인하세요. 업그레이드 과정에서 누락된 이미지로 인한 오류가 발생하면 해당 이미지를 저장소에 추가한 후 업그레이드를 다시 시도하세요. 이미지 서명을 저장소에 업로드하려면 다음과 같이 공동 서명 도구를 사용하고 3번 선택 사항에서 지정한 모든 이미지에 대한 서명을 업로드해야 합니다. 운영자 이미지를 개인 저장소에 업로드 > 이미지 풀 스니펫

```
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>
```

이전에 실행 중이던 버전으로 롤백

푸시 버튼 업그레이드 기능을 사용하여 업그레이드한 후 7일 이내에 현재 버전의 운영자를 사용하는 데 어려움이 발생하는 경우, 업그레이드 프로세스 중에 생성된 스냅샷을 사용하여 이전에 실행 중이던 버전으로 다운그레이드할 수 있습니다. 롤백하려는 클러스터 옆에 있는 메뉴를 클릭하고 _롤백_을 선택합니다.

수동 업그레이드

기존 Operator와 함께 _AgentConfiguration_이 존재하는지 확인합니다(네임스페이스가 기본값인 _netapp-monitoring_이 아닌 경우 적절한 네임스페이스로 대체하십시오):

```
 kubectl -n netapp-monitoring get agentconfiguration netapp-ci-monitoring-
configuration
_AgentConfiguration_이 존재하는 경우:
```

- 설치하다기존 연산자보다 최신 연산자가 우선합니다.
  - 당신이 있는지 확인하십시오최신 컨테이너 이미지 가져오기 사용자 정의 저장소를 사용하는 경우.

_AgentConfiguration_이 존재하지 않는 경우:

- Data Infrastructure Insights 에서 인식하는 클러스터 이름을 기록해 두세요(네임스페이스가 기본 netapp-monitoring이 아닌 경우 적절한 네임스페이스로 대체하세요).

```
 kubectl -n netapp-monitoring get agent -o
jsonpath='{.items[0].spec.cluster-name}'
* 기존 Operator의 백업을 만듭니다(네임스페이스가 기본 netapp-monitoring이 아닌 경우
적절한 네임스페이스로 대체).
```

```
 kubectl -n netapp-monitoring get agent -o yaml > agent_backup.yaml
* <<to-remove-the-kubernetes-monitoring-operator,제거>>기존 운영자.
* <<installing-the-kubernetes-monitoring-operator,설치하다>>최신 운영자.
```

- 동일한 클러스터 이름을 사용하세요.
- 최신 Operator YAML 파일을 다운로드한 후 배포하기 전에 _agent_backup.yaml_에서 찾은 모든 사용자 지정 항목을 다운로드한 _operator-config.yaml_로 이식합니다.
- 당신이 있는지 확인하십시오최신 컨테이너 이미지 가져오기 사용자 정의 저장소를 사용하는 경우.

Kubernetes 모니터링 운영자 중지 및 시작

Kubernetes Monitoring Operator를 중지하려면:

```
 kubectl -n netapp-monitoring scale deploy monitoring-operator
--replicas=0
Kubernetes Monitoring Operator를 시작하려면:
```

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

제거 중

Kubernetes Monitoring Operator를 제거하려면

Kubernetes Monitoring Operator의 기본 네임스페이스는 "netapp-monitoring"입니다. 고유한 네임스페이스를 설정한 경우 이 명령과 이후의 모든 명령 및 파일에서 해당 네임스페이스를 대체합니다.

다음 명령을 사용하여 모니터링 운영자의 최신 버전을 제거할 수 있습니다.

```
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>
```

모니터링 운영자가 자체 전용 네임스페이스에 배포된 경우 네임스페이스를 삭제합니다.

```
 kubectl delete ns <NAMESPACE>
```
참고: 첫 번째 명령에서 "리소스를 찾을 수 없습니다"라는 메시지가 반환되면 다음 지침에 따라 이전 버전의 모니터링 운영자를 제거하세요.

다음 명령을 순서대로 실행하세요. 현재 설치 환경에 따라 일부 명령은 '개체를 찾을 수 없습니다'라는 메시지를 반환할 수 있습니다. 이런 메시지는 무시해도 됩니다.

```
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>
```

이전에 보안 컨텍스트 제약 조건이 생성된 경우:

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

Kube-state-metrics에 대하여

NetApp Kubernetes Monitoring Operator는 다른 인스턴스와의 충돌을 피하기 위해 자체 kube-state-metrics를 설치합니다.

Kube-State-Metrics에 대한 정보는 다음을 참조하세요."이 페이지" .

운영자 구성/사용자 정의

이 섹션에는 운영자 구성 사용자 정의, 프록시 작업, 사용자 정의 또는 개인 Docker 저장소 사용, OpenShift 작업 등에 대한 정보가 포함되어 있습니다.

구성 옵션

가장 일반적으로 수정되는 설정은 *AgentConfiguration* 사용자 정의 리소스에서 구성할 수 있습니다. *operator-config.yaml* 파일을 편집하여 운영자를 배포하기 전에 이 리소스를 편집할 수 있습니다. 이 파일에는 주석 처리된 설정 예가 포함되어 있습니다. 목록을 확인하세요"사용 가능한 설정" 최신 버전의 연산자에 대해서.

다음 명령을 사용하여 운영자가 배포된 후에도 이 리소스를 편집할 수 있습니다.

```
 kubectl -n netapp-monitoring edit AgentConfiguration
```
배포된 운영자 버전이 _AgentConfiguration_을 지원하는지 확인하려면 다음 명령을 실행하십시오:

```
 kubectl get crd agentconfigurations.monitoring.netapp.com
```
"서버 오류(찾을 수 없음)" 메시지가 표시되면 AgentConfiguration을 사용하려면 먼저 운영자를 업그레이드해야 합니다.

프록시 지원 구성

테넌트에 프록시를 사용하여 Kubernetes Monitoring Operator를 설치할 수 있는 두 곳이 있습니다. 이는 동일하거나 별도의 프록시 시스템일 수 있습니다.

- 설치 코드 조각을 실행하는 동안 필요한 프록시("curl" 사용)는 조각이 실행되는 시스템을 Data Infrastructure Insights 환경에 연결합니다.
- 대상 Kubernetes 클러스터가 Data Infrastructure Insights 환경과 통신하는 데 필요한 프록시

이 두 가지 중 하나 또는 둘 다에 프록시를 사용하는 경우 Kubernetes Operating Monitor를 설치하려면 먼저 프록시가 Data Infrastructure Insights 환경과의 원활한 통신을 허용하도록 구성되어 있는지 확인해야 합니다. 프록시가 있고 Operator를 설치하려는 서버/VM에서 Data Infrastructure Insights 에 액세스할 수 있는 경우 프록시가 올바르게 구성된 것일 가능성이 높습니다.

Kubernetes Operating Monitor를 설치하는 데 사용되는 프록시의 경우, Operator를 설치하기 전에 *http_proxy/https_proxy* 환경 변수를 설정하세요. 일부 프록시 환경에서는 _no_proxy 환경 변수를 설정해야 할 수도 있습니다.

변수를 설정하려면 Kubernetes Monitoring Operator를 설치하기 전에 시스템에서 다음 단계를 수행하세요.

1. 현재 사용자에 대해 *https_proxy* 및/또는 *http_proxy* 환경 변수를 설정합니다.
   a. 설정 중인 프록시에 인증(사용자 이름/비밀번호)이 없는 경우 다음 명령을 실행합니다.

```
 export https_proxy=<proxy_server>:<proxy_port>
```
.. 설정 중인 프록시에 인증(사용자 이름/비밀번호)이 있는 경우 다음 명령을 실행하세요.

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

Kubernetes 클러스터가 Data Infrastructure Insights 환경과 통신하는 데 사용되는 프록시의 경우, 이 지침을 모두 읽은 후 Kubernetes Monitoring Operator를 설치하세요.

Kubernetes 모니터링 오퍼레이터를 배포하기 전에 *operator-config.yaml_*의 *_AgentConfiguration* 프록시 섹션을 구성하십시오.

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

사용자 정의 또는 개인 **Docker** 저장소 사용

기본적으로 Kubernetes Monitoring Operator는 Data Infrastructure Insights 저장소에서 컨테이너 이미지를 가져옵니다. 모니터링 대상으로 Kubernetes 클러스터를 사용하고 해당 클러스터가 사용자 정의 또는 개인 Docker 저장소나 컨테이너 레지스트리에서만 컨테이너 이미지를 가져오도록 구성된 경우 Kubernetes Monitoring Operator에 필요한 컨테이너에 대한 액세스를 구성해야 합니다.

NetApp Monitoring Operator 설치 타일에서 "이미지 풀 스니펫"을 실행합니다. 이 명령은 Data Infrastructure Insights 저장소에 로그인하고, 운영자에 대한 모든 이미지 종속성을 끌어오고, Data Infrastructure Insights 저장소에서 로그아웃합니다. 메시지가 표시되면 제공된 저장소 임시 비밀번호를 입력하세요. 이 명령은 옵션 기능을 포함하여 운영자가 사용하는 모든 이미지를 다운로드합니다. 이 이미지가 어떤 기능에 사용되는지 아래에서 확인하세요.

핵심 운영자 기능 및 Kubernetes 모니터링

- 넷앱 모니터링
- ci-kube-rbac-프록시
- ci-ksm
- ci-텔레그라프
- distroless-root-user

이벤트 로그

- ci-fluent-bit
- ci-kubernetes-이벤트-내보내기

네트워크 성능 및 맵

- ci-net-observer

회사 정책에 따라 운영자 Docker 이미지를 개인/로컬/엔터프라이즈 Docker 저장소에 푸시합니다. 저장소에 있는 이미지 태그와 해당 이미지의 디렉토리 경로가 Data Infrastructure Insights 저장소의 이미지 태그와 디렉토리 경로와 일치하는지 확인하세요.

operator-deployment.yaml에서 monitoring-operator 배포를 편집하고 모든 이미지 참조를 수정하여 개인 Docker 저장소를 사용합니다.

```
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_의 _AgentConfiguration_을 편집하여 새 docker 리포지토리 위치를 반영하세요. 개인 리포지토리에 대한 새 imagePullSecret을 생성하세요. 자세한 내용은 _https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/_를 참조하세요.

```
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
```

장기 비밀번호용 **API** 액세스 토큰

일부 환경(예: 프록시 저장소)에는 Data Infrastructure Insights docker 저장소에 대한 장기 암호가 필요합니다. 설치 시 UI에서 제공되는 암호는 24시간 동안만 유효합니다. 이 암호 대신 API 액세스 토큰을 docker 저장소 암호로 사용할 수 있습니다. 이 암호는 API 액세스 토큰이 유효한 동안 유효합니다. 이 용도로 새 API 액세스 토큰을 생성하거나 기존 토큰을 사용할 수 있습니다.

"여기를 읽어보세요" 새 API 액세스 토큰 생성 지침을 참조하십시오.

다운로드한 *operator-secrets.yaml* 파일에서 기존 API 액세스 토큰을 추출하려면 사용자는 다음을 실행할 수 있습니다.

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

실행 중인 오퍼레이터 설치에서 기존 API Access Token을 추출하려면 다음 명령을 실행하면 됩니다.

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

OpenShift 지침

OpenShift 4.6 이상 버전을 사용하는 경우, *operator-config.yaml* 파일의 *AgentConfiguration* 설정을 수정하여 *runPrivileged* 설정을 활성화해야 합니다.

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

Openshift는 일부 Kubernetes 구성 요소에 대한 액세스를 차단할 수 있는 추가 보안 수준을 구현할 수 있습니다.

관용과 오염

netapp-ci-telegraf-ds, *netapp-ci-fluent-bit-ds*, 및 *netapp-ci-net-observer-l4-ds* DaemonSets는 모든 노드에서 데이터를 올바르게 수집하기 위해 클러스터의 모든 노드에 Pod를 예약해야 합니다. 해당 운영자는 잘 알려진 몇 가지 *오염*을 허용하도록 구성되었습니다. 노드에서 사용자 정의 오염을 구성하여 모든 노드에서 포드가 실행되지 않도록 한 경우 해당 오염에 대한 *허용*을 생성할 수 있습니다."_AgentConfiguration_에서" . 클러스터의 모든 노드에 사용자 정의 테인을 적용한 경우 운영자 포드를 예약하고 실행할 수 있도록 운영자 배포에 필요한 허용 범위도 추가해야 합니다.

Kubernetes에 대해 자세히 알아보기"오염과 관용" .

로 돌아가기"* NetApp Kubernetes 모니터링 운영자 설치* 페이지"

비밀에 대한 참고 사항

Kubernetes Monitoring Operator가 클러스터 전체의 비밀을 볼 수 있는 권한을 제거하려면 설치하기 전에 *operator-setup.yaml* 파일에서 다음 리소스를 삭제하세요.

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

업그레이드인 경우 클러스터에서 리소스도 삭제하세요.

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

변경 분석이 활성화된 경우 *AgentConfiguration* 또는 _operator-config.yaml_을 수정하여 변경 관리 섹션의 주석 처리를 제거하고 변경 관리 섹션 아래에 _kindsToIgnoreFromWatch: '"secrets"'_를 포함합니다. 이 줄에서 작은따옴표와 큰따옴표의 존재와 위치에 주목하세요.

