쿠버네티스
Cloud Insights

NetApp
April 15, 2024

목차

쿠버네티스

Kubernetes 클러스터 개요

Cloud Insights Kubernetes Explorer는 Kubernetes 클러스터의 전반적인 상태 및 사용을 표시하는 강력한 도구이며 사용자는 조사 영역을 쉽게 드릴다운할 수 있습니다.

대시보드 > Kubernetes Explorer * 를 클릭하면 Kubernetes 클러스터 목록 페이지가 열립니다. 이 개요 페이지에는 사용자 환경의 Kubernetes 클러스터 테이블이 포함되어 있습니다.

Name	Overall Saturation (%)	CPU Saturation (%)	Memory Saturation (%)	Storage Saturation (%)	Nodes	Pods	Namespaces	Workloads
self	56	25	56	31	2	63	18	68
setoK3s	4	2	3	4	2	9	5	7

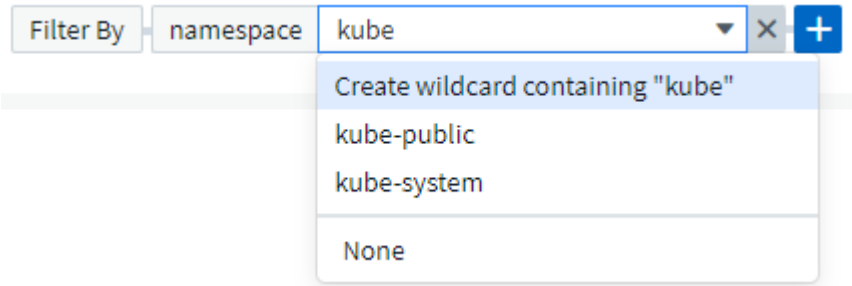
클러스터 목록

클러스터 목록에는 사용자 환경의 각 클러스터에 대한 다음 정보가 표시됩니다.

- 클러스터 * 이름 **.** 클러스터 이름을 클릭하면 가 열립니다 **"** 상세 페이지 ***"** 클러스터에 대해.
- * 채도 * 백분율. 전체 채도란 CPU, 메모리 또는 저장소 채도가 가장 높은 것입니다.
- 클러스터의 * 노드 수 * 입니다. 이 번호를 클릭하면 노드 목록 페이지가 열립니다.
- 클러스터의 Pod 수 이 번호를 클릭하면 창 목록 페이지가 열립니다.
- 클러스터의 * 네임스페이스 * 수입니다. 이 번호를 클릭하면 네임스페이스 목록 페이지가 열립니다.
- 클러스터에 있는 * 워크로드 * 수. 이 번호를 클릭하면 워크로드 목록 페이지가 열립니다.

필터 다듬기

필터링을 할 때 입력을 시작하면 현재 텍스트를 기반으로 * 와일드카드 필터 * 를 만들 수 있는 옵션이 표시됩니다. 이 옵션을 선택하면 와일드카드 식과 일치하는 모든 결과가 반환됩니다. NOT 또는 And를 사용하여 * 식 * 을 만들거나 "없음" 옵션을 선택하여 필드의 null 값을 필터링할 수도 있습니다.



와일드카드 또는 식(예 NOT, 및, "없음" 등)은 필터 필드에 진한 파란색으로 표시됩니다. 목록에서 직접 선택한 항목은 연한 파란색으로 표시됩니다.

Kubernetes 필터는 상황에 따라 다릅니다. 예를 들어 특정 노드 페이지에 있는 경우 pod_NAME 필터는 해당 노드에 관련된 Pod만 나열합니다. 또한 특정 네임스페이스에 대한 필터를 적용하는 경우 pod_NAME 필터는 해당 네임스페이스에 있는 _ 및 _ 노드에 포드만 나열합니다.

와일드카드 및 식 필터링은 텍스트 또는 목록과 함께 사용할 수 있지만 수치, 날짜 또는 부울은 사용할 수 없습니다.

NetApp Kubernetes 모니터링 운영자를 설치 또는 업그레이드하기 전에

NetApp Kubernetes 모니터링 운영자를 업그레이드하기 전에 이 정보를 읽어 보십시오

전제 조건:

- 사용자 지정 또는 프라이빗 Docker 저장소를 사용하는 경우 사용자 지정 또는 프라이빗 Docker 저장소 사용 섹션의 지침을 따릅니다
- NetApp Kubernetes 모니터링 오퍼레이터 설치는 Kubernetes 버전 1.20 이상에서 지원됩니다.
- Cloud Insights가 백엔드 스토리지를 모니터링하고 Docker 컨테이너 런타임과 함께 Kubernetes를 사용하는 경우 Cloud Insights는 NFS 및 iSCSI에 대한 POD-PV-스토리지 매핑 및 메트릭을 표시할 수 있고, 다른 런타임에는 NFS만 표시할 수 있습니다.
- 2022년 8월부터 NetApp Kubernetes Monitoring Operator는 Pod 보안 정책(PSP)을 지원합니다. 환경에서 PSP를 사용하는 경우 최신 NetApp Kubernetes Monitoring Operator로 업그레이드해야 합니다.
- OpenShift 4.6 이상에서 실행 중인 경우 이러한 전제 조건이 충족되는지 확인하는 것 외에도 아래의 OpenShift 지침을 따라야 합니다.
- 모니터링은 Linux 노드에만 설치됩니다. Cloud Insights에서는 해당 플랫폼에서 다음 Kubernetes 레이블을 찾는 Kubernetes 노드 선택기를 지정하여 Linux를 실행하는 Kubernetes 노드 모니터링을 지원합니다.

플랫폼	라벨
Kubernetes v1.20 이상	Kubernetes.IO/OS = Linux
Rancher + Cattle.IO를 오케스트레이션/Kubernetes 플랫폼으로 사용	Cattle.io/OS = Linux

- NetApp Kubernetes Monitoring Operator 및 해당 종속성(Telegraf, kubbe-state-metrics, fluentbit 등)은 Arm64 아키텍처를 실행하는 노드에서 지원되지 않습니다.
- curl, kubbtl 명령을 사용할 수 있어야 합니다. 옵션 설치 단계에서는 Docker 명령이 필요합니다. 최상의 결과를 얻으려면 이러한 명령을 경로에 추가하십시오. 최소한 에이전트, 클러스터 역할, clusterrolebindings, customresourcedefinitions, Deployments 와 같은 Kubernetes 개체에 대한 액세스 권한으로 kubctl을 구성해야 합니다. 네임스페이스, 역할, rolebindings, 비밀, serviceaccount, 등을 다룹니다. 이러한 최소 클러스터 역할 권한이 있는 YAML 파일의 예는 여기 를 참조하십시오.
- NetApp Kubernetes 모니터링 운영자 설치에 사용할 호스트는 타겟 K8s 클러스터와 통신하도록 kubtl을 구성하고 Cloud Insights 환경에 인터넷에 연결되어 있어야 합니다.
- 설치 중에 프록시 뒤에 있거나 모니터링할 K8s 클러스터를 작동하는 경우 프록시 지원 구성 섹션의 지침을 따르십시오.

- NetApp Kubernetes Monitoring Operator는 다른 인스턴스와 충돌을 피하기 위해 고유한 kube-state-metrics를 설치합니다. 정확한 감사 및 데이터 보고를 위해 NTP(Network Time Protocol) 또는 SNTP(Simple Network Time Protocol)를 사용하여 Agent 시스템의 시간을 동기화하는 것이 좋습니다.
- Operator를 다시 배포하는 경우(즉, 업데이트 또는 교체 중인 경우) _new_api 토큰을 생성할 필요가 없습니다. 이전 토큰을 다시 사용할 수 있습니다.
- 또한, 최신 NetApp Kubernetes Monitoring Operator가 설치되어 있고 재생 가능한 API 액세스 토큰을 사용 중인 경우 만료되는 토큰이 자동으로 새/업데이트된 API 액세스 토큰으로 대체됩니다.
- 네트워크 모니터링:
  - Linux 커널 버전 4.18.0 이상이 필요합니다
  - Photon OS는 지원되지 않습니다.

오퍼레이터 구성

새로운 버전의 연산자에서는 가장 일반적으로 수정된 설정을 _AgentConfiguration_custom 리소스에서 구성할 수 있습니다. 운용자를 배포하기 전에 _operator-config.YAML_file을 편집하여 이 리소스를 편집할 수 있습니다. 이 파일에는 일부 설정의 주석 처리된 예제가 포함되어 있습니다. 의 목록을 참조하십시오 "사용 가능한 설정" 를 참조하십시오.

다음 명령을 사용하여 연산자를 배포한 후에도 이 리소스를 편집할 수 있습니다.

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

배포된 버전의 운영자가 `AgentConfiguration`을 지원하는지 확인하려면 다음 명령을 실행합니다.

```
kubectl get crd agentconfigurations.monitoring.netapp.com
```

"서버에서 오류 발생(`NotFound`)" 메시지가 표시되는 경우 `AgentConfiguration`을 사용하려면 먼저 연산자를 업그레이드해야 합니다.

시작하기 전에 유의해야 할 중요 사항

을(를) 사용하여 실행 중인 경우 프록시, 가 있습니다 사용자 지정 리포지토리또는 을(를) 사용하고 있습니다 OpenShift다음 섹션을 주의 깊게 읽으십시오.

에 대해서도 읽어 보세요 권한.

이전 설치에서 업그레이드하는 경우 를 읽으십시오 업그레이드 중 정보.

프록시 지원을 구성하는 중입니다

NetApp Kubernetes Monitoring Operator를 설치하기 위해 사용자 환경에서 프록시를 사용할 수 있는 두 가지 위치가 있습니다. 이러한 시스템은 동일하거나 별도의 프록시 시스템일 수 있습니다.

- 설치 코드 조각을 실행하는 동안("curl" 사용) 프록시가 있어야 스니펫이 실행되는 시스템을 Cloud Insights 환경에 연결할 수 있습니다
- 대상 Kubernetes 클러스터에서 Cloud Insights 환경과 통신하는 데 프록시가 필요합니다

둘 중 하나 또는 둘 모두에 대해 프록시를 사용하는 경우 NetApp Kubernetes 운영 모니터를 설치하려면 먼저 프록시가 Cloud Insights 환경에 대한 올바른 통신을 허용하도록 구성되어 있는지 확인해야 합니다. 예를 들어 운영자를 설치하려는 서버/VM에서 Cloud Insights에 액세스하고 Cloud Insights에서 바이너리를 다운로드할 수 있어야 합니다.

NetApp Kubernetes 운영 모니터를 설치하는 데 사용되는 프록시에 대해 Operator를 설치하기 전에 _http_proxy/https_proxy_environment 변수를 설정하십시오. 일부 프록시 환경에서는 _no_proxy 환경_변수를 설정해야 할 수도 있습니다.

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

1. 현재 사용자에 대한 _https_proxy_and/or_http_proxy_environment 변수를 설정합니다.
   a. 설정 중인 프록시에 인증(사용자 이름/암호)이 없으면 다음 명령을 실행합니다.

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

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

Kubernetes 클러스터가 Cloud Insights 환경과 통신하는 데 사용되는 프록시의 경우 이러한 지침을 모두 읽은 후 NetApp Kubernetes 모니터링 운영자를 설치하십시오.

NetApp Kubernetes Monitoring Operator를 구축하기 전에 operator-config.yAML에서 AgentConfiguration의 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 지정 또는 프라이빗 **Docker** 저장소 사용

기본적으로 NetApp Kubernetes 모니터링 운영자는 Cloud Insights 저장소에서 컨테이너 이미지를 가져옵니다. 모니터링을 위한 타겟으로 사용되는 Kubernetes 클러스터가 있고 해당 클러스터가 사용자 지정 또는 프라이빗 Docker 저장소 또는 컨테이너 레지스트리에서만 컨테이너 이미지를 풀도록 구성된 경우 NetApp Kubernetes Monitoring Operator가 필요로 하는 컨테이너에 대한 액세스를 구성해야 합니다.

NetApp 모니터링 오퍼레이터 설치 타일에서 "이미지 풀 스니펫"을 실행합니다. 이 명령은 Cloud Insights 리포지토리에 로그인하고 오퍼레이터의 모든 이미지 종속성을 풀한 다음 Cloud Insights 리포지토리에서 로그아웃합니다. 메시지가 표시되면 제공된 리포지토리 임시 암호를 입력합니다. 이 명령은 옵션 기능을 포함하여 오퍼레이터가 사용하는 모든 이미지를 다운로드합니다. 이러한 이미지가 사용되는 기능은 아래를 참조하십시오.

핵심 운영자 기능 및 Kubernetes 모니터링

- NetApp - 모니터링
- kubbe-RBAC-proxy입니다
- Kudbe-state-metrics를 나타냅니다
- 텔레그라프
- distroless-root-user.(거리 없는 루트 사용자

이벤트 로그

- 유창한 비트
- Kubernetes - 이벤트 - 수출자

네트워크 성능 및 맵

- CI-NET-관찰자

회사 정책에 따라 운영 Docker 이미지를 프라이빗/로컬/엔터프라이즈 Docker 저장소로 밀어 넣습니다. 리포지토리에서 이러한 이미지에 대한 이미지 태그 및 디렉터리 경로가 Cloud Insights 리포지토리의 이미지 태그 및 디렉터리 경로와 일치하는지 확인합니다.

운영자 배포에서 모니터링 - 운영자 배포를 편집합니다. YAML을 편집하고 모든 이미지 참조를 수정하여 개인 Docker 저장소를 사용하십시오.

```
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.yAML에서 AgentConfiguration을 편집하여 새로운 Docker 저장소 위치를 반영하십시오. 개인 리포지토리에 대한 새 imagePullSecret을 만듭니다. 자세한 내용은 *https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/* 참조하십시오

```
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: https://docs.netapp.com/us-
en/cloudinsights/task_config_telegraf_agent_k8s.html#using-a-custom-or-
private-docker-repository
  dockerRepo: your.docker.repo/long/path/to/test
  # Optional: A docker image pull secret that maybe needed for your
private docker registry
  dockerImagePullSecret: docker-secret-name
```

OpenShift 지침

OpenShift 4.6 이상에서 실행 중인 경우 _operator-config.yAML_에서 AgentConfiguration을 편집하여 _Privileged_setting을 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 수준 높은 보안을 구현할 수 있습니다.

권한

모니터링 중인 클러스터에 ClusterRole이 없는 사용자 지정 리소스가 포함되어 있는 경우 "보려는 애그리게이트", 이벤트 로그로 이러한 리소스를 모니터링하려면 운영자에게 수동으로 이러한 리소스에 대한 액세스 권한을 부여해야

합니다.

1. edit_operator-additional-permissions.yaml_를 설치하기 전이나 설치 후 resource_ClusterRole/<namespace>-additional-permissions_를 편집합니다
2. 동사 ["get", "watch", "list"]를 사용하여 원하는 apiGroups 및 리소스에 대한 새 규칙을 만듭니다. https://kubernetes.io/docs/reference/access-authn-authz/rbac/ 를 참조하십시오
3. 변경 사항을 클러스터에 적용합니다

톨레러스와 얼룩을 볼 수 있습니다

NetApp-CI-Telegraf-DS_,*netapp-ci-fluent-bit-DS* 및_netapp-ci-net-observer-L4-DS_Demets는 모든 노드에서 데이터를 올바르게 수집하기 위해 클러스터의 모든 노드에 Pod를 예약해야 합니다. 운용자는 잘 알려진 일부 * 얼룩을 견딜 수 있도록 구성되었습니다. 노드에서 사용자 지정 얼룩을 구성하여 모든 노드에서 Pod가 실행되지 않도록 하는 경우 이러한 얼룩에 대해 * 공차 * 를 생성할 수 있습니다 "상담원 구성 _ 에서". 클러스터의 모든 노드에 사용자 지정 얼룩을 적용한 경우 운영자 포드를 예약 및 실행할 수 있도록 운영자 구축에 필요한 허용 오차도 추가해야 합니다.

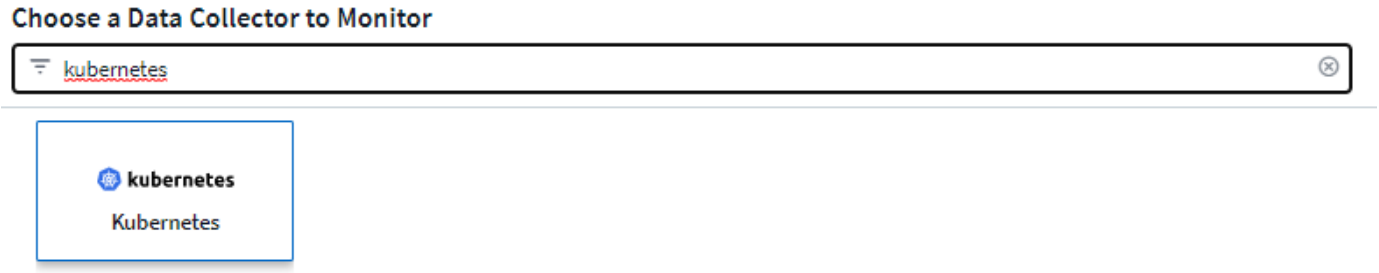
Kubernetes에 대해 자세히 알아보십시오 "오염과 내약입니다".

로 돌아갑니다 "* NetApp Kubernetes 모니터링 오퍼레이터 설치 * 페이지"

Kubernetes Monitoring Operator 설치 및 구성

Cloud Insights는 Kubernetes 컬렉션을 위한 * NetApp Kubernetes Monitoring Operator * (NKMO)를 제공합니다. 데이터 수집기 추가 시 "Kubernetes" 타일을 선택하면 됩니다.

> Cloud Insights Federal Edition을 사용하는 경우 설치 및 구성 지침이 이 페이지의 지침과 다를 수 있습니다. Cloud Insights의 지침에 따라 NetApp Kubernetes 모니터링 운영자를 설치합니다.

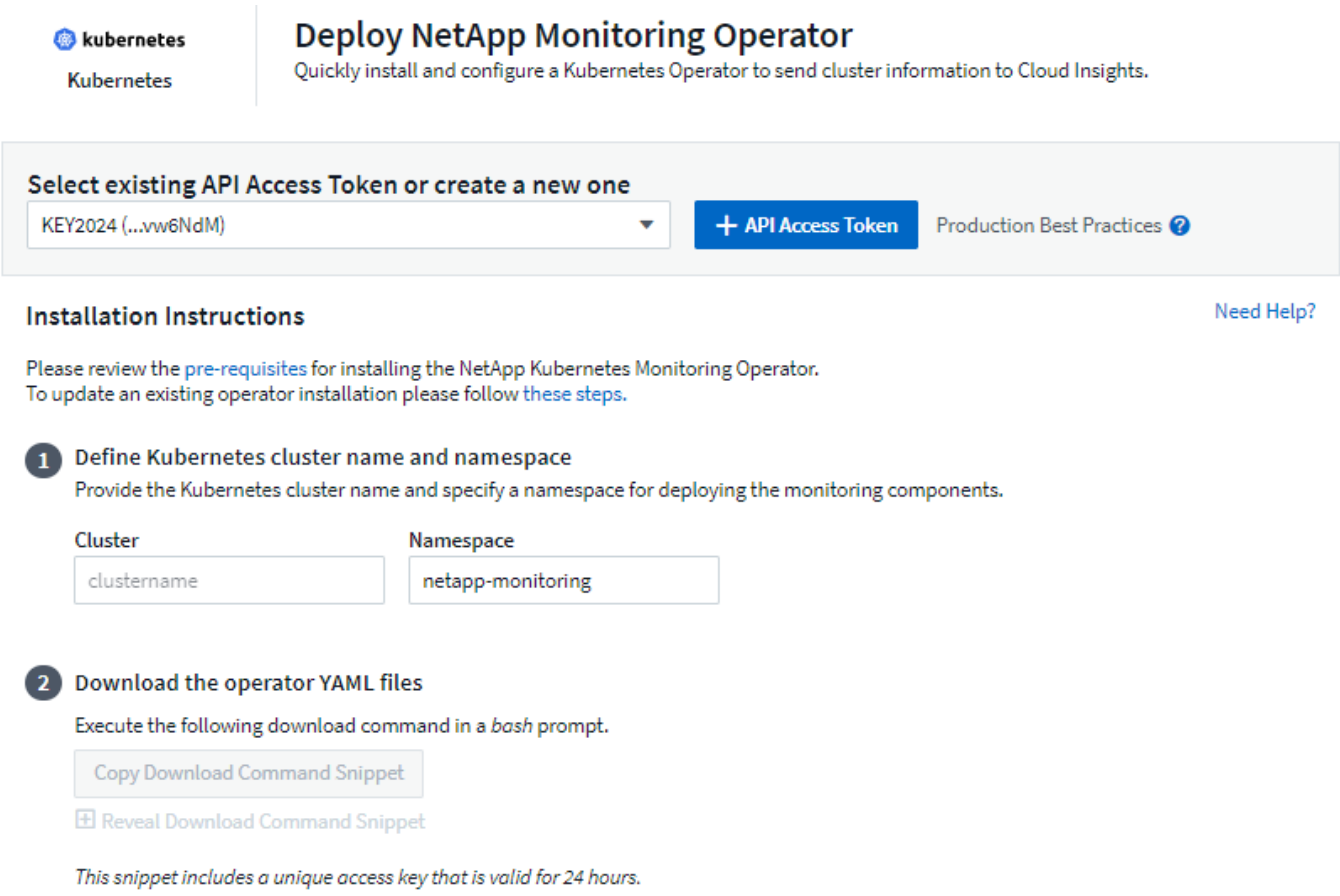


Cloud Insights Docker 레지스트리에서 Kubernetes Operator 및 데이터 수집기를 다운로드할 수 있습니다. 그런 다음 운영자는 이 수집기의 수명 주기 관리를 포함하여 데이터를 획득하기 위해 Kubernetes 클러스터 노드에 배포된 운영자 호환 수집기를 관리합니다. 이 체인에 이어 수집기에서 데이터를 획득하여 Cloud Insights로 전송합니다.

NetApp Kubernetes Monitoring Operator를 설치하기 전에

> 를 읽습니다 "설치 또는 업그레이드 전" NetApp Kubernetes 모니터링 운영자를 설치 또는 업그레이드하기 전에 사전 설명서를 제공해야 합니다.

NetApp Kubernetes Monitoring Operator 설치



3 Optional: Upload the operator images to your private repository

By default, the operator pulls container images from the Cloud Insights repository. To use a private repository, download the required images using the Image Pull command. Then upload them to your private repository maintaining the same tags and directory structure.
Finally, update the image paths in operator-deployment.yaml and the docker repository settings in operator-config.yaml.
For more information review the documentation.

Copy Image Pull Snippet

Reveal Image Pull Snippet

Copy Repository Password

Reveal Repository Password

This password is valid for 24 hours.

4 Optional: Review available configuration options

Configure custom options such as proxy and private repository settings. Review the instructions and available options.

5 Deploy the operator (create new or upgrade existing)

Execute the *kubectl* snippet to apply the following operator YAML files.

- operator-setup.yaml - Create the operator's dependencies.
- operator-secrets.yaml - Create secrets holding your API key.
- operator-deployment.yaml, operator-cr.yaml - Deploy the NetApp Kubernetes Monitoring Operator.
- operator-config.yaml - Apply the configuration settings if not already present.

Copy kubectl Apply Snippet

Reveal kubectl Apply Snippet

After deploying the operator, **delete or securely store operator-secrets.yaml.**

6 Next

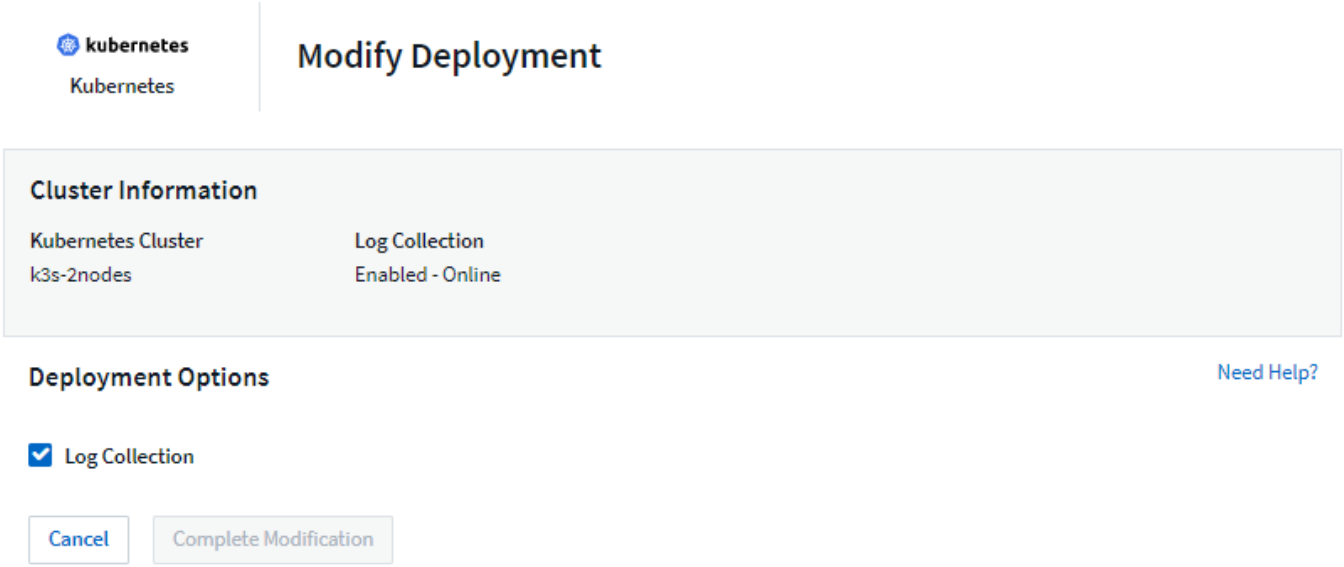
Kubernetes에 **NetApp Kubernetes Monitoring Operator Agent**를 설치하는 단계:

1. 고유한 클러스터 이름 및 네임스페이스를 입력합니다. 있는 경우 업그레이드 중 이전 Kubernetes Operator에서 동일한 클러스터 이름과 네임스페이스를 사용합니다.
2. 이러한 내용을 입력하면 다운로드 명령 스니펫을 클립보드에 복사할 수 있습니다.
3. 스니펫을 _bash_window에 붙여 넣고 실행합니다. 오퍼레이터 설치 파일이 다운로드됩니다. 스니펫에는 고유한 키가 있으며 24시간 동안 유효합니다.
4. 사용자 지정 또는 개인 리포지토리가 있는 경우 선택적 이미지 풀 스니펫을 복사하여 _bash_shell에 붙여 넣고 실행합니다. 이미지를 가져온 후 개인 저장소에 복사합니다. 동일한 태그 및 폴더 구조를 유지해야 합니다. operator-deployment.YAML_의 경로와 _operator-config.YAML_의 Docker 리포지토리 설정을 업데이트합니다.
5. 필요한 경우 프록시 또는 개인 리포지토리 설정과 같은 사용 가능한 구성 옵션을 검토합니다. 에 대해 자세히 알아볼 수 있습니다 "구성 옵션".
6. 준비가 되면 kubeck Apply 스니펫을 복사하고 다운로드한 다음 실행하여 Operator 를 배포합니다.
7. 설치가 자동으로 진행됩니다. 완료되면 *Next* 단추를 클릭합니다.
8. 설치가 완료되면 *Next* 단추를 클릭합니다. 또한 _operator-비밀.YAML_파일을 삭제하거나 안전하게 보관하십시오.

에 대해 자세히 알아보십시오 프록시를 구성하는 중입니다.

에 대해 자세히 알아보십시오 사용자 지정/프라이빗 Docker 저장소 사용.

NetApp Kubernetes Monitoring Operator를 설치할 때 Kubernetes EMS 로그 수집이 기본적으로 활성화됩니다. 설치 후 이 컬렉션을 비활성화하려면 Kubernetes 클러스터 세부 정보 페이지 상단의 * Modify Deployment * 버튼을 클릭하고 "Log collection"을 선택 취소합니다.



이 화면에는 현재 로그 수집 상태도 표시됩니다. 가능한 상태는 다음과 같습니다.

- 사용 안 함
- 활성화됨
- Enabled - 설치가 진행 중입니다
- 사용 - 오프라인
- 활성화됨 - 온라인
- 오류 - API 키에 권한이 부족합니다

업그레이드 중

최신 **NetApp Kubernetes** 모니터링 사업자로 업그레이드

기존 운영자와 함께 AgentConfiguration이 존재하는지 확인합니다(네임스페이스가 DEFAULT_NetApp-MONITORING_이 아닌 경우 해당 네임스페이스를 대체합니다).