```
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 모니터링 운영자 이미지 서명 확인

운영자의 이미지와 배포하는 모든 관련 이미지는 NetApp 에서 서명합니다. cosign 도구를 사용하여 설치 전에 이미지를 수동으로 검증하거나 Kubernetes 입장 컨트롤러를 구성할 수 있습니다. 자세한 내용은 다음을 참조하세요."쿠버네티스 문서" .

이미지 서명을 확인하는 데 사용되는 공개 키는 선택 사항*:* 운영자 이미지를 개인 저장소에 업로드 > 이미지 서명 공개 키 아래의 모니터링 운영자 설치 타일에서 사용할 수 있습니다.

이미지 서명을 수동으로 확인하려면 다음 단계를 수행하세요.

1. 이미지 풀 스니펫을 복사하여 실행하세요.
2. 메시지가 표시되면 저장소 비밀번호를 복사하여 입력하세요.
3. 이미지 서명 공개 키(예시에서는 dii-image-signing.pub)를 저장합니다.
4. 공동 서명을 사용하여 이미지를 확인하세요. 다음은 공동 서명 사용의 예입니다.

```
$ 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}]
```

문제 해결

Kubernetes Monitoring Operator를 설정하는 데 문제가 발생하면 다음을 시도해 보세요.

문제:	다음을 시도해 보세요:
Kubernetes 영구 볼륨과 해당 백엔드 스토리지 장치 사이에 하이퍼링크/연결이 보이지 않습니다. 내 Kubernetes 영구 볼륨은 스토리지 서버의 호스트 이름을 사용하여 구성됩니다.	기존 Telegraf 에이전트를 제거하는 단계를 따른 다음, 최신 Telegraf 에이전트를 다시 설치합니다. Telegraf 버전 2.0 이상을 사용해야 하며, Kubernetes 클러스터 스토리지는 Data Infrastructure Insights 에서 적극적으로 모니터링되어야 합니다.
로그에서 다음과 유사한 메시지가 표시됩니다. E0901 15:21:39.962145 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: *v1.MutatingWebhookConfiguration을 나열하는 데 실패했습니다. 서버가 요청한 리소스를 찾을 수 없습니다. E0901 15:21:43.168161 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: *v1.Lease를 나열하는 데 실패했습니다. 서버가 요청한 리소스를 찾을 수 없습니다(get leases.coordination.k8s.io) 등.	Kubernetes 버전이 1.20 미만인 경우 kube-state-metrics 버전 2.0.0 이상을 실행하는 경우 이러한 메시지가 나타날 수 있습니다. Kubernetes 버전을 가져오려면: *kubectl version* kube-state-metrics 버전을 가져오려면: *kubectl get deploy/kube-state-metrics -o jsonpath='{..image}'* 이러한 메시지가 발생하지 않도록 하려면 사용자는 kube-state-metrics 배포를 수정하여 다음 임대를 비활성화할 수 있습니다. *mutatingwebhookconfigurations validatingwebhookconfigurations volumeattachments resources* 보다 구체적으로 다음 CLI 인수를 사용할 수 있습니다. 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 기본 리소스 목록은 다음과 같습니다. "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"

문제:	다음을 시도해 보세요:
Telegraf에서 다음과 유사한 오류 메시지가 표시되지만 Telegraf는 시작되고 실행됩니다. 10월 11일 14:23:41 ip-172-31-39-47 systemd[1]: InfluxDB에 메트릭을 보고하기 위한 플러그인 기반 서버 에이전트가 시작되었습니다. 10월 11일 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="캐시 디렉토리를 생성하지 못했습니다. /etc/telegraf/.cache/snowflake, err: mkdir /etc/telegraf/.ca che: 권한이 거부되었습니다. 무시되었습니다.\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" 10월 11일 14:23:41 ip-172-31-39-47 telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="열지 못했습니다. 무시됨. open /etc/telegraf/.cache/snowflake/ocsp_response_cache.json: 해당 파일이나 디렉토리가 없습니다.\n" func="gosnowflake.(*defaultLogger).Errorf" file="log.go:120" 10월 11일 14:23:41 ip-172-31-39-47 telegraf[1827]: 2021-10-11T14:23:41Z 나! Telegraf 1.19.3 시작하기	이는 알려진 문제입니다. 참조하다"이 GitHub 문서" 자세한 내용은. Telegraf가 실행되는 동안 사용자는 이러한 오류 메시지를 무시할 수 있습니다.
Kubernetes에서 Telegraf 포드가 다음 오류를 보고합니다. "마운트 통계 정보 처리 중 오류 발생: 마운트 통계 파일(/hostfs/proc/1/mountstats)을 열 수 없습니다. 오류: /hostfs/proc/1/mountstats를 엽니다. 권한이 거부되었습니다."	SELinux가 활성화되어 있고 적용되어 있는 경우 Telegraf 포드가 Kubernetes 노드의 /proc/1/mountstats 파일에 액세스하지 못할 가능성이 높습니다. 이러한 제한을 극복하려면 에이전트 구성을 편집하고 runPrivileged 설정을 활성화하세요. 자세한 내용은 OpenShift 지침을 참조하세요.
Kubernetes에서 Telegraf ReplicaSet 포드가 다음 오류를 보고합니다. [inputs.prometheus] 플러그인 오류: 키 쌍 /etc/kubernetes/pki/etcd/server.crt:/etc/kubernetes/pki/etcd/server.key를 로드할 수 없습니다. /etc/kubernetes/pki/etcd/server.crt를 엽니다. 해당 파일이나 디렉토리가 없습니다.	Telegraf ReplicaSet 포드는 마스터 또는 etcd로 지정된 노드에서 실행되도록 설계되었습니다. 이러한 노드 중 하나에서 ReplicaSet 포드가 실행되고 있지 않으면 이러한 오류가 발생합니다. 마스터/etcd 노드에 오염이 있는지 확인하세요. 그렇다면 Telegraf ReplicaSet, telegraf-rs에 필요한 허용 범위를 추가합니다. 예를 들어, ReplicaSet을 편집합니다… kubectl edit rs telegraf-rs …그리고 사양에 적절한 허용 범위를 추가합니다. 그런 다음 ReplicaSet 포드를 다시 시작합니다.
저는 PSP/PSA 환경을 사용하고 있습니다. 이것이 모니터링 운영자에게 영향을 미칩니까?	Kubernetes 클러스터가 Pod 보안 정책(PSP) 또는 Pod 보안 승인(PSA)을 적용하여 실행되는 경우 최신 Kubernetes 모니터링 운영자로 업그레이드해야 합니다. PSP/PSA를 지원하는 현재 운영자로 업그레이드하려면 다음 단계를 따르세요. 1. 제거 이전 모니터링 연산자: 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. 설치하다 모니터링 운영자의 최신 버전입니다.

문제:	다음을 시도해 보세요:
Operator를 배포하려고 하다가 문제가 발생했고, PSP/PSA를 사용 중입니다.	1. 다음 명령을 사용하여 에이전트를 편집합니다: kubectl -n <네임스페이스> edit agent 2. 'security-policy-enabled'를 'false'로 표시합니다. 이렇게 하면 Pod 보안 정책과 Pod 보안 입장이 비활성화되고 운영자가 배포할 수 있습니다. 다음 명령을 사용하여 확인하세요. kubectl get psp(Pod 보안 정책이 제거되었음을 표시해야 함) kubectl get all -n <네임스페이스>
grep -i psp(아무것도 발견되지 않았음을 표시해야 함)	"ImagePullBackoff" 오류가 발생했습니다.
이러한 오류는 사용자 지정 또는 개인 Docker 저장소가 있고 Kubernetes Monitoring Operator가 이를 올바르게 인식하도록 아직 구성하지 않은 경우 나타날 수 있습니다. 더 읽어보세요 사용자 정의/개인 저장소 구성에 대한 정보입니다.	모니터링 운영자 배포에 문제가 있는데, 현재 문서에서는 이를 해결하는 데 도움이 되지 않습니다.

다음 명령의 출력을 캡처하거나 기록해 두고 기술 지원팀에 문의하세요.
`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 네임스페이스의 net-observer(워크로드 맵) 포드는 CrashLoopBackOff에 있습니다.

이러한 포드는 네트워크 관찰을 위한 워크로드 맵 데이터 수집기에 해당합니다. 다음을 시도해 보세요. • 포드 중 하나의 로그를 확인하여 최소 커널 버전을 확인하세요. 예: ---- {"ci-tenant-id":"your-tenant-id","collector-cluster":"your-k8s-cluster-name","environment":"prod","level":"error","msg":"유효성 검사에 실패했습니다. 이유: 커널 버전 3.10.0은 최소 커널 버전 4.18.0보다 낮습니다.","time":"2022-11-09T08:23:08Z"} ---- • Net-observer 포드에는 Linux 커널 버전이 최소 4.18.0이어야 합니다. "uname -r" 명령을 사용하여 커널 버전을 확인하고 버전이 4.18.0 이상인지 확인하세요.	Pod는 Operator 네임스페이스(기본값: netapp-monitoring)에서 실행되지만 쿼리의 워크로드 맵이나 Kubernetes 메트릭에 대한 데이터가 UI에 표시되지 않습니다.
K8S 클러스터의 노드에서 시간 설정을 확인하세요. 정확한 감사 및 데이터 보고를 위해서는 NTP(Network Time Protocol) 또는 SNTP(Simple Network Time Protocol)를 사용하여 에이전트 컴퓨터의 시간을 동기화하는 것이 좋습니다.	Operator 네임스페이스의 일부 net-observer 포드가 보류 상태입니다.