```
 kubectl -n netapp-monitoring get agentconfiguration netapp-monitoring-
configuration
AgentConfiguration이 있는 경우:
```

- 설치합니다 기존 오퍼레이터에 대한 최신 운영자.
  - 항상 확인하십시오 최신 컨테이너 이미지를 가져오는 중입니다 사용자 지정 리포지토리를 사용하는 경우

AgentConfiguration이 없는 경우:

- Cloud Insights에서 인식하는 클러스터 이름을 기록합니다(네임스페이스가 기본 NetApp 모니터링이 아닌 경우 해당 네임스페이스를 대체).

```
 kubectl -n netapp-monitoring get agent -o
jsonpath='{.items[0].spec.cluster-name}'
* 기존 운영자의 백업을 생성합니다(네임스페이스가 기본 NetApp 모니터링이 아닌 경우 적절한
네임스페이스를 대체).
```

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

  - 동일한 클러스터 이름을 사용합니다.
  - 최신 운영자 YAML 파일을 다운로드한 후 배포하기 전에 agent_backup.YAML에서 발견된 모든 사용자 정의를 다운로드한 operator-config.YAML에 이식하십시오.
  - 항상 확인하십시오 최신 컨테이너 이미지를 가져오는 중입니다 사용자 지정 리포지토리를 사용하는 경우

NetApp Kubernetes Monitoring Operator를 중지하고 시작합니다

NetApp Kubernetes Monitoring Operator를 중지하려면 다음을 수행합니다.

```
 kubectl -n netapp-monitoring scale deploy monitoring-operator
--replicas=0
NetApp Kubernetes Monitoring Operator를 시작하려면 다음을 수행합니다.
```

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

제거 중

NetApp Kubernetes Monitoring Operator를 제거하려면 다음을 수행합니다

NetApp Kubernetes Monitoring Operator의 기본 네임스페이스는 "NetApp 모니터링"입니다. 고유한 네임스페이스를 설정한 경우 이러한 네임스페이스 및 모든 후속 명령 및 파일로 대체합니다.

다음 명령을 사용하여 모니터링 연산자의 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>
첫 번째 명령이 "리소스를 찾을 수 없음"을 반환하면 다음 지침에 따라 모니터링 연산자의 이전
버전을 제거합니다.
```

다음 명령을 순서대로 실행합니다. 현재 설치에 따라 이러한 명령 중 일부는 '개체를 찾을 수 없음' 메시지를 반환할 수 있습니다. 이러한 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>
```

보안 컨텍스트 제약 조건이 이전에 생성된 경우:

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

Kube-state-metrics 정보

NetApp Kubernetes Monitoring Operator가 자동으로 Kudbe-state-metrics를 설치하므로 사용자 개입이 필요하지 않습니다.

Kudbe-state-Metrics 카운터

다음 링크를 사용하여 이러한 kubbe 상태 메트릭 카운터에 대한 정보에 액세스할 수 있습니다.

1. "ConfigMap 메트릭입니다"
2. "메트릭 분월 설정"

3. "구현 메트릭"
4. "수신 메트릭"
5. "네임스페이스 메트릭"
6. "노드 메트릭"
7. "영구 볼륨 메트릭"
8. "잔류 볼륨 클레임 메트릭"
9. "POD 메트릭"
10. "ReplicaSet 메트릭입니다"
11. "비밀 지표"
12. "서비스 메트릭"
13. "StatefulSet 메트릭입니다"

```
== Configuring the Operator
새로운 버전의 연산자에서는 가장 일반적으로 수정된 설정을 _AgentConfiguration_custom
리소스에서 구성할 수 있습니다. 운용자를 배포하기 전에 _operator-config.YAML_file을
편집하여 이 리소스를 편집할 수 있습니다. 이 파일에는 일부 설정의 주석 처리된 예제가 포함되어
있습니다. 의 목록을 참조하십시오
xref:{relative_path}telegraf_agent_k8s_config_options.html["사용 가능한 설정"]
를 참조하십시오.
```

다음 명령을 사용하여 연산자를 배포한 후에도 이 리소스를 편집할 수 있습니다.

```
kubectl -n netapp-monitoring edit AgentConfiguration
배포된 버전의 운영자가 AgentConfiguration을 지원하는지 확인하려면 다음 명령을 실행합니다.
```

```
kubectl get crd agentconfigurations.monitoring.netapp.com
"서버에서 오류 발생(NotFound)" 메시지가 표시되는 경우 AgentConfiguration을 사용하려면
먼저 연산자를 업그레이드해야 합니다.
```

프록시 지원을 구성하는 중입니다

NetApp Kubernetes Monitoring Operator를 설치하기 위해 사용자 환경에서 프록시를 사용할 수 있는 두 가지 위치가 있습니다. 이러한 시스템은 동일하거나 별도의 프록시 시스템일 수 있습니다.

- 설치 코드 조각을 실행하는 동안("curl" 사용) 프록시가 있어야 스니펫이 실행되는 시스템을 Cloud Insights 환경에 연결할 수 있습니다
- 대상 Kubernetes 클러스터에서 Cloud Insights 환경과 통신하는 데 프록시가 필요합니다

이 중 하나 또는 둘 모두에 대해 프록시를 사용하는 경우 NetApp Kubernetes 운영 모니터를 설치하려면 먼저 프록시가 Cloud Insights 환경에 대한 올바른 통신을 허용하도록 구성되어 있는지 확인해야 합니다. 프록시가 있고 운영자를 설치하려는 서버/VM에서 Cloud Insights에 액세스할 수 있는 경우 프록시가 제대로 구성되었을 수 있습니다.

NetApp Kubernetes 운영 모니터를 설치하는 데 사용되는 프록시에 대해 Operator를 설치하기 전에 _http_proxy/https_proxy_environment 변수를 설정하십시오. 일부 프록시 환경에서는 _no_proxy 환경_변수를 설정해야 할 수도 있습니다.

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

1. 현재 사용자에 대한 _https_proxy_and/or_http_proxy_environment 변수를 설정합니다.
   a. 설정 중인 프록시에 인증(사용자 이름/암호)이 없으면 다음 명령을 실행합니다.

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

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

Kubernetes 클러스터가 Cloud Insights 환경과 통신하는 데 사용되는 프록시의 경우 이러한 지침을 모두 읽은 후 NetApp Kubernetes 모니터링 운영자를 설치하십시오.

NetApp Kubernetes Monitoring Operator를 구축하기 전에 operator-config.yAML에서 AgentConfiguration의 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 지정 또는 프라이빗 **Docker** 저장소 사용

기본적으로 NetApp Kubernetes 모니터링 운영자는 Cloud Insights 저장소에서 컨테이너 이미지를 가져옵니다. 모니터링을 위한 타겟으로 사용되는 Kubernetes 클러스터가 있고 해당 클러스터가 사용자 지정 또는 프라이빗 Docker 저장소 또는 컨테이너 레지스트리에서만 컨테이너 이미지를 풀도록 구성된 경우 NetApp Kubernetes Monitoring Operator가 필요로 하는 컨테이너에 대한 액세스를 구성해야 합니다.

NetApp 모니터링 오퍼레이터 설치 타일에서 "이미지 풀 스니펫"을 실행합니다. 이 명령은 Cloud Insights 리포지토리에 로그인하고 오퍼레이터의 모든 이미지 종속성을 풀한 다음 Cloud Insights 리포지토리에서 로그아웃합니다. 메시지가 표시되면 제공된 리포지토리 임시 암호를 입력합니다. 이 명령은 옵션 기능을 포함하여 오퍼레이터가 사용하는 모든 이미지를 다운로드합니다. 이러한 이미지가 사용되는 기능은 아래를 참조하십시오.

핵심 운영자 기능 및 Kubernetes 모니터링

- NetApp - 모니터링
- ci-kube-RBAC-프록시
- CI - KSM을 참조하십시오
- CI - 텔레그라프
- distroless-root-user.(거리 없는 루트 사용자

이벤트 로그

- CI 플루언트 비트
- ci-cubernetes-event-exporter를 의미합니다

네트워크 성능 및 맵

- CI-NET-관찰자

회사 정책에 따라 운영 Docker 이미지를 프라이빗/로컬/엔터프라이즈 Docker 저장소로 밀어 넣습니다. 리포지토리에서 이러한 이미지에 대한 이미지 태그 및 디렉터리 경로가 Cloud Insights 리포지토리의 이미지 태그 및 디렉터리 경로와 일치하는지 확인합니다.

운영자 배포에서 모니터링 - 운영자 배포를 편집합니다. YAML을 편집하고 모든 이미지 참조를 수정하여 개인 Docker 저장소를 사용하십시오.

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

operator-config.yAML에서 AgentConfiguration을 편집하여 새로운 Docker 저장소 위치를 반영하십시오. 개인 리포지토리에 대한 새 imagePullSecret을 만듭니다. 자세한 내용은 *https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/* 참조하십시오

```
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: https://docs.netapp.com/us-
en/cloudinsights/task_config_telegraf_agent_k8s.html#using-a-custom-or-
private-docker-repository
  dockerRepo: your.docker.repo/long/path/to/test
  # Optional: A docker image pull secret that maybe needed for your
private docker registry
  dockerImagePullSecret: docker-secret-name
```

OpenShift 지침

OpenShift 4.6 이상에서 실행 중인 경우 _operator-config.yAML_에서 AgentConfiguration을 편집하여 _Privileged_setting을 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 수준 높은 보안을 구현할 수 있습니다.

비밀에 대한 참고 사항

NetApp Kubernetes 모니터링 운영자가 클러스터 전체의 비밀을 볼 수 있는 권한을 제거하려면 설치하기 전에

_operator-setup.yaml_file 에서 다음 리소스를 삭제하십시오.

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

업그레이드인 경우 클러스터에서 리소스도 삭제합니다.

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

변경 분석이 활성화된 경우 _AgentConfiguration_or_operator-config.yaml_를 수정하여 변경 관리 섹션의 주석을 해제하고 변경 관리 섹션 아래에 _kindsToIgnoreFromWatch: '"비밀"'_를 포함시킵니다. 이 줄에서 작은따옴표와 큰따옴표의 존재 및 위치를 확인합니다.