문제:	다음을 시도해 보세요:
Net-observer는 DaemonSet이며 k8s 클러스터의 각 노드에서 Pod를 실행합니다. • 보류 상태인 포드를 확인하고 CPU 또는 메모리 리소스 문제가 발생하는지 확인하세요. 노드에서 필요한 메모리와 CPU를 사용할 수 있는지 확인하세요.	Kubernetes Monitoring Operator를 설치한 직후 로그에 다음과 같은 내용이 표시됩니다. [inputs.prometheus] 플러그인 오류: http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics에 대한 HTTP 요청을 만드는 중 오류가 발생했습니다. http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics를 가져옵니다. tcp 다이얼: kube-state-metrics.<namespace>.svc.cluster.local을 조회합니다. 해당 호스트가 없습니다.
이 메시지는 일반적으로 새로운 운영자가 설치되고 *ksm* 포드가 작동하기 전에 *telegraf-rs* 포드가 작동할 때만 나타납니다. 모든 포드가 실행되면 이러한 메시지는 더 이상 표시되지 않습니다.	내 클러스터에 있는 Kubernetes CronJob에 대해 수집된 메트릭이 보이지 않습니다.
Kubernetes 버전을 확인하세요(예: `kubectl version`). v1.20.x 이하인 경우 이는 예상되는 제한 사항입니다. Kubernetes Monitoring Operator와 함께 배포된 kube-state-metrics 릴리스는 v1.CronJob만 지원합니다. Kubernetes 1.20.x 이하에서는 CronJob 리소스가 v1beta.CronJob에 있습니다. 결과적으로 kube-state-metrics는 CronJob 리소스를 찾을 수 없습니다.	운영자를 설치한 후, telegraf-ds 포드가 CrashLoopBackOff에 진입하고 포드 로그에 "su: 인증 실패"가 표시됩니다.
_AgentConfiguration_에서 telegraf 섹션을 편집하고, _dockerMetricCollectionEnabled_를 false로 설정하세요. 자세한 내용은 operator의 "구성 옵션"를 참조하세요. … spec: … telegraf: … - name: docker run-mode: - DaemonSet substitutions: - key: DOCKER_UNIX_SOCK_PLACEHOLDER value: unix:///run/docker.sock … …	Telegraf 로그에서 다음과 유사한 오류 메시지가 반복해서 나타납니다. E! [에이전트] outputs.http에 쓰는 중 오류가 발생했습니다. 게시물 "https://<tenant_url>/rest/v1/lake/ingest/influxdb": 컨텍스트 마감일이 초과되었습니다(헤더를 기다리는 동안 Client.Timeout이 초과되었습니다).
_AgentConfiguration_의 telegraf 섹션을 편집하고 _outputTimeout_을 10초로 늘립니다. 자세한 내용은 운영자에게 문의하세요."구성 옵션" .	일부 이벤트 로그에 대한 *involvedobject* 데이터가 없습니다.
다음 단계를 따랐는지 확인하세요."권한" 위 섹션 참조.	두 개의 모니터링 운영자 포드가 실행 중인 것을 보는 이유는 무엇입니까? 하나는 netapp-ci-monitoring-operator-<pod>이고 다른 하나는 monitoring-operator-<pod>입니다.
2023년 10월 12일부터 Data Infrastructure Insights 사용자에게 더 나은 서비스를 제공하기 위해 운영자를 리팩토링했습니다. 이러한 변경 사항을 완전히 적용하려면 다음을 수행해야 합니다.이전 연산자를 제거하세요 그리고새로운 것을 설치하다 .	내 Kubernetes 이벤트가 예기치 않게 Data Infrastructure Insights 에 보고를 중단했습니다.
이벤트 내보내기 포드의 이름을 검색합니다. `kubectl -n netapp-monitoring get pods`	grep event-exporter

문제:	다음을 시도해 보세요:
awk '{print $1}'	sed 's/event-exporter./event-exporter/'` "netapp-ci-event-exporter" 또는 "event-exporter"여야 합니다. 다음으로 모니터링 에이전트를 편집합니다. `kubectl -n netapp-monitoring edit agent`, 그리고 LOG_FILE의 값을 이전 단계에서 찾은 적절한 이벤트 내보내기 포드 이름을 반영하도록 설정합니다. 보다 구체적으로, LOG_FILE은 "/var/log/containers/netapp-ci-event-exporter.log" 또는 "/var/log/containers/event-exporter*.log"로 설정되어야 합니다. …. fluent-bit: … - name: event-exporter-ci substitutions: - key: LOG_FILE values: - /var/log/containers/netapp-ci-event-exporter*.log … …. 또는 다음도 가능합니다.제거 그리고다시 설치하다 대리인.
Kubernetes Monitoring Operator가 배포한 Pod가 리소스가 부족하여 충돌하는 현상이 발생합니다.	Kubernetes Monitoring Operator를 참조하세요."구성 옵션" 필요에 따라 CPU 및/또는 메모리 한도를 늘립니다.
이미지가 누락되었거나 구성이 잘못되어 netapp-ci-kube-state-metrics 포드가 시작되지 않거나 준비되지 않았습니다. 이제 StatefulSet이 멈춰 있고 구성 변경 사항이 netapp-ci-kube-state-metrics 포드에 적용되지 않습니다.	StatefulSet은 다음과 같습니다."고장난" 상태. 모든 구성 문제를 해결한 후 netapp-ci-kube-state-metrics 포드를 반송합니다.
Kubernetes Operator 업그레이드를 실행한 후 netapp-ci-kube-state-metrics 포드가 시작되지 않고 ErrImagePull(이미지를 가져오는 데 실패) 오류가 발생합니다.	포드를 수동으로 재설정해보세요.
Kubernetes 클러스터의 로그 분석에서 "maxEventAgeSeconds보다 오래되어 이벤트가 삭제되었습니다"라는 메시지가 관찰되었습니다.	Operator *agentconfiguration_*을 수정하고 *_event-exporter-maxEventAgeSeconds*(즉, 60초), *event-exporter-kubeQPS*(즉, 100), *event-exporter-kubeBurst*(즉, 500)를 늘립니다. 이러한 구성 옵션에 대한 자세한 내용은 다음을 참조하세요."구성 옵션" 페이지.
Telegraf는 잠글 수 있는 메모리가 부족하여 경고하거나 충돌합니다.	기본 운영 체제/노드에서 Telegraf의 잠금 가능 메모리 한도를 늘려보세요. 한도를 늘리는 것이 불가능한 경우 NKMO 에이전트 구성을 수정하고 *unprotected_*를 *_true_*로 설정하세요. 이렇게 하면 *Telegraf*는 잠긴 메모리 페이지를 예약하지 않습니다. 복호화된 비밀이 디스크로 옮겨갈 수 있으므로 보안 위험이 발생할 수 있지만, 잠긴 메모리를 예약할 수 없는 환경에서 실행할 수 있습니다. _보호되지 않은 구성 옵션에 대한 자세한 내용은 다음을 참조하세요."구성 옵션" 페이지.

문제:	다음을 시도해 보세요:
Telegraf에서 다음과 유사한 경고 메시지를 보았습니다: _W! [inputs.diskio] "vdc"에 대한 디스크 이름을 수집할 수 없습니다. /dev/vdc를 읽는 중 오류가 발생했습니다. 해당 파일이나 디렉토리가 없습니다.	Kubernetes 모니터링 오퍼레이터의 경우 이러한 경고 메시지는 무해하며 무시해도 됩니다. 또는 AgentConfiguration에서 telegraf 섹션을 편집하고 _runDsPrivileged_를 true로 설정하십시오. 자세한 내용은 "운영자 구성 옵션"을(를) 참조하십시오.

내 fluent-bit pod가 다음 오류로 인해 실패하고 있습니다. [2024/10/16 14:16:23] [오류] [/src/fluent-bit/plugins/in_tail/tail_fs_inotify.c:360 errno=24] 열려 있는 파일이 너무 많습니다. [2024/10/16 14:16:23] [오류] 입력 tail.0을 초기화하지 못했습니다. [2024/10/16 14:16:23] [오류] [엔진] 입력 초기화에 실패했습니다.	클러스터에서 *fsnotify* 설정을 변경해보세요.
	`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-bit를 다시 시작합니다.
	참고: 노드 재시작 시에도 이러한 설정을 유지하려면 _/etc/sysctl.conf_에 다음 줄을 넣어야 합니다.
	`fs.inotify.max_user_instances=<something larger than current setting>`
	`fs.inotify.max_user_watches=<something larger than current setting>`

문제:	다음을 시도해 보세요:
Telegraf DS Pod는 TLS 인증서의 유효성을 검사할 수 없어 Kubernetes 입력 플러그인이 HTTP 요청을 수행하지 못한다는 오류를 보고하고 있습니다. 예를 들어: E! [inputs.kubernetes] 플러그인 오류: HTTP 요청을 만드는 중 오류가 발생했습니다."<a href="https://<kubelet_IP>:10250/stats/summary":" class="bare">https://<kubelet_IP>:10250/stats/summary": 얻다"<a href="https://<kubelet_IP>:10250/stats/summary":" class="bare">https://<kubelet_IP>:10250/stats/summary": tls: 인증서 확인에 실패했습니다: x509: IP SAN이 포함되어 있지 않으므로 <kubelet_IP>에 대한 인증서를 확인할 수 없습니다.	이는 kubelet이 자체 서명된 인증서를 사용하거나 지정된 인증서에 인증서 *Subject Alternative Name* 목록에 <kubelet_IP>가 포함되지 않은 경우 발생합니다. 이를 해결하려면 사용자가 다음을 수정할 수 있습니다. "에이전트 구성" , _telegraf:insecureK8sSkipVerify_를 _true_로 설정합니다. 이렇게 하면 Telegraf 입력 플러그인이 검증을 건너뛰도록 구성됩니다. 또는 사용자는 kubelet을 구성할 수 있습니다."서버TLS부트스트랩" 그러면 'certificates.k8s.io' API에서 인증서 요청이 트리거됩니다.
Fluent-bit 포드에서 다음과 같은 오류가 발생하고 포드를 시작할 수 없습니다: 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 파일이 있는 호스트 디렉터리에 적절한 읽기/쓰기 권한이 있는지 확인하십시오. 특히, 호스트 디렉터리는 루트가 아닌 사용자에게 읽기/쓰기 권한을 부여해야 합니다. 기본 DB 파일 위치는 fluent-bit-dbFile *agentconfiguration* 옵션으로 재정의하지 않는 한 /var/log/입니다. SELinux가 활성화된 경우 fluent-bit-seLinuxOptionsType agentconfiguration 옵션을 'spc_t'로 설정해 보십시오.

추가 정보는 다음에서 찾을 수 있습니다."지원하다" 페이지 또는"데이터 수집기 지원 매트릭스" .

Kubernetes 모니터링 운영자 구성 옵션

그만큼 "쿠버네티스 모니터링 운영자" AgentConfiguration 파일을 통해 광범위한 사용자 지정 옵션을 제공합니다. 리소스 제한, 수집 간격, 프록시 설정, 허용 오차 및 구성 요소별 설정을 구성하여 Kubernetes 환경의 모니터링을 최적화할 수 있습니다. 이러한 옵션을 사용하여 telegraf, kube-state-metrics, 로그 수집, 워크로드 매핑, 변경 관리 및 기타 모니터링 구성 요소를 사용자 지정할 수 있습니다.

샘플 AgentConfiguration 파일

아래는 각 옵션에 대한 설명이 포함된 *AgentConfiguration* 파일의 예시입니다.

```
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)
```

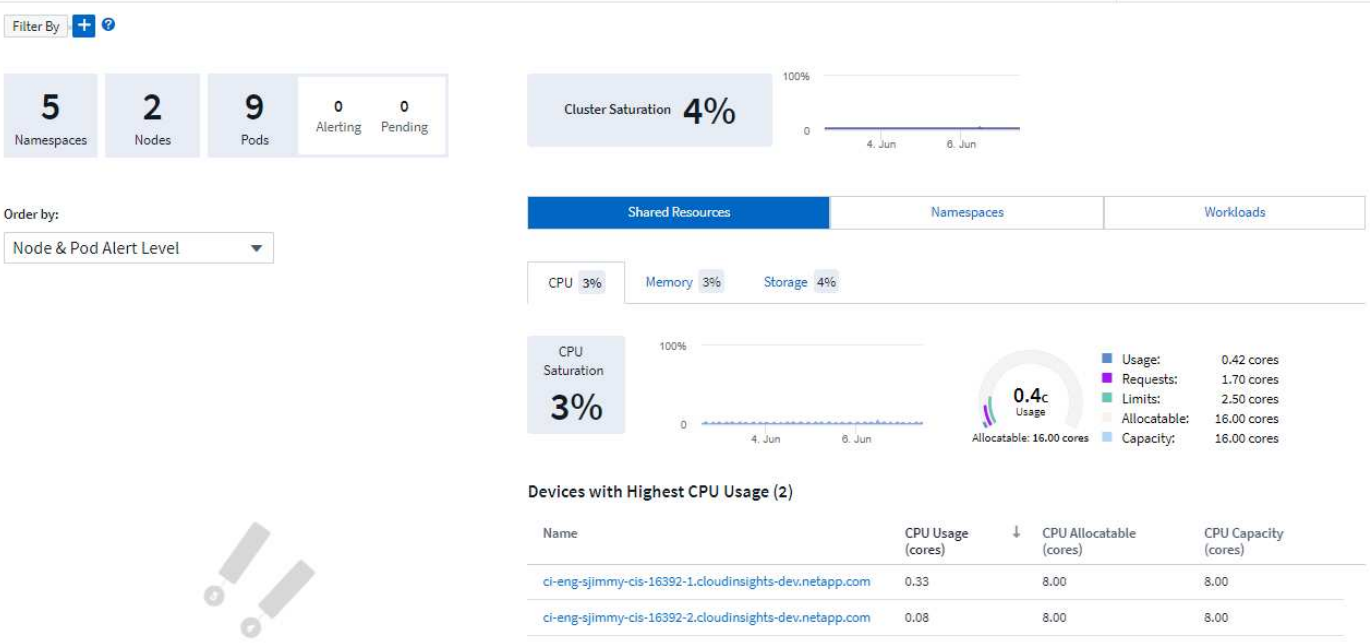
```
    ## Set metricAggregationInterval between 30 and 120
    ##
    # metricAggregationInterval: '60'

    ##
    ## 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 클러스터 세부 정보 페이지

Kubernetes 클러스터 세부 정보 페이지에는 Kubernetes 클러스터에 대한 자세한 개요가 표시됩니다.



네임스페이스, 노드 및 Pod 수

페이지 상단의 카운트는 클러스터에 있는 네임스페이스, 노드, 포드의 총 수와 현재 경고 중이거나 보류 중인 POP의 수를 보여줍니다.

공유 리소스 및 포화

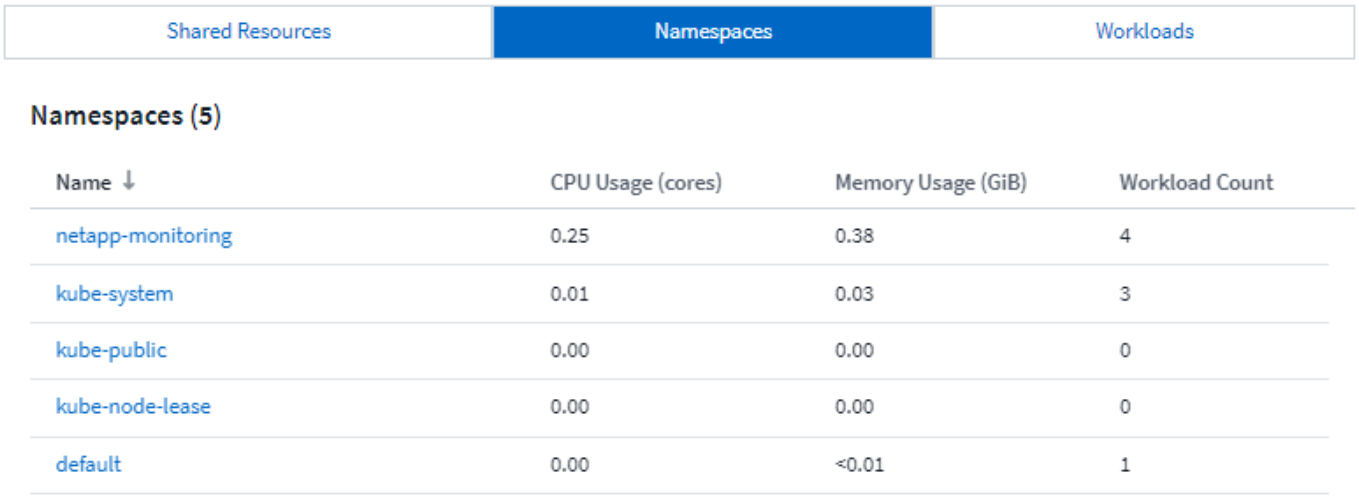
세부 정보 페이지의 오른쪽 상단에는 클러스터 포화도가 현재 백분율로 표시되고, 시간 경과에 따른 최근 추세를 보여주는 그래프도 표시됩니다. 클러스터 포화도는 각 시점에서 CPU, 메모리 또는 스토리지의 가장 높은 포화도를 말합니다.

그 아래에는 기본적으로 공유 리소스 사용량이 표시되며 CPU, 메모리, 저장소에 대한 탭이 있습니다. 각 탭에는 추가 사용 세부 정보와 함께 포화율과 시간 경과에 따른 추세가 표시됩니다. 저장의 경우, 표시되는 값은 백엔드 포화도와 파일 시스템 포화도 중 더 큰 값이며, 이 둘은 독립적으로 계산됩니다.

가장 많이 사용되는 기기는 아래 표에 표시되어 있습니다. 링크를 클릭하여 해당 장치를 살펴보세요.

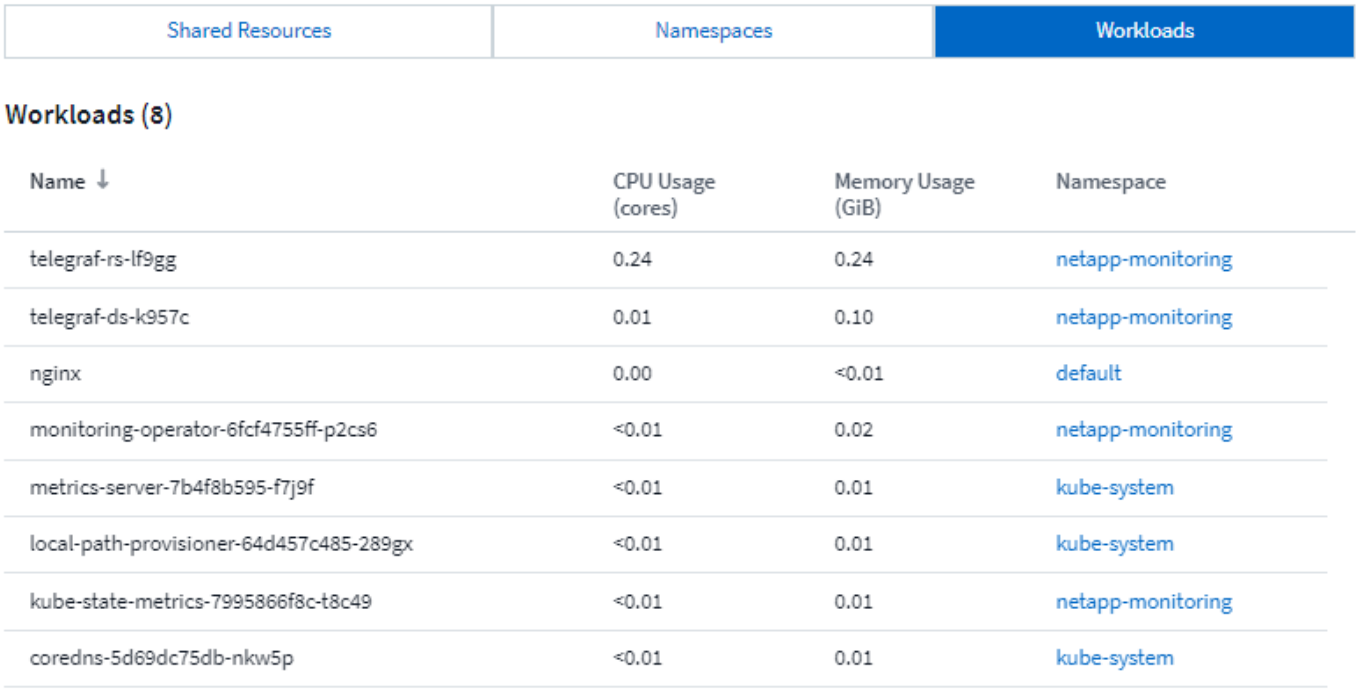
네임스페이스

네임스페이스 탭에는 Kubernetes 환경의 모든 네임스페이스 목록이 표시되며, CPU 및 메모리 사용량과 각 네임스페이스의 워크로드 수가 표시됩니다. 각 네임스페이스를 살펴보려면 이름 링크를 클릭하세요.



작업 부하

마찬가지로, 작업 부하 탭에는 각 네임스페이스의 작업 부하 목록이 표시되며, CPU 및 메모리 사용량도 다시 표시됩니다. 네임스페이스 링크를 클릭하면 각 항목으로 이동합니다.



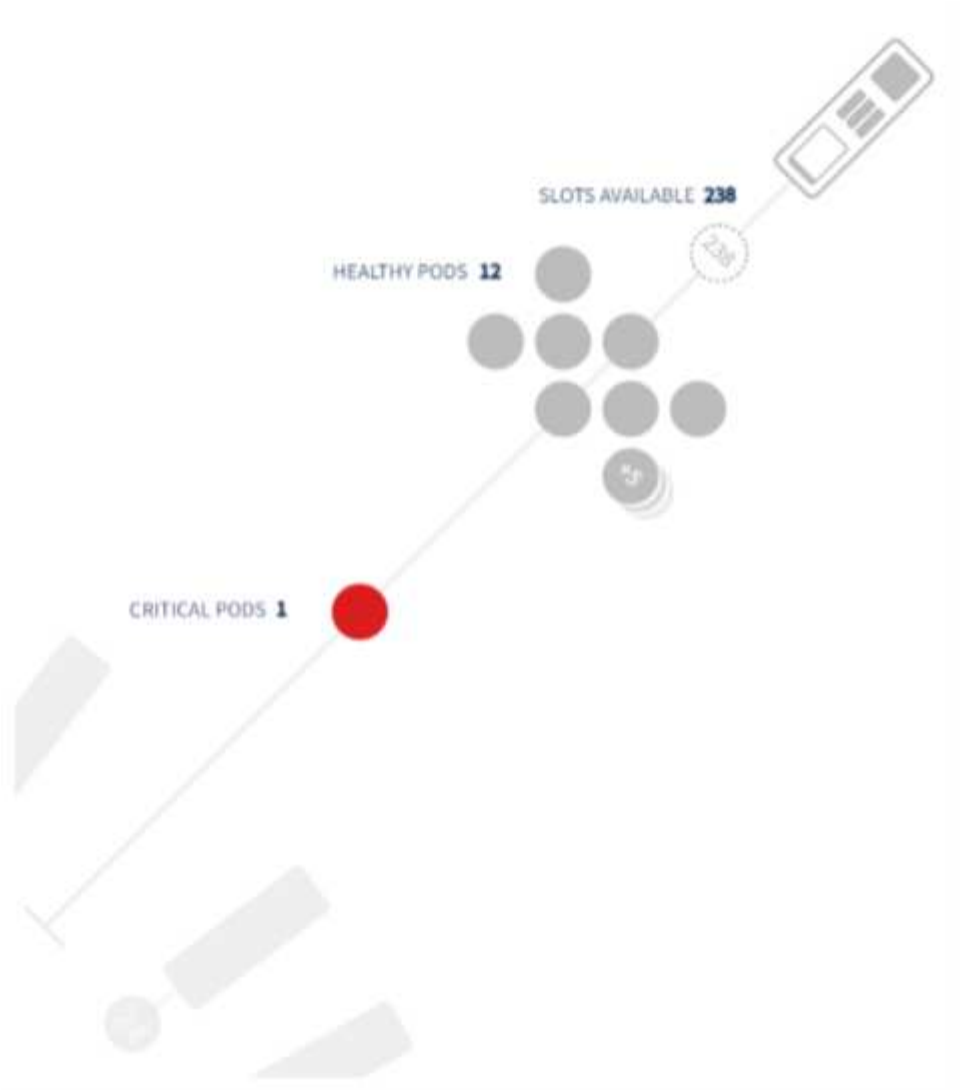
클러스터 "휠"

클러스터 "휠" 섹션은 노드와 포드 상태를 한눈에 보여주며, 자세한 정보를 확인할 수 있습니다. 클러스터에 이 페이지 영역에 표시할 수 있는 것보다 많은 노드가 포함되어 있는 경우, 제공된 버튼을 사용하여 휠을 돌릴 수 있습니다.