```
# 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 체크섬 확인 중

Cloud Insights 에이전트 설치 프로그램은 무결성 검사를 수행하지만 일부 사용자는 다운로드한 아티팩트를 설치하거나 적용하기 전에 자체 검증을 수행하려고 할 수 있습니다. 기본 다운로드 및 설치 대신 다운로드 전용 작업을 수행하기 위해 이러한 사용자는 UI에서 가져온 에이전트 설치 명령을 편집하고 뒤에 오는 "설치" 옵션을 제거할 수 있습니다.

다음 단계를 수행하십시오.

1. 지시에 따라 Agent Installer 스니펫을 복사합니다.
2. 코드 조각을 명령 창에 붙여 넣는 대신 텍스트 편집기에 붙여 넣습니다.
3. 명령에서 뒤에 오는 "--install"을 제거합니다.
4. 텍스트 편집기에서 전체 명령을 복사합니다.
5. 이제 명령 창(작업 디렉토리)에 붙여넣고 실행합니다.
    - 다운로드 및 설치(기본값):

```
 installerName=cloudinsights-rhel_centos.sh … && sudo -E -H
./$installerName --download --install
** 다운로드 전용:
```

```
installerName=cloudinsights-rhel_centos.sh … && sudo -E -H
./$installerName --download
```

download-only 명령은 필요한 모든 아티팩트를 Cloud Insights에서 작업 디렉토리로 다운로드합니다. 아티팩트에는 다음이 포함되지만 이에 국한되지는 않습니다.

- 설치 스크립트
- 환경 파일입니다
- YAML 파일
- 서명된 체크섬 파일(SHA256.signed)
- 서명 확인을 위한 PEM 파일(NetApp_cert.pem

육안 검사를 통해 설치 스크립트, 환경 파일 및 YAML 파일을 확인할 수 있습니다.

PEM 파일의 지문이 다음과 같은 것인지 확인하여 PEM 파일을 확인할 수 있습니다.

```
 1A918038E8E127BB5C87A202DF173B97A05B4996
보다 구체적으로,
```

```
 openssl x509 -fingerprint -sha1 -noout -inform pem -in netapp_cert.pem
서명된 체크섬 파일은 PEM 파일을 사용하여 확인할 수 있습니다.
```

```
 openssl smime -verify -in sha256.signed -CAfile netapp_cert.pem -purpose
any
모든 아티팩트가 만족스럽게 확인되면 다음을 실행하여 에이전트 설치를 시작할 수 있습니다.
```

```
sudo -E -H ./<installation_script_name> --install
```

문제 해결

NetApp Kubernetes Monitoring Operator 설정 시 문제가 발생할 경우 다음과 같은 사항을 고려해야 합니다.

문제:	다음을 시도해 보십시오.
Kubernetes 영구 볼륨과 해당 백엔드 스토리지 장치 간의 하이퍼링크/연결이 표시되지 않습니다. 내 Kubernetes 영구 볼륨은 스토리지 서버의 호스트 이름을 사용하여 구성됩니다.	기존 Telegraf 에이전트를 제거한 다음 최신 Telegraf 에이전트를 다시 설치하는 단계를 따릅니다. Telegraf 버전 2.0 이상을 사용해야 하며 Kubernetes 클러스터 스토리지를 Cloud Insights에서 능동적으로 모니터링해야 합니다.
로그에 다음과 같은 메시지가 표시됩니다. E0901 15:21:39.962145 1 reflector.go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: * v1.mutatingWebhookConfiguration: 서버에서 요청한 리소스를 찾을 수 없습니다 E0901 15:21:43.168161 반사판. go:178] k8s.io/kube-state-metrics/internal/store/builder.go:352: 목록*v1. 임대: 서버가 요청된 리소스를 찾을 수 없습니다(Get leases.coordination.k8s.io). 등	이러한 메시지는 Kubernetes 버전이 1.20 미만인 경우 kube-state-metrics 버전 2.0.0 이상을 실행하는 경우에 발생할 수 있습니다. Kubernetes 버전 가져오기: *kubectl* 버전 _ *kube-state-metrics* 버전을 얻으려면*:* *_kubectl get deploy/kube-state-metrics -o jsonpath='{..image}'* 이러한 메시지가 발생하지 않도록 하기 위해 사용자는 kube-state-metrics 배포를 수정하여 다음 임대 서비스를 비활성화할 수 있습니다. *mutatingwebhookconfiguration* *validingwebhookconfiguration* _volumeAttachments 리소스 _ 보다 구체적으로 다음과 같은 CLI 인수를 사용할 수 있습니다. 리소스 = certificatesigningrequests, configmap, cronjobs, demonset, 배포, 엔드포인트, 수평적 podautoscalers, ingresses, 작업, 제한 범위, 네임스페이스, 네트워크 정책, 노드, 영구 볼륨권, podrightiondecudies, 포드, 자원 컨트롤러, 리플리케이션, 풀 소스, 서비스 기본 리소스 목록은 다음과 같습니다. "인증 요청, 구성 맵, cronjobs, demonset, 배포, 엔드포인트, 수평 포드오토칼러, ingresses, 작업, 임대, 제한 범위, mutatingwebhookconfiguration, 네임스페이스, 네트워크 정책, 노드, 지속형, 지속형, 지속형, 볼륨, 볼륨 구성, 복제, 웹후크구성, 볼륨 첨부 파일 확인"

문제:	다음을 시도해 보십시오.
Telegraf에서 다음과 유사한 오류 메시지가 표시되지만 Telegraf가 시작되고 실행됩니다. 10월 11일 14:23:41 IP-172-31-39-47 systemd [1]: 인플루엔자 DB에 메트릭을 보고하기 위한 플러그인 기반 서버 에이전트를 시작했습니다. 10월 11일 14:23:41 IP-172-31-39-47 Telegraf[1827]: time="2021-10-11T14:23:41Z" level=error msg="캐시 디렉토리를 만들지 못했습니다. /etc/Telegraf/.cache/snowflake, err:mkdir/etc/Telegraf/.ca CHE: 권한이 거부되었습니다. 무시됨 \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="를 열지 못했습니다. 무시되었습니다. /etc/Telegraf/.cache/snowflake/OCSP_response_cache.json을 엽니다. 해당 없음 파일 또는 디렉터리\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 i! 텔레그라프 1.19.3 시작	이는 알려진 문제입니다. 을 참조하십시오 "이 GitHub 기사를 참조하십시오" 를 참조하십시오. Telegraf가 실행 중인 경우 사용자는 이러한 오류 메시지를 무시할 수 있습니다.
Kubernetes에서 Telegraf POD가 다음 오류를 보고합니다. "mountstats 정보 처리 중 오류 발생: mountstats 파일을 열지 못했습니다. /hostfs/proc/1/mountstats, 오류: open/hostfs/proc/1/mountstats: permission denied"	SELinux가 설정되어 있고 강제 적용되는 경우 Telegraf 포드가 Kubernetes 노드의 /proc/1/mountstats 파일에 액세스하지 못할 수 있습니다. 이 제한을 해결하려면 agentconfiguration을 편집하고 runPrivileged 설정을 활성화하십시오. 자세한 내용은 다음을 참조하십시오. https://docs.netapp.com/us-en/cloudinsights/task_config_telegraf_agent_k8s.html#openshift-instructions.
Kubernetes에서 내 Telegraf ReplicaSet POD가 다음 오류를 보고합니다. [inputs.prometheus] 플러그인 오류: keypair /etc/kubernetes/pki/etcd/server.crt:/etc/kubernetes/pki/etcd/server.key: open/etc/Kubernetes/pki/etcd/server.crt: 해당 파일 또는 디렉토리가 없습니다	Telegraf ReplicaSet POD는 마스터나 etcd로 지정된 노드에서 실행되도록 설계되었습니다. ReplicaSet 포드가 이러한 노드 중 하나에서 실행되고 있지 않으면 이러한 오류가 발생합니다. 마스터/etcd 노드에 문제가 있는지 확인합니다. 만약 그렇다면, 텔레그라프 ReplicaSet, 텔레그라프-RS에 필요한 내약성을 추가한다. 예를 들어 ReplicaSet을 편집합니다. kubectl 편집 RS Telegraf-RS … 그리고 사양에 적절한 공차를 추가합니다. 그런 다음 ReplicaSet 포드를 다시 시작합니다.

문제:	다음을 시도해 보십시오.
PSP/PSA 환경이 있습니다. 이 문제가 모니터링 오퍼레이터에게 영향을 미칩니까?	PSP(Pod Security Policy) 또는 PSA(Pod Security Admission)를 통해 Kubernetes 클러스터를 실행 중인 경우, 최신 NetApp Kubernetes Monitoring Operator로 업그레이드해야 합니다. 다음 단계에 따라 PSP/PSA를 지원하여 현재 NKMO로 업그레이드합니다. 1. 설치 제거 이전 모니터링 오퍼레이터: kubectl delete agent-monitoring-netapp-n netapp-monitoring kubectl delete ns NetApp-monitoring kubectl 삭제 CRD agents.monitoring.netapp.com clusterrole agent-manager-role agent-proxy-role agent-metrics-reader를 삭제합니다 clusterrolebinding agent-manager-rolebinding agent-proxy-rolebinding agent-cluster-admin-rolebinding 을 삭제합니다 2. 설치합니다 모니터링 운용자의 최신 버전
NKMO를 배포하는 데 문제가 발생했고 PSP/PSA를 사용하고 있습니다.	다음 명령을 사용하여 에이전트를 편집합니다. kubectl -n <name-space> 편집 에이전트 '보안 정책 사용'을 '거짓'으로 표시합니다. 이렇게 하면 Pod 보안 정책과 Pod 보안 입장은 비활성화되고 NKMO가 배포될 수 있습니다. 다음 명령을 사용하여 확인합니다. kubectl get psp(Pod 보안 정책이 제거됨) kubectl get all -n <namespace>
grep -i psp (아무것도 발견되지 않았음을 보여줌)	"ImagePullBackoff" 오류가 표시됩니다
이러한 오류는 사용자 지정 또는 프라이빗 Docker 저장소가 있고 NetApp Kubernetes Monitoring Operator가 이를 제대로 인식하도록 구성하지 않은 경우 나타날 수 있습니다. 자세히 보기 사용자 지정/개인 저장소 구성 정보	모니터링 운영자 구축에 문제가 있는데 현재 설명서를 참조해도 문제를 해결하는 데 도움이 되지 않습니다.

문제:	다음을 시도해 보십시오.

다음 명령의 출력을 캡처하거나 기록해 두고 기술 지원 팀에 문의하십시오.
 	