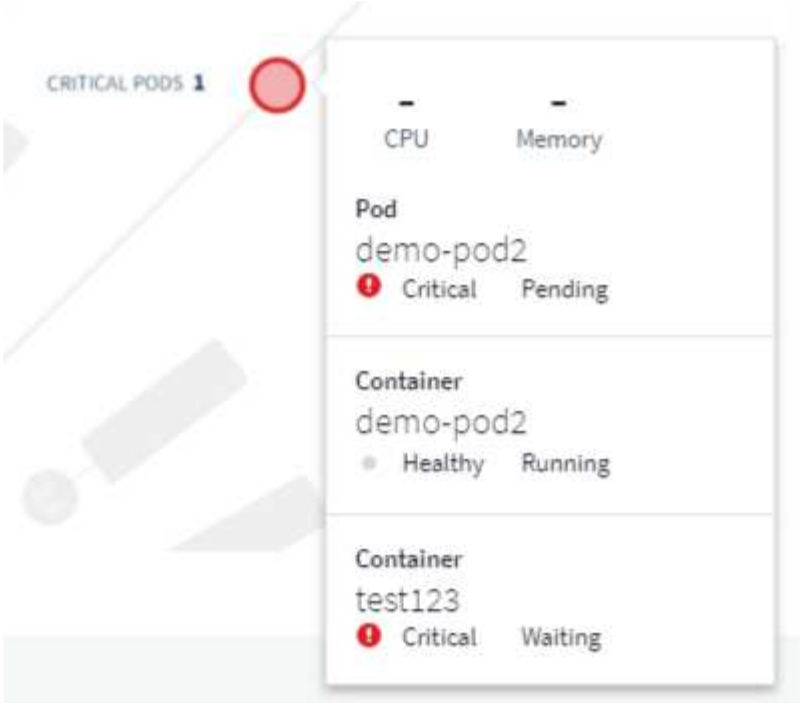
경고 포드 또는 노드는 빨간색으로 표시됩니다. "경고" 구역은 주황색으로 표시됩니다. 예약되지 않은(즉, 연결되지 않은) Pod는 클러스터 "휠"의 아래쪽 모서리에 표시됩니다.

포드(원)나 노드(막대) 위에 마우스를 올리면 해당 노드의 보기가 확장됩니다.

해당 보기에서 포드나 노드를 클릭하면 확장된 노드 보기로 확대됩니다.



여기에서 요소 위에 마우스를 올려 놓으면 해당 요소에 대한 세부 정보가 표시됩니다. 예를 들어, 이 예에서 중요한 포드 위에 마우스를 올리면 해당 포드에 대한 세부 정보가 표시됩니다.



노드 요소 위에 마우스를 올리면 파일 시스템, 메모리, CPU 정보를 볼 수 있습니다.

게이지에 대한 참고 사항

메모리와 CPU 게이지는 _할당 가능한 용량_과 _총 용량_에 대한 _사용됨_을 나타내므로 세 가지 색상으로 표시됩니다.

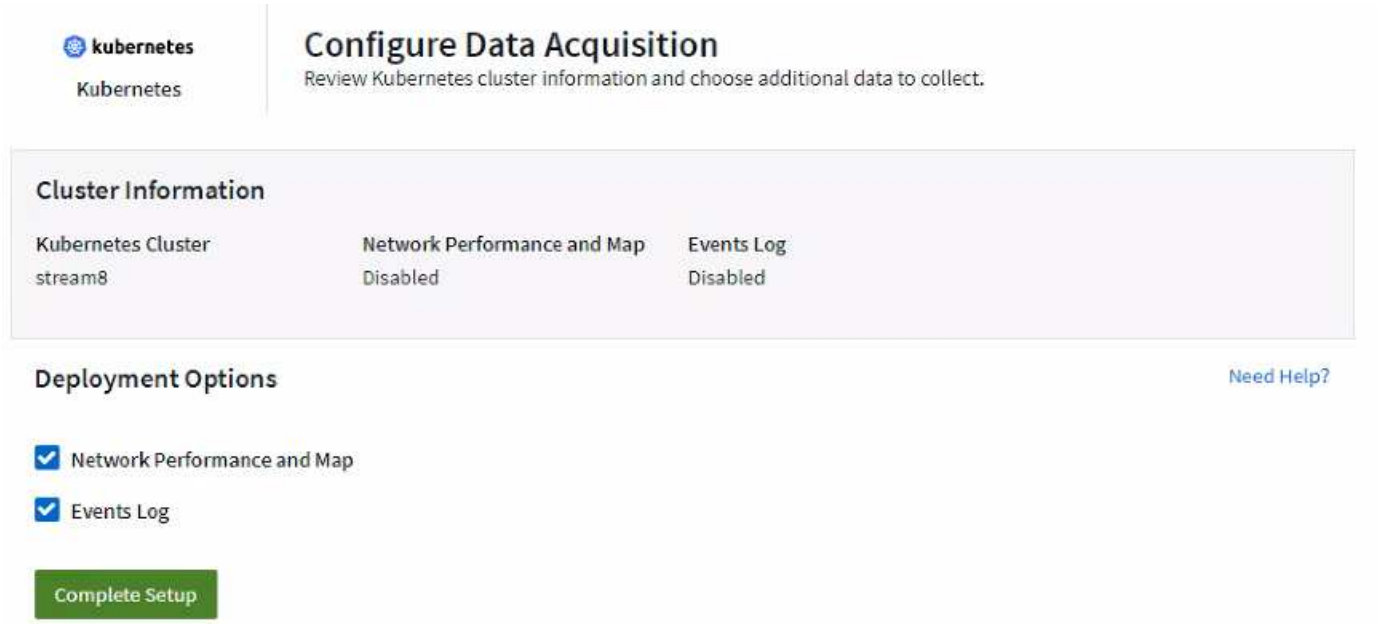
Kubernetes 네트워크 성능 모니터링 및 맵

Kubernetes 네트워크 성능 모니터링 및 맵 기능은 서비스(워크로드라고도 함) 간 종속성을 매핑하여 문제 해결을 간소화하고, 네트워크 성능 지연 시간과 이상 현상에 대한 실시간 가시성을 제공하여 사용자에게 영향을 미치기 전에 성능 문제를 식별합니다. 이 기능은 조직이 Kubernetes 트래픽 흐름을 분석하고 감사하여 전반적인 비용을 절감하는 데 도움이 됩니다.

주요 기능: • 워크로드 맵은 Kubernetes 워크로드 종속성과 흐름을 표시하고 네트워크 및 성능 문제를 강조합니다. • Kubernetes 포드, 워크로드 및 노드 간의 네트워크 트래픽을 모니터링하고 트래픽 및 지연 문제의 원인을 식별합니다. • 유입, 유출, 지역 간, 영역 간 네트워크 트래픽을 분석하여 전반적인 비용을 절감합니다.

전제 조건

Kubernetes 네트워크 성능 모니터링 및 맵을 사용하려면 먼저 다음을 구성해야 합니다."NetApp Kubernetes 모니터링 운영자" 이 옵션을 활성화하려면. 운영자 배포 중에 "네트워크 성능 및 맵" 확인란을 선택하여 활성화합니다. Kubernetes 랜딩 페이지로 이동하여 "배포 수정"을 선택하여 이 옵션을 활성화할 수도 있습니다.



모니터

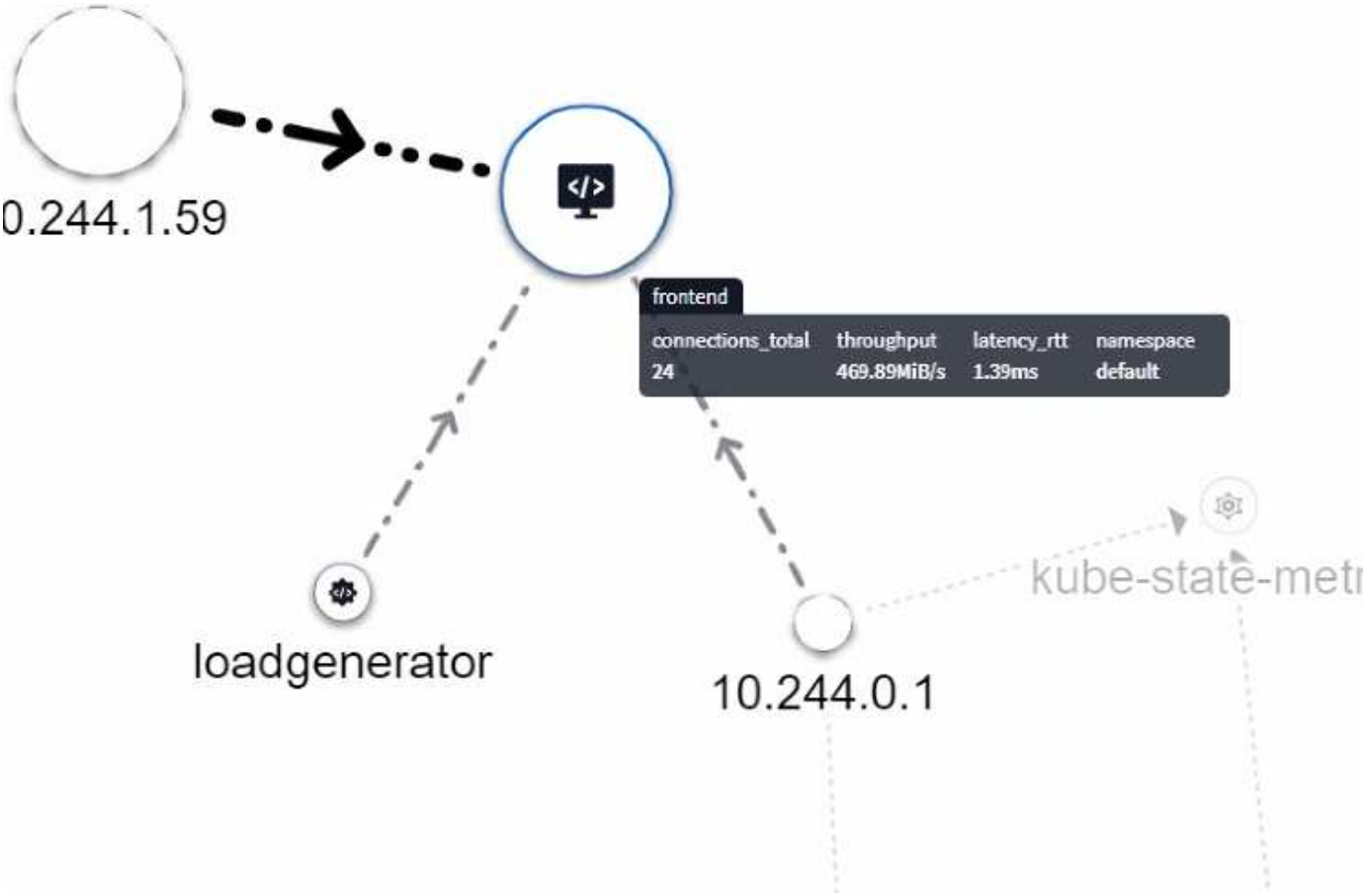
작업 부하 맵은 다음을 사용합니다."모니터" 정보를 도출하다. Data Infrastructure Insights 여러 가지 기본 Kubernetes Monitor를 제공합니다(기본적으로 일시 중지_될 수 있음). 원하는 모니터를 _재개(즉, 활성화)할 수 있으며, 워크로드 맵에서도 사용할 Kubernetes 객체에 대한 사용자 정의 모니터를 만들 수도 있습니다.

아래의 모든 개체 유형에 대해 Data Infrastructure Insights 메트릭 알림을 만들 수 있습니다. 데이터가 기본 개체 유형별로 그룹화되어 있는지 확인하세요.

- 쿠버네티스.워크로드
- 쿠버네티스 데몬셋
- 쿠버네티스 배포
- 쿠버네티스.크론잡
- 쿠버네티스.잡
- 쿠버네티스 레플리카셋
- 쿠버네티스.statefulset
- 쿠버네티스.포드
- 쿠버네티스 네트워크 트래픽 L4

지도

지도는 서비스/워크로드와 그들 간의 관계를 보여줍니다. 화살표는 교통 방향을 나타냅니다. 작업 부하 위에 마우스를 올리면 해당 작업 부하에 대한 요약 정보가 표시됩니다(다음 예에서 볼 수 있음).



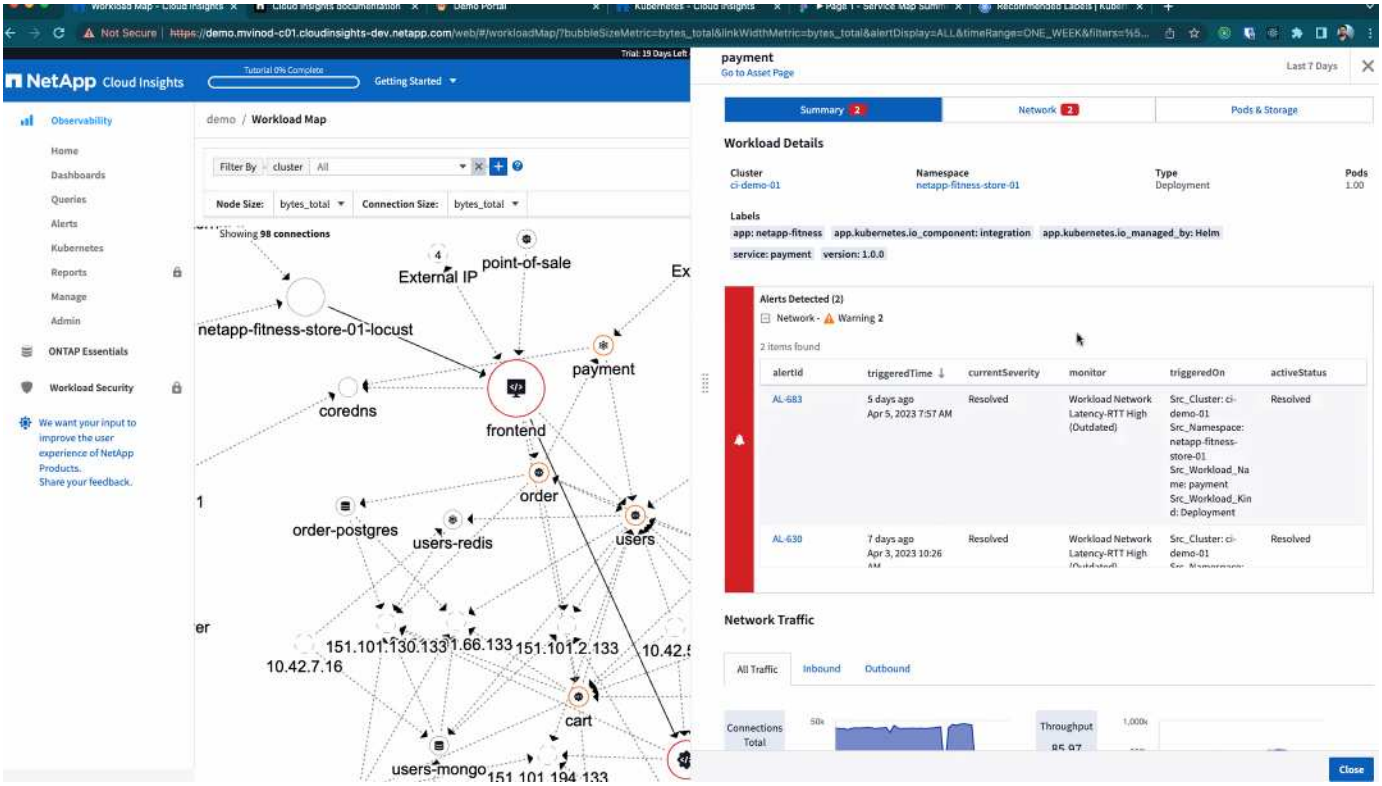
원 안의 아이콘은 다양한 서비스 유형을 나타냅니다. 아이콘은 기본 개체가 있는 경우에만 표시됩니다.라벨 .

각 원의 크기는 노드 크기를 나타냅니다. 이러한 크기는 상대적이며, 브라우저의 확대/축소 수준이나 화면 크기에 따라 실제 원 크기가 달라질 수 있습니다. 마찬가지로, 교통선 스타일을 통해 연결 규모를 한눈에 파악할 수 있습니다. 굵은 실선은 교통량이 많은 선이고, 연한 점선은 교통량이 적은 선입니다.

원 안의 숫자는 현재 서비스에서 처리 중인 외부 연결의 수입니다.

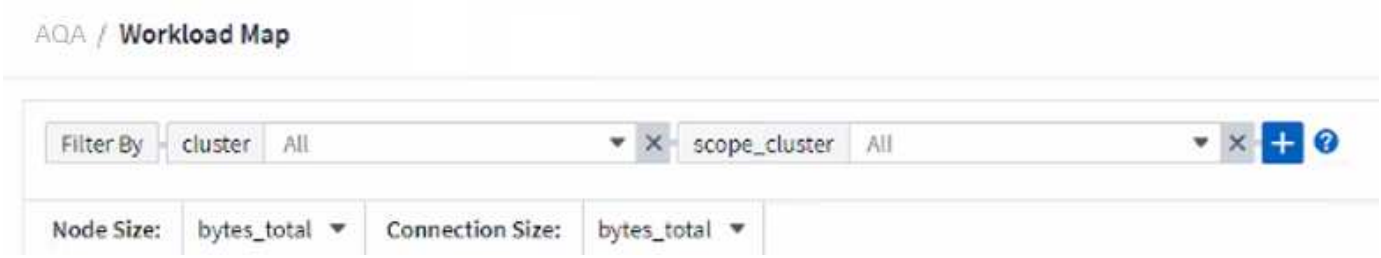
작업 부하 세부 정보 및 알림

색상으로 표시된 원은 작업 부하에 대한 경고 또는 위험 수준의 경보를 나타냅니다. 문제 요약을 보려면 원 위에 마우스를 올려놓고, 더 자세한 내용을 보려면 원을 클릭하세요.

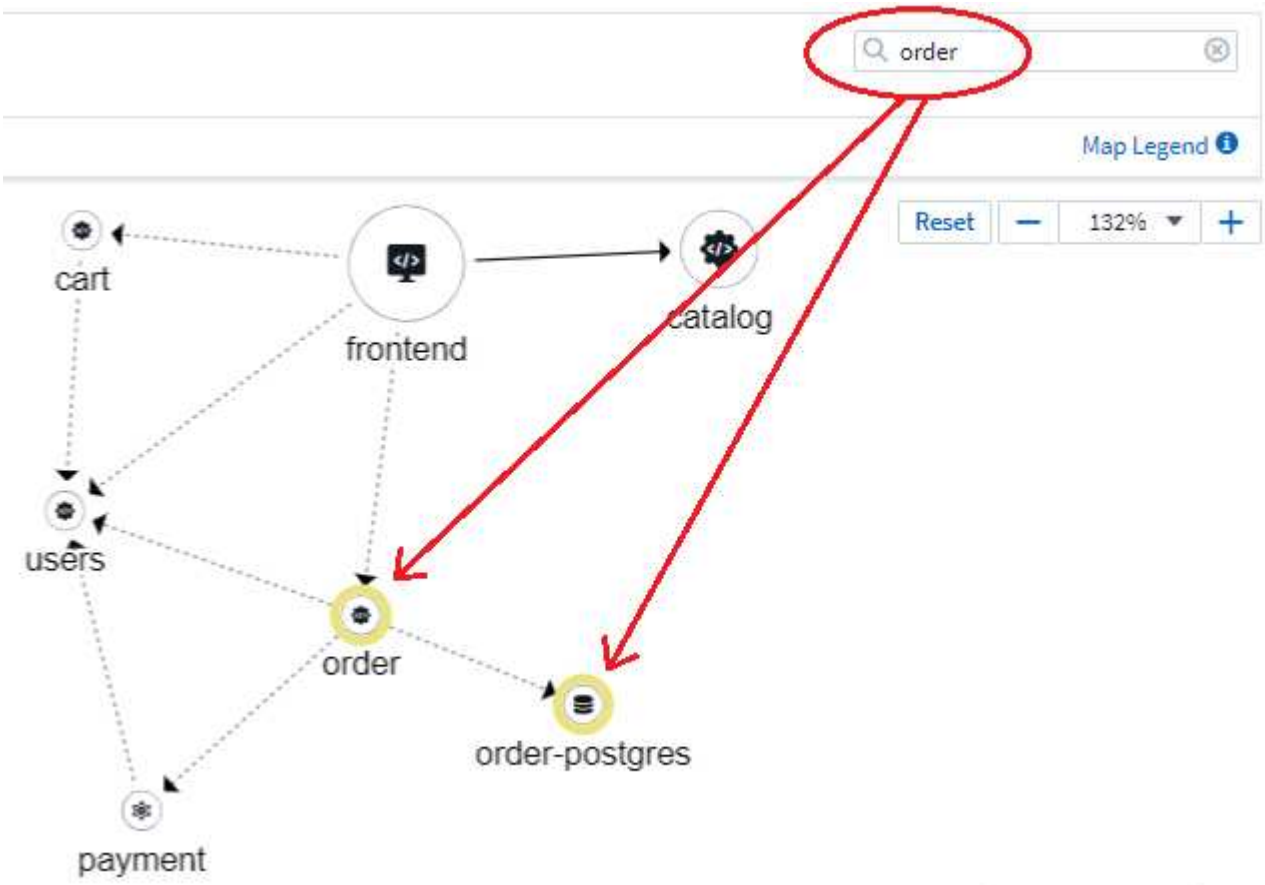


찾기 및 필터링

다른 Data Infrastructure Insights 기능과 마찬가지로 필터를 쉽게 설정하여 원하는 특정 개체나 워크로드 속성에 초점을 맞출 수 있습니다.



마찬가지로, 찾기 필드에 문자열을 입력하면 일치하는 워크로드가 강조 표시됩니다.



작업량 레이블

맵에서 표시된 작업 부하 유형(예: 원형 아이콘)을 식별하려면 작업 부하 레이블이 필요합니다. 라벨은 다음과 같이 파생됩니다.

- 일반적으로 실행되는 서비스/애플리케이션의 이름