`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`
	NKMO 네임스페이스의 net-observer(워크로드 맵) 포드가 CrashLoopBackOff에 있습니다

이러한 포드는 네트워크 관찰 가능성을 위한 워크로드 맵 데이터 수집기에 해당합니다. 다음을 시도해 보십시오. • 최소 커널 버전을 확인하려면 pods 중 하나의 로그를 확인하십시오. 예를 들면 다음과 같습니다. — {"ci-tenant-id":"your-tenant-id","collector-cluster":"your-k8s-cluster-name","environment":"prod","level":"error","msg":"유효성 검사에 실패했습니다. 이유: 커널 버전 3.10.0이 최소 커널 버전 4.18.0", "time":"2022-11-09T08:23:08Z"}보다 작습니다 — • Net-observer Pod를 사용하려면 Linux 커널 버전이 4.18.0 이상이어야 합니다. "uname -r" 명령을 사용하여 커널 버전을 확인하고 해당 버전이 4.18.0 이상인지 확인합니다	Pod는 NKMO 네임스페이스(기본값: NetApp 모니터링)에서 실행되지만, 쿼리의 워크로드 맵 또는 Kubernetes 메트릭의 UI에는 데이터가 표시되지 않습니다
K8S 클러스터의 노드에서 시간 설정을 확인합니다. 정확한 감사 및 데이터 보고를 위해 NTP(Network Time Protocol) 또는 SNTP(Simple Network Time Protocol)를 사용하여 Agent 시스템의 시간을 동기화하는 것이 좋습니다.	NKMO 네임스페이스의 일부 net-observer POD가 Pending 상태입니다
Net-observer는 DemonSet로, k8s 클러스터의 각 노드에서 포드를 실행합니다. • 보류 중인 Pod를 확인하고 CPU 또는 메모리에 리소스 문제가 있는지 확인합니다. 노드에서 필요한 메모리 및 CPU를 사용할 수 있는지 확인합니다.	NetApp Kubernetes 모니터링 운영자를 설치한 직후 로그에 다음이 표시됩니다. [inputs.prometheus] 플러그인 오류: HTTP 요청을 보내는 중 오류가 발생했습니다 http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: 가져오기 http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics: TCP: lookup kudo-state-metrics.<namespace>.svc.cluster.local: 해당 호스트가 없습니다

문제:	다음을 시도해 보십시오.
이 메시지는 일반적으로 새 오퍼레이터가 설치되어 있고 _Telegraf-RS_POD가 _KSM_POD가 가동되기 전에 작동 중일 때만 표시됩니다. 이러한 메시지는 모든 Pod가 실행되면 중지되어야 합니다.	클러스터에 존재하는 Kubernetes CronJobs에 대해 어떤 메트릭도 수집되지 않습니다.
Kubernetes 버전 확인(예 `kubectl version`)를 클릭합니다. v1.20.x 이하일 경우 이는 예상되는 제한 사항입니다. NetApp Kubernetes 모니터링 오퍼레이터와 함께 구축되는 Kubbe-state-metrics 릴리즈는 v1.crontjob만 지원합니다. Kubernetes 1.20.x 이하에서는 crontjob 리소스가 v1beta.crontjob에 있습니다. 따라서 kube-state-metrics는 crontjob 리소스를 찾을 수 없습니다.	운용자 설치 후, Telegraf-ds Pod는 CrashLoopBackOff로 진입하고 POD 로그는 "su:Authentication failure"를 나타낸다.
*AgentConfiguration_*에서 *Telegraf* 섹션을 편집하고 *_dockerMetricCollectionEnabled_*를 *false* 로 설정합니다. 자세한 내용은 조작자를 참조하십시오 *"구성 옵션".* 참고*: Cloud Insights Federal Edition*을 사용하는 경우, *Docker* 소켓에 액세스하려면 *Telegraf* 컨테이너를 루트로 실행하거나 *_su_*를 사용하여 *Telegraf* 사용자를 *Docker* 그룹에 추가해야 하기 때문에 *_su* 사용이 제한된 사용자는 Docker 메트릭을 수집할 수 없습니다. Docker 메트릭 수집 및 _su_의 사용은 기본적으로 활성화되어 있습니다. 두 가지를 모두 사용하지 않으려면 _AgentConfiguration_파일에서 _Telegraf.docker_entry를 제거하십시오. … 사양: … 텔레그라프: … -name: docker입니다 실행 모드: - DemonSet 대체: -key:docker_unix_sock_placeholder입니다 값: UNIX:///run/docker.sock … …	Telegraf 로그에 다음과 유사한 오류 메시지가 반복적으로 표시됩니다. E! [agent] outputs.http:Post에 쓰는 동안 오류가 발생했습니다 "<a href="https://<tenant_url>/rest/v1/lake/ingest/influxdb":" class="bare">https://<tenant_url>/rest/v1/lake/ingest/influxdb": 컨텍스트 마감일이 초과되었습니다(헤더를 기다리는 동안 클라이언트 시간 초과됨).
_AgentConfiguration_에서 Telegraf 섹션을 편집하고 _dockerMetricCollectionEnabled_를 false 로 설정합니다. 자세한 내용은 조작자를 참조하십시오 "구성 옵션".	일부 이벤트 로그에 대한 _divedobject_data가 없습니다.
의 단계를 수행했는지 확인합니다 "권한" 섹션을 참조하십시오.	두 개의 모니터링 운영자 Pod가 실행 중인 것을 볼 수 있는데, 하나는 netapp-ci-monitoring-operator-<pod>이고 다른 하나는 monitoring-operator-<pod>입니다.

문제:	다음을 시도해 보십시오.
2023년 10월 12일부터 Cloud Insights은 사용자에게 더 나은 서비스를 제공하기 위해 운영자를 리팩토링했습니다. 변경 사항을 완전히 채택하려면 반드시 필요합니다 기존 연산자를 제거합니다 및 새 장치를 장착하십시오.	내 Kubernetes 이벤트가 예기치 않게 Cloud Insights 보고에 대한 보고를 중단했습니다.
이벤트 내보내기 포드의 이름을 검색합니다. `kubectl -n netapp-monitoring get pods`	grep event-exporter
awk '{print $1}'	sed 's/event-exporter./event-exporter/'` "netapp-ci-event-exporter" 또는 "event-exporter"여야 합니다. 그런 다음 모니터링 에이전트를 편집합니다 `kubectl -n netapp-monitoring edit agent`를 입력하고 log_file 의 값을 이전 단계에서 찾은 적절한 이벤트 내보내기 포드 이름을 반영하도록 설정합니다. 보다 구체적으로 log_file을 "/var/log/containers/netapp-ci-event-exporter.log" 또는 "/var/log/containers/event-exporter **.log"**로 설정해야 합니다. **….** **fluent-bit:** **…** **- name: event-exporter-ci** **substitutions:** **- key: LOG_FILE** **values:** **- /var/log/containers/netapp-ci-event-exporter**.log … …. 또는, 하나를 할 수도 있습니다 설치 제거 및 다시 설치합니다 에이전트
리소스 부족으로 인해 NetApp Kubernetes Monitoring Operator에 의해 구축된 Pod가 충돌하는 것을 볼 수 있습니다.	자세한 내용은 NetApp Kubernetes 모니터링 오퍼레이터를 참조하십시오 "구성 옵션" 필요한 경우 CPU 및/또는 메모리 제한을 늘립니다.

추가 정보는 에서 찾을 수 있습니다 "지원" 페이지 또는 에 있습니다 "Data Collector 지원 매트릭스".

NetApp Kubernetes 모니터링 오퍼레이터 구성 옵션

를 클릭합니다 "NetApp Kubernetes 모니터링 운영자" 설치 및 구성을 사용자 지정할 수 있습니다.

아래 표에는 AgentConfiguration 파일에 사용할 수 있는 옵션이 나와 있습니다.

구성 요소	옵션을 선택합니다	설명
에이전트		운용자가 설치할 수 있는 모든 구성품에 공통으로 적용되는 구성 옵션. 이러한 옵션은 "글로벌" 옵션으로 간주할 수 있습니다.
	dockerRepo	Cloud Insights Docker repo와 비교하여 dockerRepo를 재정의하여 고객의 프라이빗 Docker Repos에서 이미지를 가져올 수 있습니다. 기본값은 Cloud Insights Docker입니다
	dockerImagePullSecret	선택 사항: 고객의 비공개 리포의 비밀
	클러스터 이름	모든 고객 클러스터에서 클러스터를 고유하게 식별하는 자유 텍스트 필드입니다. 이는 Cloud Insights 테넌트 전체에서 고유해야 합니다. 기본값은 고객이 "클러스터 이름" 필드의 UI에 입력하는 것입니다
	프록시 형식: 프록시: 서버: 포트: 사용자 이름: 암호: 프록시 없음: 텔레그라프프록시 사용: isAuProxyEnabled: 사용 안함: isFluentbitProxyEnabled: isCollectorProxyEnabled(isCollectorProxyEnabled):	프록시를 설정하는 선택 사항입니다. 일반적으로 고객의 기업 대리인입니다.
텔레그라프		작업자의 Telegraf 설치를 사용자 정의할 수 있는 구성 옵션
	수집 간격	메트릭 수집 간격(초)(최대 = 60초)
	dsCpuLimit 를 선택합니다	Telegraf DS의 CPU 제한
	dsMemLimit	Telegraf DS의 메모리 제한
	dsCpuRequest 를 참조하십시오	Telegraf DS에 대한 CPU 요청
	dsMemRequest입니다	Telegraf DS에 대한 메모리 요청
	rsCpuLimit	Telegraf RS의 CPU 제한
	메모리 제한	Telegraf RS의 메모리 제한
	rsCpuRequest 를 참조하십시오	Telegraf RS에 대한 CPU 요청
	rsMemRequest입니다	Telegraf RS에 대한 메모리 요청

구성 요소	옵션을 선택합니다	설명
	dockerMountPoint를 참조하십시오	dockerMountPoint 경로에 대한 재정의입니다. 이는 클라우드 파운드리 같은 k8s 플랫폼에 비표준 Docker 설치를 위한 것입니다
	dockerUnixSocket 을 참조하십시오	dockerUnixSocket 경로에 대한 재정의입니다. 이는 클라우드 파운드리 같은 k8s 플랫폼에 비표준 Docker 설치를 위한 것입니다.
	crioSockPath를 참조하십시오	crioSockPath 경로에 대한 재정의입니다. 이는 클라우드 파운드리 같은 k8s 플랫폼에 비표준 Docker 설치를 위한 것입니다.
	runPrivileged(권한이	권한 모드에서 Telegraf 컨테이너를 실행합니다. k8s 노드에서 SELinux가 설정된 경우 이 값을 TRUE로 설정합니다
	BatchSize(부atchSize)	을 참조하십시오 "Telegraf 구성 문서"
	백혈구 한계	을 참조하십시오 "Telegraf 구성 문서"
	RoundInterval(라운드 간격	을 참조하십시오 "Telegraf 구성 문서"
	콜렉션 지터	을 참조하십시오 "Telegraf 구성 문서"
	정밀도	을 참조하십시오 "Telegraf 구성 문서"
	FlushInterval(플러시간격	을 참조하십시오 "Telegraf 구성 문서"
	플러시지터	을 참조하십시오 "Telegraf 구성 문서"
	출력 제한 시간	을 참조하십시오 "Telegraf 구성 문서"
	dockerMetricCollectionEnabled를 참조하십시오	Docker 메트릭을 수집합니다. 기본적으로 이 옵션은 true로 설정되어 있으며, Docker 기반 k8s 구축 시 Docker 메트릭이 수집됩니다. Docker 메트릭 수집을 해제하려면 FALSE로 설정합니다.
	dsTolerations 를 선택합니다	텔레그라프 - DS 추가 허용.
	RsTolerations를 선택합니다	Telegraf-RS 추가 허용.
Kudbe-state-metrics를 나타냅니다		작업자의 kudbe 상태 메트릭 설치를 사용자 지정할 수 있는 구성 옵션입니다
	cpuLimit	kubbe-state-metrics 구축을 위한 CPU 제한입니다
	MemLimit	kubbe-state-metrics 구축을 위한 MEM 한도
	cpuRequest입니다	kubbe 상태 메트릭 구축을 위한 CPU 요청입니다
	MemRequest입니다	MEM은 kudo 상태 메트릭 배포를 요청합니다
	리소스	캡처할 리소스의 쉼표로 구분된 목록입니다. 예: cronjobs, demonset, 배포, 링스, 작업, 네임스페이스, 노드, persistentvolumeclaims, persistentvolumes, dPOD, replicasets, resourcequotas, 서비스, statefulsets
	공차	Kudbe-state-metrics 추가 공약입니다.