- 소스가 포드인 경우:
  - 레이블은 포드의 작업 부하 레이블에서 파생됩니다.
  - 워크로드에 대한 예상 레이블: app.kubernetes.io/component
  - 라벨 이름 참조: https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels/
  - 추천 라벨:
    - 프런트엔드

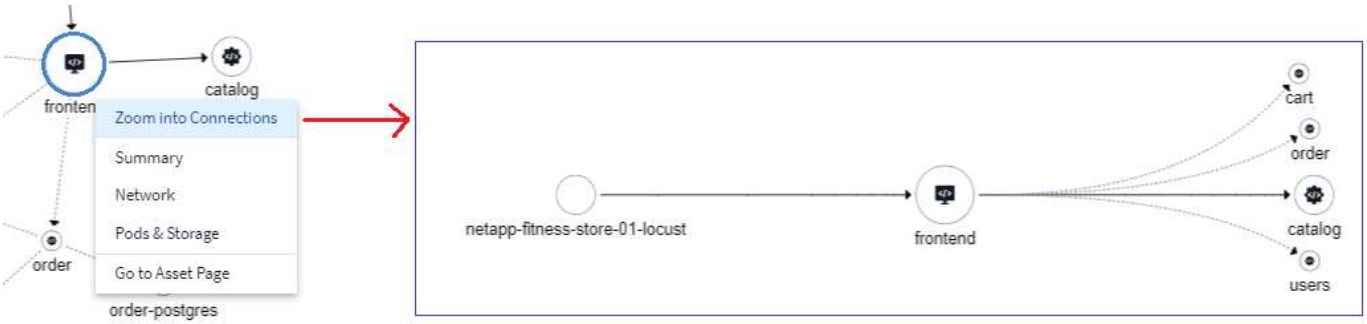
- 백엔드
- 데이터 베이스
- 은닉처
- 대기줄
- 카프카

- 소스가 Kubernetes 클러스터 외부에 있는 경우:
  - Data Infrastructure Insights DNS에서 확인된 이름을 구문 분석하여 서비스 유형을 추출하려고 시도합니다.

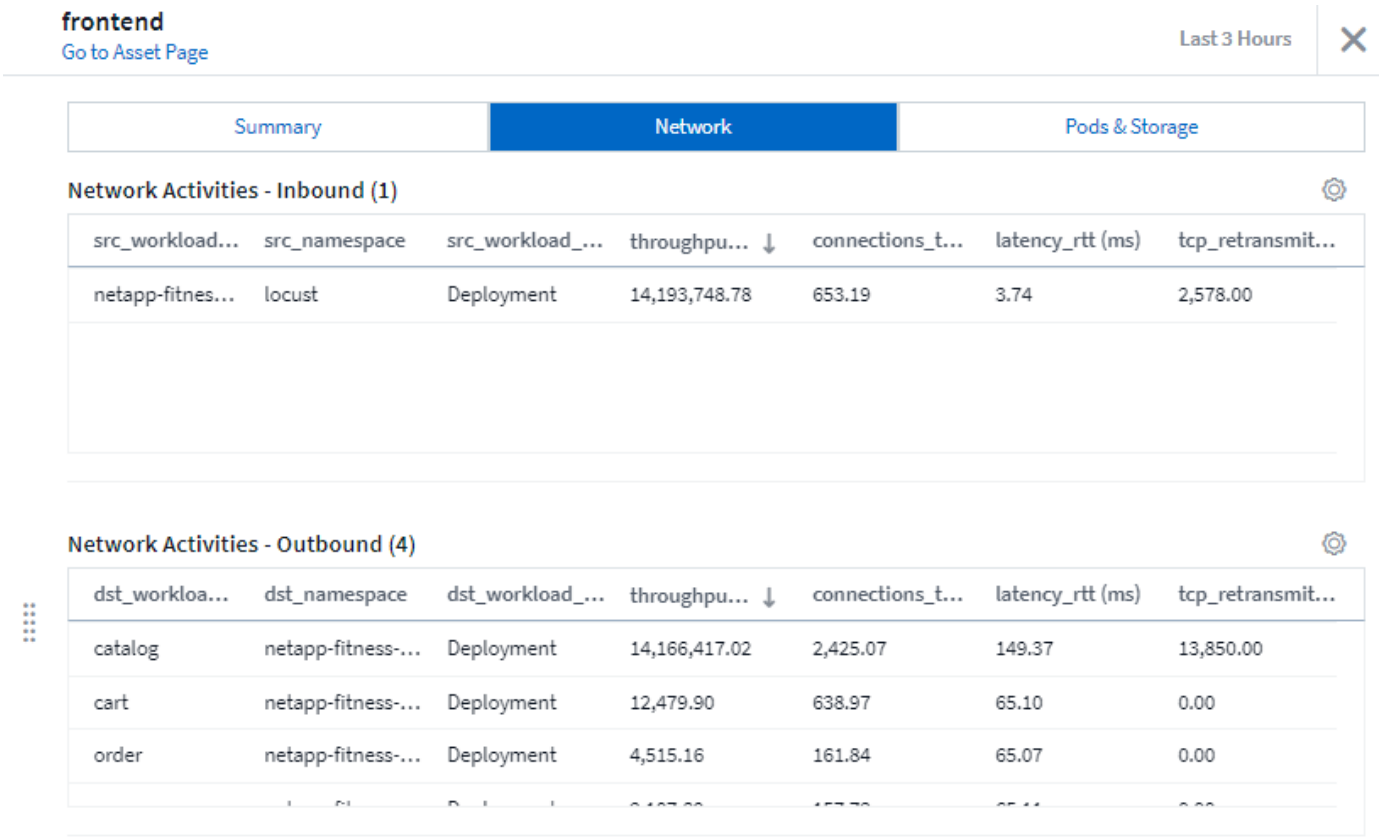
    예를 들어, DNS에서 확인된 이름이 _s3.eu-north-1.amazonaws.com_인 경우, 확인된 이름을 구문 분석하여 서비스 유형으로 _s3_를 얻습니다.

깊이 파고들다

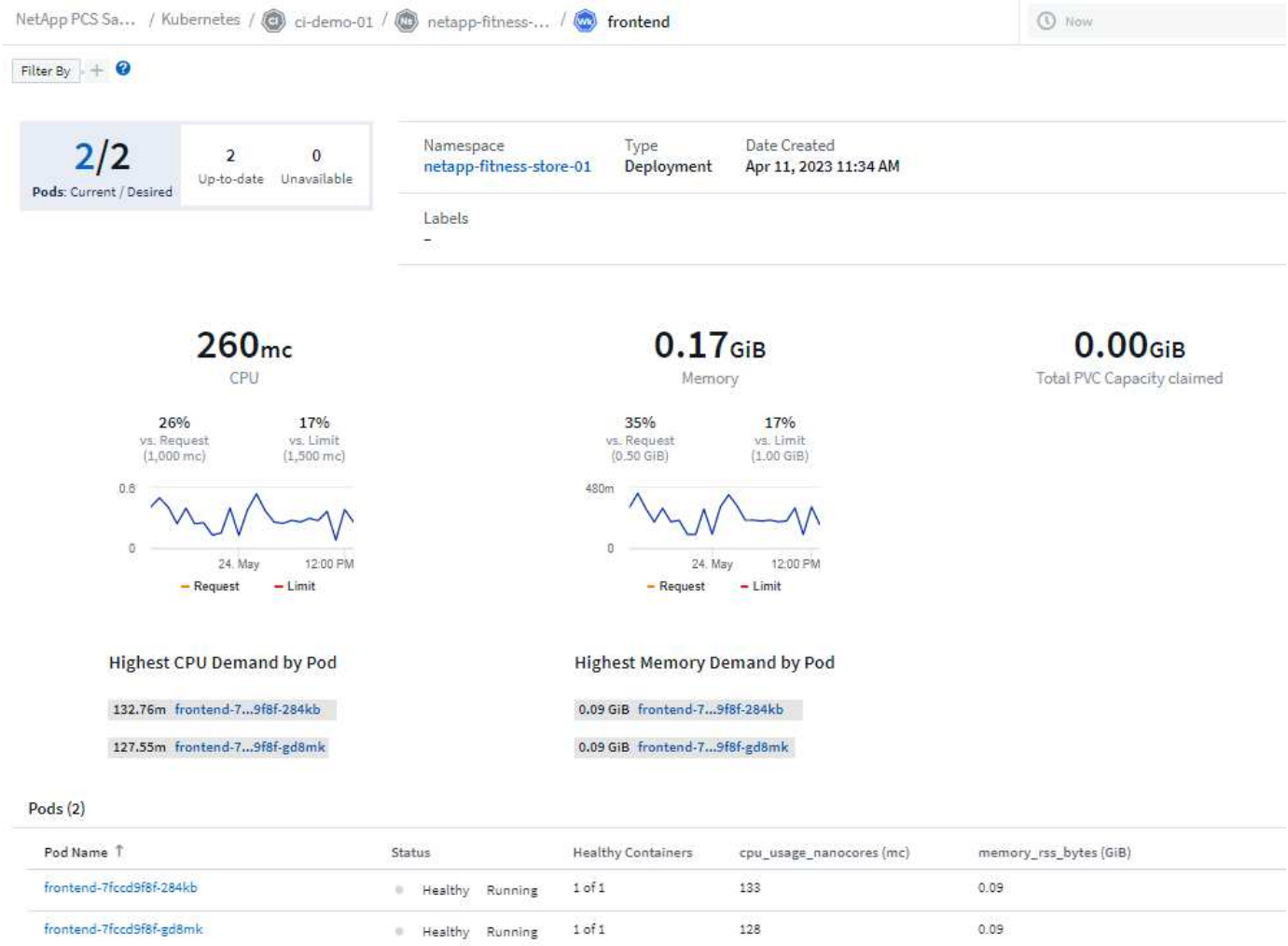
작업 부하를 마우스 오른쪽 버튼으로 클릭하면 더 자세히 탐색할 수 있는 추가 옵션이 표시됩니다. 예를 들어, 여기에서 확대하여 해당 작업 부하에 대한 연결을 볼 수 있습니다.



또는 세부 정보 슬라이드아웃 패널을 열어 요약, 네트워크 또는 *Pod* 및 저장소 탭을 직접 볼 수 있습니다.



마지막으로, _자산 페이지로 이동_을 선택하면 해당 워크로드에 대한 자세한 자산 랜딩 페이지가 열립니다.



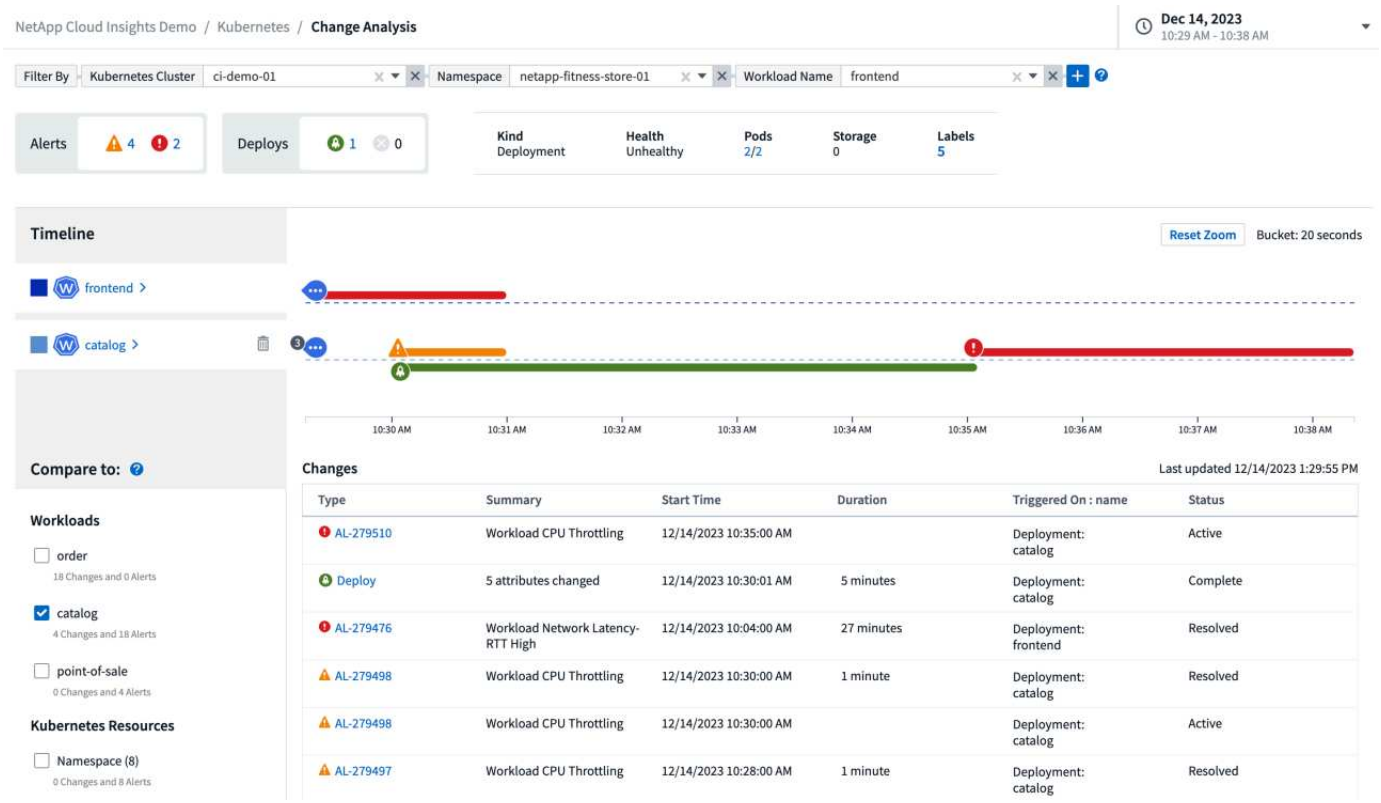
쿠버네티스 변경 분석

Kubernetes Change Analytics를 사용하면 K8s 환경에서 발생한 최근 변경 사항을 한눈에 볼 수 있습니다. 알림과 배포 상태를 손쉽게 확인할 수 있습니다. Change Analytics를 사용하면 모든 배포 및 구성 변경 사항을 추적하고 이를 K8s 서비스, 인프라 및 클러스터의 상태와 성능과 연관시킬 수 있습니다.

변화 분석은 어떻게 도움이 되나요?

- 멀티 테넌트 Kubernetes 환경에서는 잘못 구성된 변경 사항으로 인해 중단이 발생할 수 있습니다. 변경 분석은 워크로드와 구성 변경의 상태를 보고 상관 관계를 파악할 수 있는 단일 창을 제공하여 이를 지원합니다. 이는 동적 Kubernetes 환경의 문제 해결에 도움이 될 수 있습니다.

Kubernetes 변경 분석을 보려면 *Kubernetes > 변경 분석*으로 이동하세요.



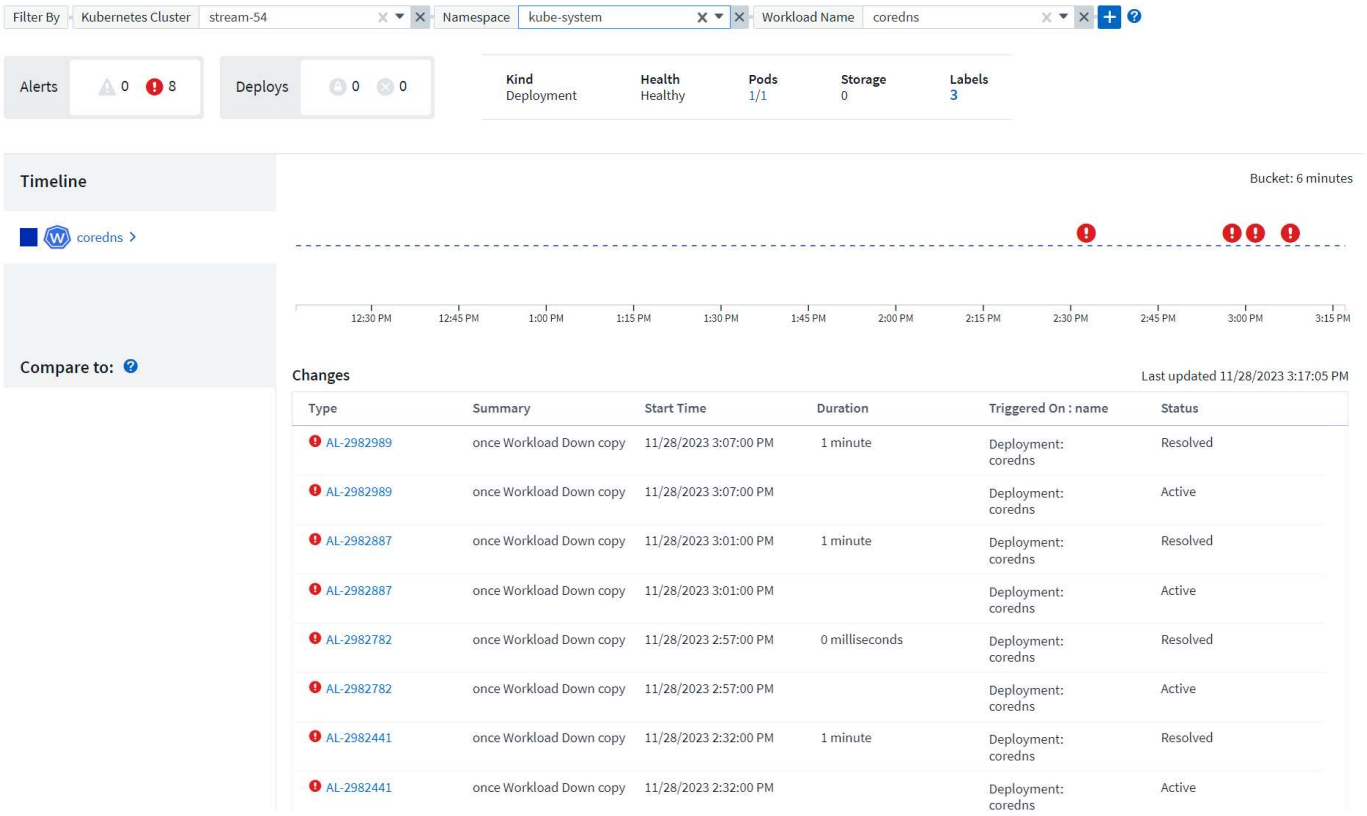
.

현재 선택된 Data Infrastructure Insights 시간 범위에 따라 페이지가 자동으로 새로 고쳐집니다. 시간 범위가 작을수록 화면 새로 고침이 더 자주 이루어집니다.

필터링

Data Infrastructure Insights 의 모든 기능과 마찬가지로 변경 목록을 필터링하는 것은 직관적입니다. 페이지 상단에서 Kubernetes 클러스터, 네임스페이스 또는 워크로드에 대한 값을 입력하거나 선택하거나 {+] 버튼을 선택하여 필터를 추가하세요.

특정 클러스터, 네임스페이스, 워크로드(설정한 다른 필터와 함께)로 필터링하면 해당 클러스터의 네임스페이스에서 해당 워크로드에 대한 배포 및 알림 타임라인이 표시됩니다. 그래프를 클릭하고 드래그하여 더욱 구체적인 시간 범위에 초점을 맞춰 확대해 보세요.

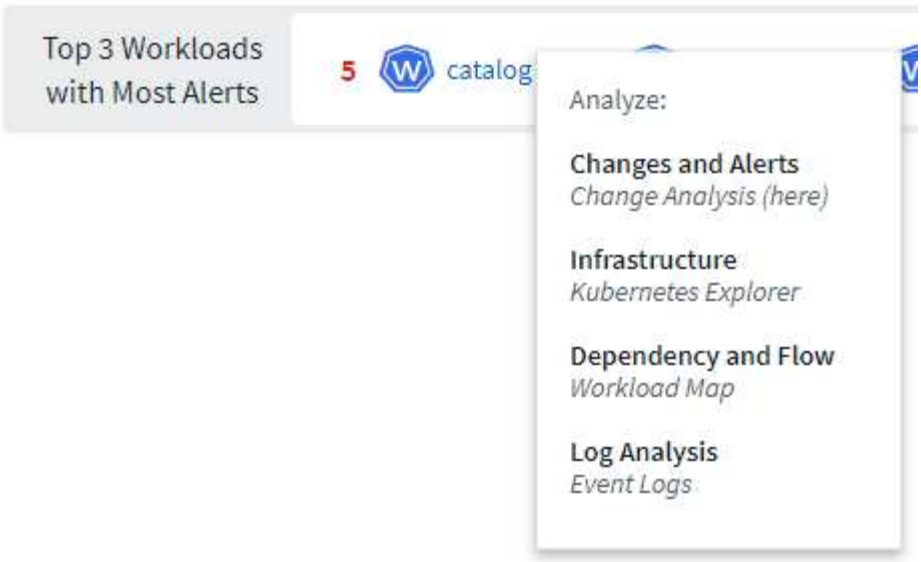


빠른 상태

필터링 영역 아래에는 여러 가지 상위 수준 지표가 있습니다. 왼쪽에는 경고(경고 및 중요)의 수가 표시됩니다. 이 숫자에는 활성 알림과 해결 알림이 모두 포함됩니다. 활성 알림만 보려면 "상태"에 대한 필터를 설정하고 "활성"을 선택하세요.

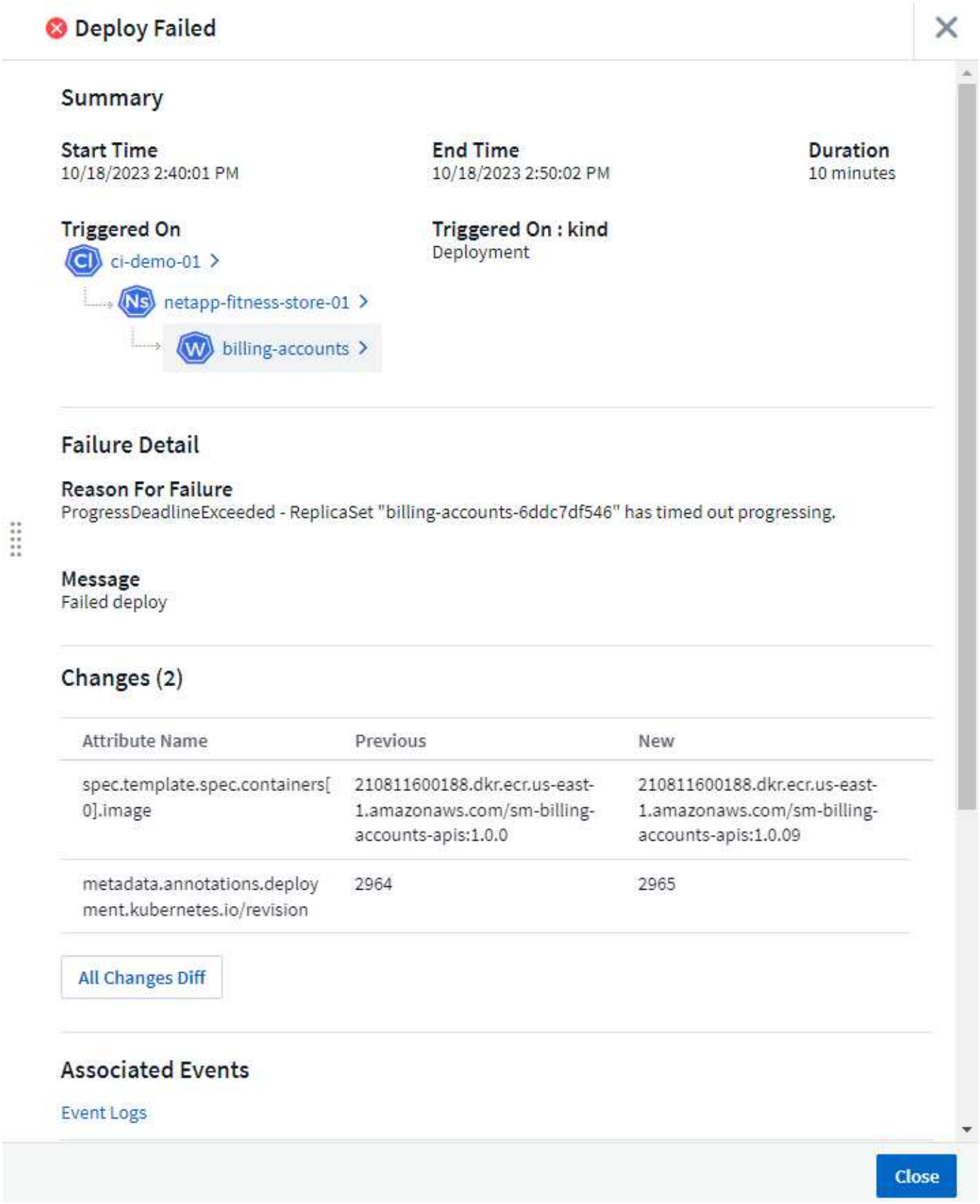
배포 상태도 여기에 표시됩니다. 다시 말해, 기본값은 시작됨, 완료, _실패_된 배포 수를 표시하는 것입니다. 실패한 배포만 보려면 "상태"에 대한 필터를 설정하고 "실패"를 선택하세요.

가장 많은 알림이 있는 상위 3개 워크로드가 다음입니다. 각 작업 부하 옆에 있는 빨간색 숫자는 해당 작업 부하와 관련된 알림 수를 나타냅니다. 워크로드 링크를 클릭하면 인프라(Kubernetes Explorer), 종속성(워크로드 맵) 또는 로그 분석(이벤트 로그)을 탐색할 수 있습니다.



세부 정보 패널

목록에서 변경 사항을 선택하면 변경 사항을 더 자세히 설명하는 패널이 열립니다. 예를 들어, 실패한 배포를 선택하면 배포 요약이 표시되며, 여기에는 시작 및 종료 시간, 기간, 배포가 트리거된 위치와 해당 리소스를 탐색할 수 있는 링크가 포함됩니다. 또한 실패 이유, 관련 변경 사항 및 연관된 이벤트도 표시됩니다.



알림을 선택하면 알림을 트리거한 모니터를 비롯하여 알림에 대한 세부 정보와 알림에 대한 시각적 타임라인을 보여주는 차트가 제공됩니다.