구성 요소	옵션을 선택합니다	설명
	라벨	kuba-state-metrics가 캡처해야 하는 쉼표로 구분된 리소스 목록입니다 예: cronjobs=[*], demonsets=[*], 배포=[*], 링스=[*], 작업=[*], 네임스페이스=[*], 노드=[*], persistentvolumeclaims=[*], persistentvolumes=[*], pod=[*], replicasets=[*], resourcequotas=[*], services=[*], statefulsets=[*]
로그		운용자의 로그 수집 및 설치를 사용자 정의할 수 있는 구성 옵션입니다
	readFromHead(readFromHead	참/거짓, 유창한 비트가 로그에서 로그를 읽어야 합니다
	시간 초과	시간 초과(초
	dnsMode를 선택합니다	TCP/UDP, DNS 모드
	유창한 비트 내약성	Fluent-bit-DS 추가 허용.
	이벤트-수출자-내약성	이벤트-수출자 추가 허용.
워크로드 맵		작업자의 작업량 맵 수집 및 설치를 사용자 정의할 수 있는 구성 옵션입니다
	cpuLimit	순 관찰자 DS에 대한 CPU 제한입니다
	MemLimit	순 관찰자 DS에 대한 MEM 한도
	cpuRequest입니다	net observer DS에 대한 CPU 요청입니다
	MemRequest입니다	net observer DS에 대한 MEM 요청
	MetricAggregationInterval입니다	메트릭 집계 간격(초
	bpfPollInterval입니다	BPF 폴링 간격(초
	enableDNSLookup	True/false, DNS 조회를 사용하도록 설정합니다
	L4-공차	NET-observer-L4-DS 추가 허용 오차
	runPrivileged(권한이	참/거짓 - Kubernetes 노드에서 SELinux가 활성화된 경우 runprivileged 를 true 로 설정합니다.
변경 관리		Kubernetes 변경 관리 및 분석에 대한 구성 옵션
	cpuLimit	change-observer-watch-RS에 대한 CPU 제한값입니다
	MemLimit	change-observer-watch-RS에 대한 MEM 한계
	cpuRequest입니다	change-observer-watch-RS에 대한 CPU 요청입니다
	MemRequest입니다	change-observer-watch-RS에 대한 MEM 요청
	FailureDeclarationIntervalMins 를 참조하십시오	실패한 워크로드 배포가 실패로 표시되는 간격(분)입니다

구성 요소	옵션을 선택합니다	설명
	deployAggrIntervalSeconds입니다	작업 부하 배포 진행 중 이벤트가 전송되는 빈도입니다
	비작업 로드 AggrIntervalSeconds입니다	비워크로드 구축이 결합되고 전송되는 빈도입니다
	TERmsToRedact 를 참조하십시오	env 이름 및 데이터 맵에서 사용되는 정규식 집합으로, 값이 교정됩니다 예제 용어: "pwd", "password", "token", "apikey", "api-key", "JWT"
	AditionalKindsToWatch 를 참조하십시오	수집기에서 감시하는 기본 종류 집합에서 볼 수 있는 추가 종류의 쉼표로 구분된 목록
	KindsToIgnoreFromWatch 를 참조하십시오	수집기에서 감시하는 기본 종류의 집합에서 감시하는 것을 무시할 수 있는 쉼표로 구분된 종류의 목록입니다
	LogRecordAggrIntervalSeconds입니다	수집기에서 CI로 로그 레코드를 보내는 빈도입니다
	시계의 내약성	change-observer-watch-DS 추가 허용 오차. 축약된 단일 선 형식만 해당. 예: '{key:taint1,operator:exists,effect:NoSchedule}, {key:taint2,operator:exists,effect:NoExecute}'

AgentConfiguration 파일 예

다음은 AgentConfiguration 파일의 예입니다.

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

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

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

    # # [Required Field] By default, the operator uses the CI repository.
    # # To use a private repository, change this field to your repository
name.
    # # Please see documentation here: https://docs.netapp.com/us-
en/cloudinsights/task_config_telegraf_agent_k8s.html#using-a-custom-or-
private-docker-repository
    dockerRepo: 'DOCKER_REPO_PLACEHOLDER'
    # # [Required Field] The name of the imagePullSecret for dockerRepo.
    # # If you are using a private repository, change this field from
'docker' to the name of your secret.
    {{ if not (contains .Values.config.cloudType "aws") }}# {{ end -}}
    dockerImagePullSecret: 'docker'

    # # Allow the operator to automatically rotate its ApiKey before
expiration.
    # tokenRotationEnabled: '{{
.Values.telegraf_installer.kubernetes.rs.shim_token_rotation  }}'
    # # Number of days before expiration that the ApiKey should be
rotated. This must be less than the total ApiKey duration.
    # tokenRotationThresholdDays: '{{
.Values.telegraf_installer.kubernetes.rs.shim_token_rotation_threshold_day
s  }}'

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

```
    # # The default time telegraf will wait between inputs for all plugins
(interval). Max=60
    # collectionInterval: '{{
.Values.telegraf_installer.agent_resources.collection_interval }}'
    # # Maximum number of records per output that telegraf will write in
one batch (metric_batch_size).
    # batchSize: '{{
.Values.telegraf_installer.agent_resources.metric_batch_size }}'
    # # Maximum number of records per output that telegraf will cache
pending a successful write (metric_buffer_limit).
    # bufferLimit: '{{
.Values.telegraf_installer.agent_resources.metric_buffer_limit }}'
    # # Collect metrics on multiples of interval (round_interval).
    # roundInterval: '{{
.Values.telegraf_installer.agent_resources.round_interval }}'
    # # Each plugin waits a random amount of time between the scheduled
collection time and that time + collection_jitter before collecting inputs
(collection_jitter).
    # collectionJitter: '{{
.Values.telegraf_installer.agent_resources.collection_jitter }}'
    # # Collected metrics are rounded to the precision specified. When set
to "0s" precision will be set by the units specified by interval
(precision).
    # precision: '{{ .Values.telegraf_installer.agent_resources.precision
}}'
    # # Time telegraf will wait between writing outputs (flush_interval).
Max=collectionInterval
    # flushInterval: '{{
.Values.telegraf_installer.agent_resources.flush_interval }}'
    # # Each output waits a random amount of time between the scheduled
write time and that time + flush_jitter before writing outputs
(flush_jitter).
    # flushJitter: '{{
.Values.telegraf_installer.agent_resources.flush_jitter }}'
    # # Timeout for writing to outputs (timeout).
    # outputTimeout: '{{
.Values.telegraf_installer.http_output_plugin.timeout }}'

    # # telegraf-ds CPU/Mem limits and requests.
    # # See https://kubernetes.io/docs/concepts/configuration/manage-
resources-containers/
    dsCpuLimit: '{{
.Values.telegraf_installer.telegraf_resources.ds_cpu_limits  }}'
    dsMemLimit: '{{
.Values.telegraf_installer.telegraf_resources.ds_mem_limits  }}'
    dsCpuRequest: '{{
```

```
.Values.telegraf_installer.telegraf_resources.ds_cpu_request  }}'
    dsMemRequest: '{{
.Values.telegraf_installer.telegraf_resources.ds_mem_request  }}'

    # # telegraf-rs CPU/Mem limits and requests.
    rsCpuLimit: '{{
.Values.telegraf_installer.telegraf_resources.rs_cpu_limits  }}'
    rsMemLimit: '{{
.Values.telegraf_installer.telegraf_resources.rs_mem_limits  }}'
    rsCpuRequest: '{{
.Values.telegraf_installer.telegraf_resources.rs_cpu_request  }}'
    rsMemRequest: '{{
.Values.telegraf_installer.telegraf_resources.rs_mem_request  }}'

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

    # # Set runPrivileged to true if SELinux is enabled on your Kubernetes
nodes.
    # runPrivileged: 'false'

    # # Collect NFS IO metrics.
    # dsNfsIOEnabled: '{{
.Values.telegraf_installer.kubernetes.ds.shim_nfs_io_processing }}'

    # # Collect kubernetes.system_container metrics and objects in the
kube-system|cattle-system namespaces for managed kubernetes clusters (EKS,
AKS, GKE, managed Rancher).  Set this to true if you want collect these
metrics.
    # managedK8sSystemMetricCollectionEnabled: '{{
.Values.telegraf_installer.kubernetes.shim_managed_k8s_system_metric_colle
ction }}'

    # # Collect kubernetes.pod_volume (pod ephemeral storage) metrics.
Set this to true if you want to collect these metrics.
    # podVolumeMetricCollectionEnabled: '{{
.Values.telegraf_installer.kubernetes.shim_pod_volume_metric_collection
}}'

    # # Declare Rancher cluster as managed.  Set this to true if your
```

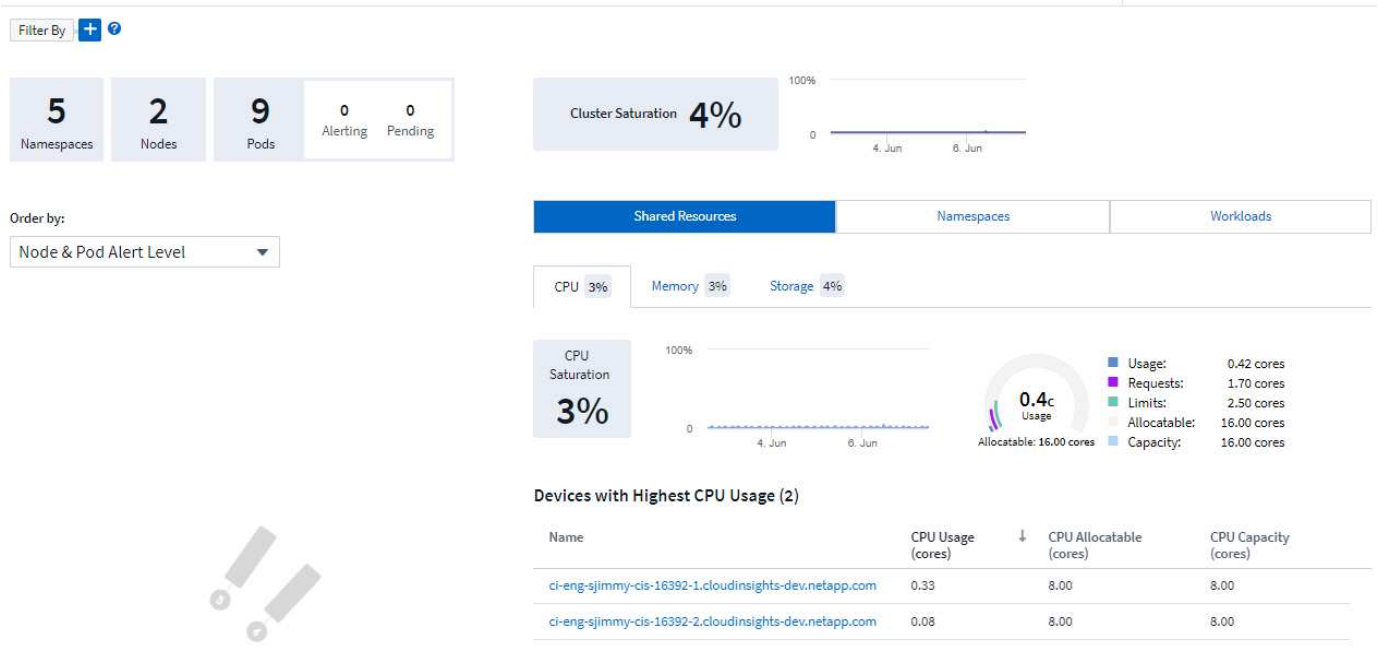
```
Rancher cluster is managed as opposed to on-premise.
    # isManagedRancher: '{{
.Values.telegraf_installer.kubernetes.is_managed_rancher }}'

  # kube-state-metrics:
    # # kube-state-metrics CPU/Mem limits and requests. By default, when
unset, kube-state-metrics has no CPU/Mem limits nor request.
    # cpuLimit:
    # memLimit:
    # cpuRequest:
    # memRequest:

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

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

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

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

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

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

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

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

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

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

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

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

  # change-management:
    # # change-observer-watch-rs CPU/Mem limits and requests.
    # # See https://kubernetes.io/docs/concepts/configuration/manage-
resources-containers/
```

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

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

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

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

    # # A set of regular expressions used in env names and data maps whose
value will be redacted
    # termsToRedact: '"pwd", "password", "token", "apikey", "api-key",
"jwt"'

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

    # # A comma separated list of 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'
    # kindsToIgnoreFromWatch: ''

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

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

Kubernetes 클러스터 세부 정보 페이지

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



네임스페이스, 노드 및 Pod 수

페이지 맨 위의 숫자는 클러스터의 네임스페이스, 노드 및 포드의 총 개수와 현재 경고 및 보류 중인 포드의 수를 보여줍니다.

공유 리소스 및 채도

상세 페이지의 오른쪽 위에는 현재 비율로 클러스터 채도가 표시되고, 시간의 경과에 따른 최근 추세를 보여 주는 그래프가 표시됩니다. 클러스터 포화도는 각 시점마다 CPU, 메모리 또는 스토리지 포화도가 가장 높은 것입니다.

그 아래 페이지는 기본적으로 CPU, 메모리 및 스토리지 탭이 있는 * 공유 리소스 * 사용을 보여줍니다. 각 탭은 시간에 따른 채도 백분율 및 추세를 추가 사용 세부 정보와 함께 표시합니다. 스토리지의 경우 표시된 값은 독립적으로 계산되는 백엔드 및 파일 시스템 포화도의 값입니다.

사용량이 가장 높은 장치는 하단의 테이블에 표시됩니다. 이 장치를 탐색하려면 아무 링크나 클릭하십시오.

네임스페이스

Namespaces 탭에는 Kubernetes 환경의 모든 네임스페이스 목록이 표시됩니다. 이 목록에는 CPU 및 메모리 사용량과 각 네임스페이스의 워크로드 수가 표시됩니다. 이름 링크를 클릭하여 각 네임스페이스를 탐색합니다.

Shared Resources | **Namespaces** | Workloads

Namespaces (5)

Name ↓	CPU Usage (cores)	Memory Usage (GiB)	Workload Count
netapp-monitoring	0.25	0.38	4
kube-system	0.01	0.03	3
kube-public	0.00	0.00	0
kube-node-lease	0.00	0.00	0
default	0.00	<0.01	1

워크로드

마찬가지로 워크로드 탭에는 각 네임스페이스의 워크로드 목록이 표시되며 CPU 및 메모리 사용량이 표시됩니다. Namespace 를 클릭하면 각 에 드릴이 연결됩니다.

Shared Resources | Namespaces | **Workloads**

Workloads (8)

Name ↓	CPU Usage (cores)	Memory Usage (GiB)	Namespace
telegraf-rs-lf9gg	0.24	0.24	netapp-monitoring
telegraf-ds-k957c	0.01	0.10	netapp-monitoring
nginx	0.00	<0.01	default
monitoring-operator-6fcf4755ff-p2cs6	<0.01	0.02	netapp-monitoring
metrics-server-7b4f8b595-f7j9f	<0.01	0.01	kube-system
local-path-provisioner-64d457c485-289gx	<0.01	0.01	kube-system
kube-state-metrics-7995866f8c-t8c49	<0.01	0.01	netapp-monitoring
coredns-5d69dc75db-nkw5p	<0.01	0.01	kube-system

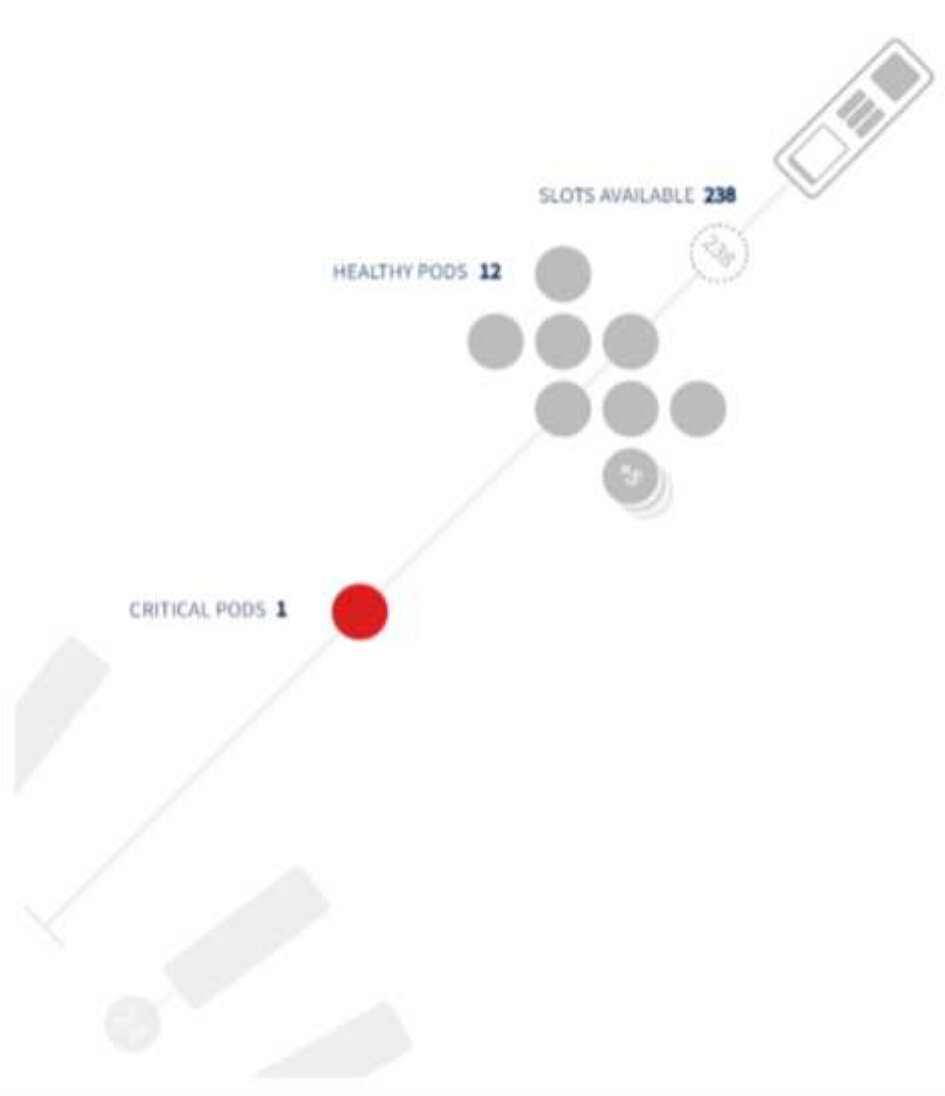
클러스터 "휠"

클러스터 "Wheel" 섹션은 노드 및 POD 상태를 한눈에 파악할 수 있도록 하여 자세한 정보를 확인할 수 있습니다. 클러스터에 페이지의 이 영역에 표시할 수 있는 것보다 많은 노드가 포함된 경우 사용 가능한 버튼을 사용하여 휠을 돌릴 수 있습니다.

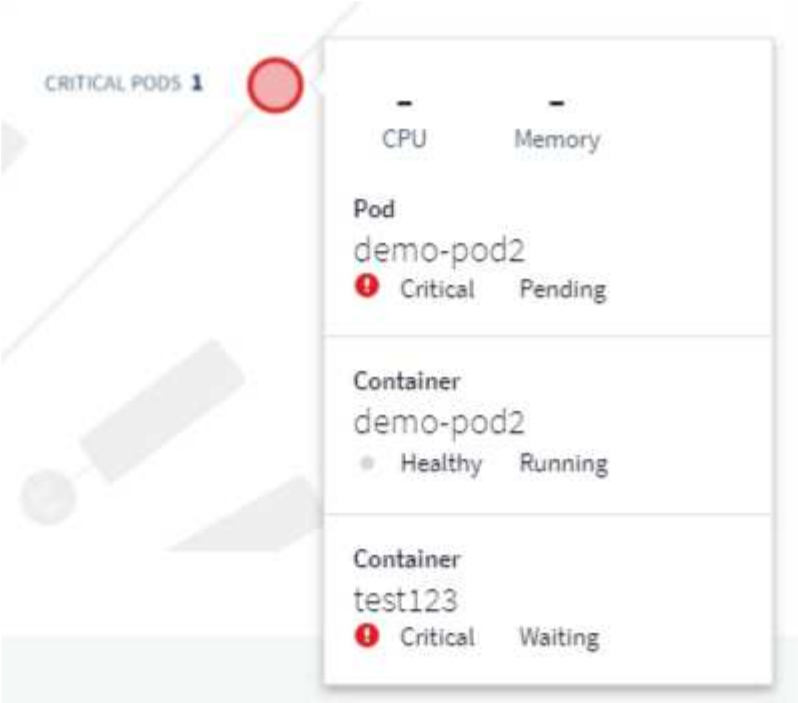
경고 Pod 또는 노드가 빨간색으로 표시됩니다. "경고" 영역은 주황색으로 표시됩니다. 예약되지 않은(즉, 연결되지 않은) 포드는 클러스터 "휠"의 하단 모서리에 표시됩니다.

POD(원) 또는 Node(막대) 위로 마우스를 이동하면 노드의 보기가 확장됩니다.

해당 보기에서 Pod 또는 노드를 클릭하면 확장된 노드 보기로 확대됩니다.



여기에서 요소 위로 마우스를 가져가면 해당 요소에 대한 세부 정보가 표시됩니다. 예를 들어, 이 예제에서 중요 POD 위에 마우스를 놓으면 해당 POD에 대한 세부 정보가 표시됩니다.



노드 요소 위로 마우스를 이동하면 파일 시스템, 메모리 및 CPU 정보를 볼 수 있습니다.

게이지에 대한 참고 사항

Memory 및 CPU 게이지는 _allocatable capacity_와 _total capacity_에 대해 _used_를 표시하기 때문에 세 가지 색상으로 표시됩니다.

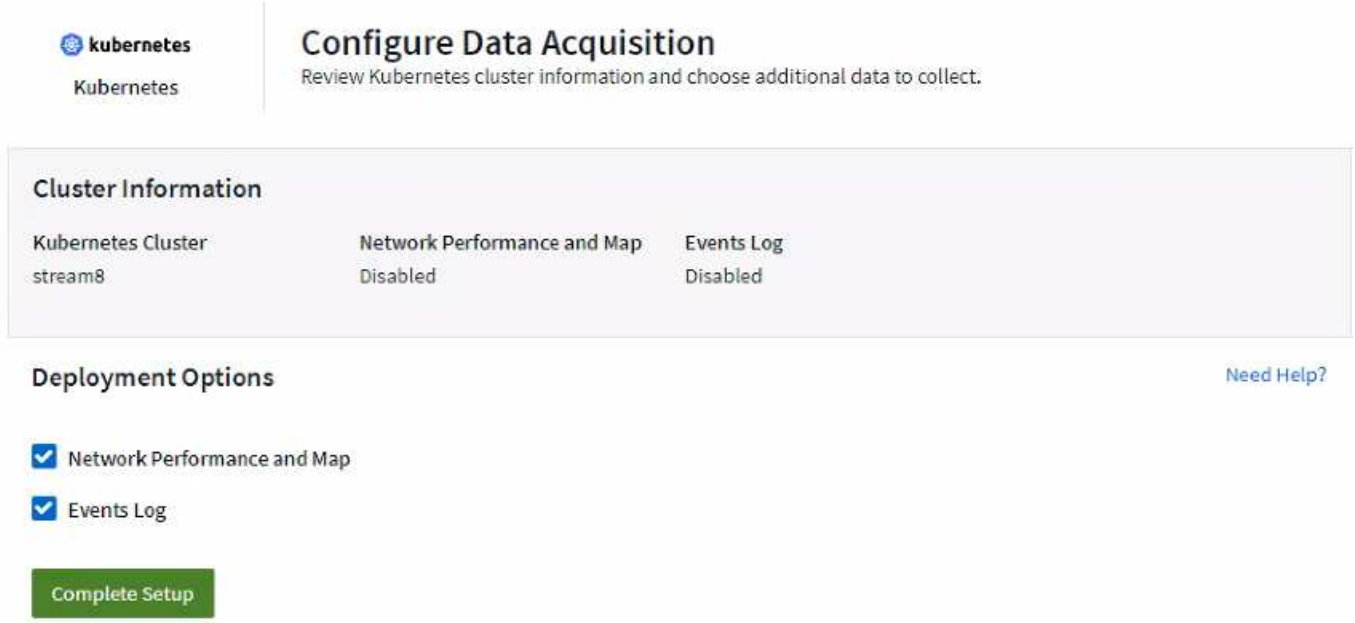
Kubernetes 네트워크 성능 모니터링 및 맵

Kubernetes 네트워크 성능 모니터링 및 맵 기능은 서비스 간 종속성(워크로드라고도 함)을 매핑하여 문제 해결을 간소화하고 네트워크 성능 지연 시간 및 이상 징후를 실시간으로 확인하여 사용자에게 영향을 미치기 전에 성능 문제를 식별합니다.
이 기능은 조직이 Kubernetes 트래픽 흐름을 분석하고 감사하여 전체 비용을 절감할 수 있도록 도와줍니다.

주요 기능: • 워크로드 맵은 Kubernetes 워크로드 종속성 및 흐름을 제공하고 네트워크 및 성능 문제를 강조합니다. • Kubernetes Pod, 워크로드 및 노드 간의 네트워크 트래픽을 모니터링하고, 트래픽 및 지연 문제의 원인을 식별합니다. • 수신, 송신, 지역 간 및 교차 영역 네트워크 트래픽을 분석하여 전체 비용을 절감합니다.

필수 구성 요소

Kubernetes Network Performance Monitoring and Map을 사용하려면 먼저 을 구성해야 합니다 "NetApp Kubernetes 모니터링 운영자" 를 눌러 이 옵션을 활성화합니다. 오퍼레이터 배포 중에 "네트워크 성능 및 맵" 확인란을 선택하여 활성화합니다. Kubernetes 랜딩 페이지로 이동하여 "배포 수정"을 선택하여 이 옵션을 활성화할 수도 있습니다.



모니터

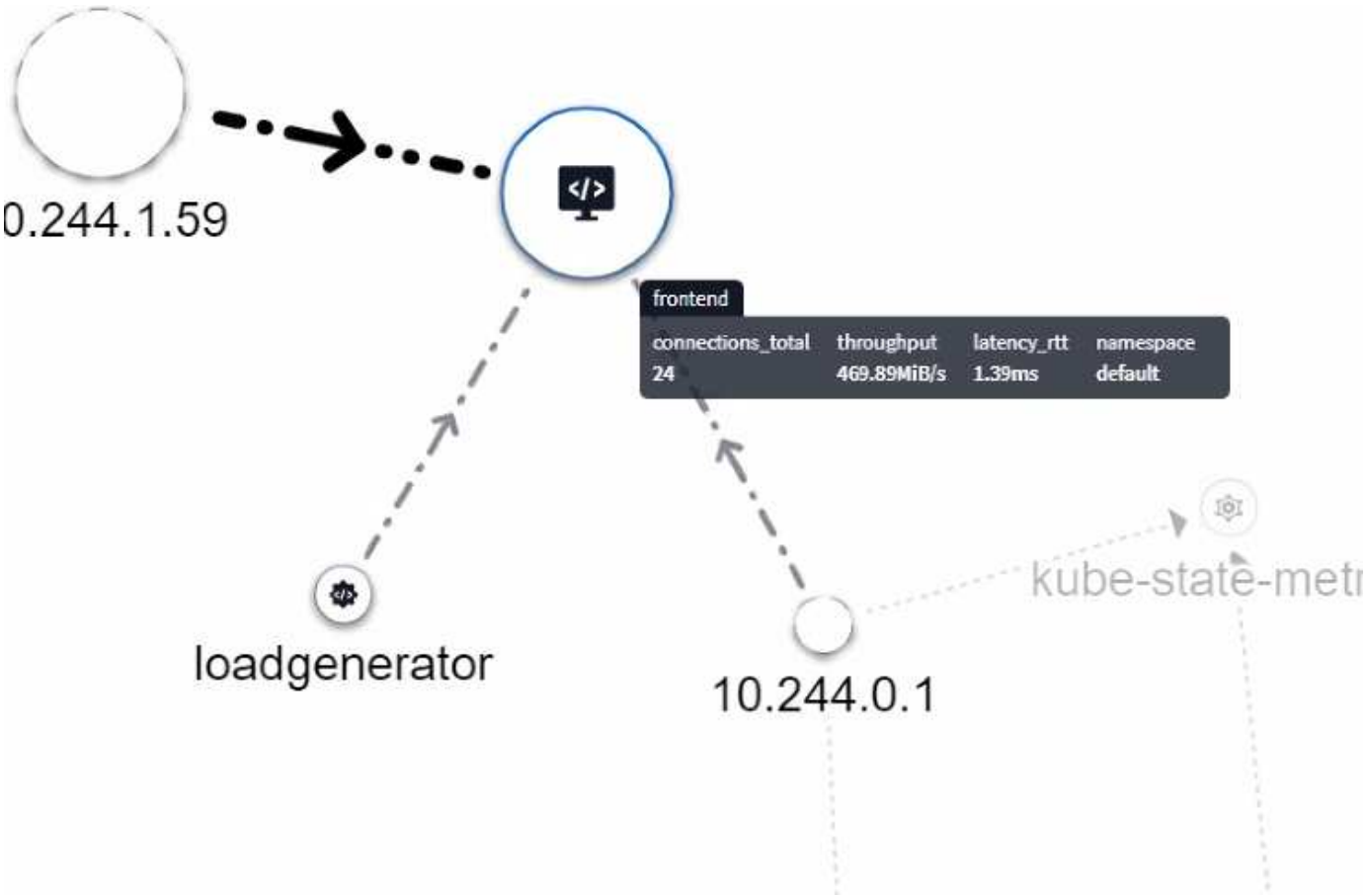
워크로드 맵은 을 사용합니다 "모니터" 정보를 도출합니다. Cloud Insights에서는 다양한 기본 Kubernetes Monitor를 제공합니다(기본적으로 일시 중지됨_일 수 있음). 원하는 모니터를 _*Resume*(예: 활성화)하거나, Kubernetes 객체에 대한 사용자 지정 모니터를 생성할 수 있습니다. 이 경우 워크로드 맵도 사용됩니다.

아래 개체 유형에 대해 Cloud Insights 메트릭 알림을 생성할 수 있습니다. 데이터가 기본 개체 유형별로 그룹화되어 있는지 확인합니다.

- Kubernetes 워크로드
- Kubernetes 발병
- kubernetes.deployment
- cub네티스.crontjob을 제공합니다
- Kubernetes, 작업
- 복제 복제
- Kubernetes.statefulset입니다
- 선택하십시오
- kubernetes.network_traffic_l4

지도

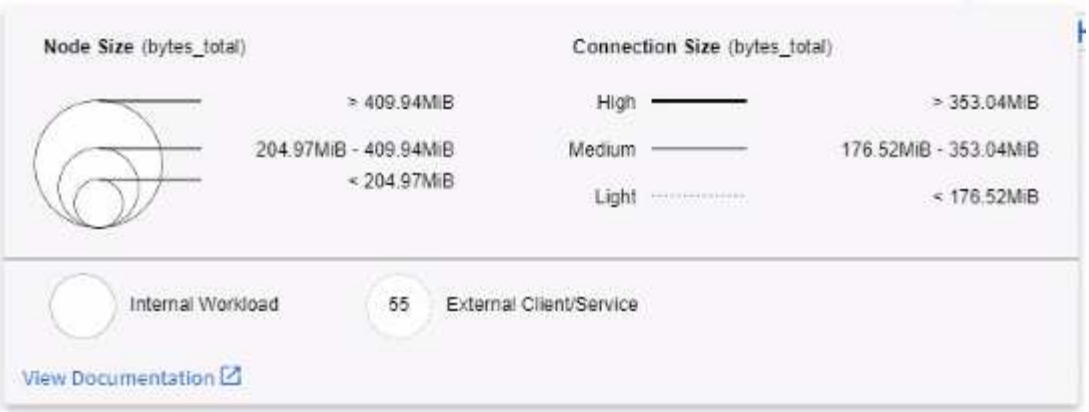
지도에는 서비스/작업 부하 및 상호 관계가 표시됩니다. 화살표는 교통 방향을 표시합니다. 작업 부하 위로 마우스를 가져가면 다음 예에서와 같이 해당 작업 부하에 대한 요약 정보가 표시됩니다.



원 안에 있는 아이콘은 다양한 서비스 유형을 나타냅니다. 아이콘은 기본 개체에 있는 경우에만 표시됩니다 라벨.

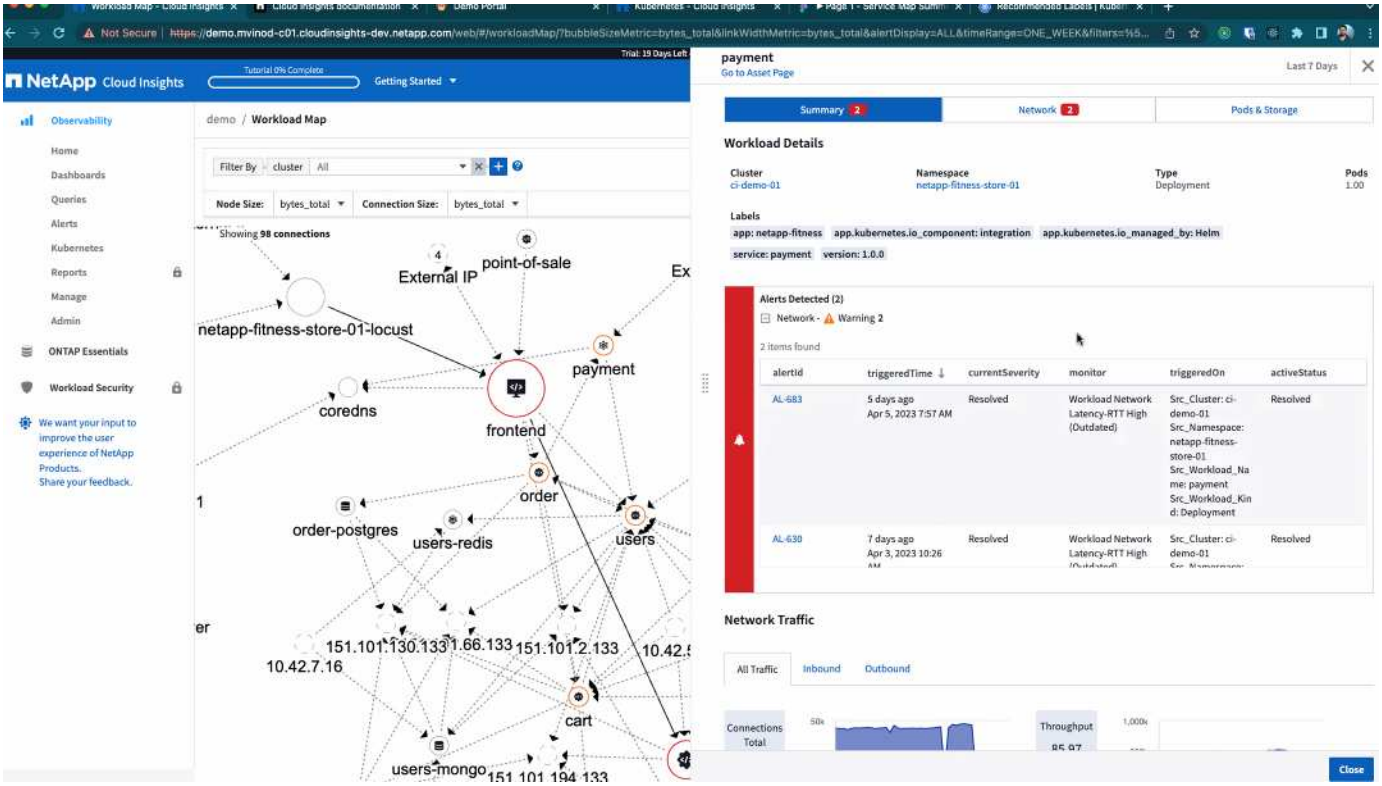
각 원의 크기는 노드 크기를 나타냅니다. 이러한 크기는 상대적이며 브라우저 확대/축소 수준 또는 화면 크기가 실제 원형 크기에 영향을 줄 수 있습니다. 같은 방법으로, 트래픽 회선 스타일은 연결 크기를 한 눈에 볼 수 있게 합니다. 굵은 실선은 트래픽이 높고 밝은 점선은 트래픽이 적습니다.

원 안에 있는 숫자는 서비스에서 현재 처리 중인 외부 연결의 수입니다.



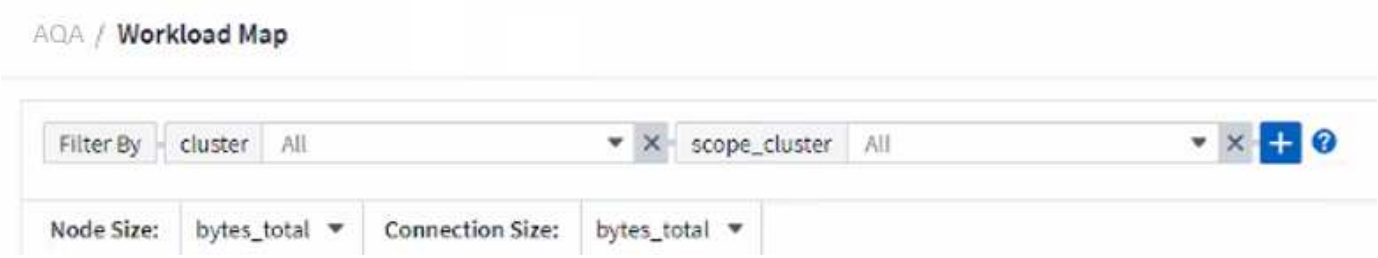
워크로드 세부 정보 및 알림

색상으로 표시된 원은 워크로드에 대한 경고 또는 위험 수준 경고를 나타냅니다. 원 위로 마우스를 가져가면 문제 요약이 표시됩니다. 원을 클릭하여 자세히 표시된 슬라이드 아웃 패널을 엽니다.

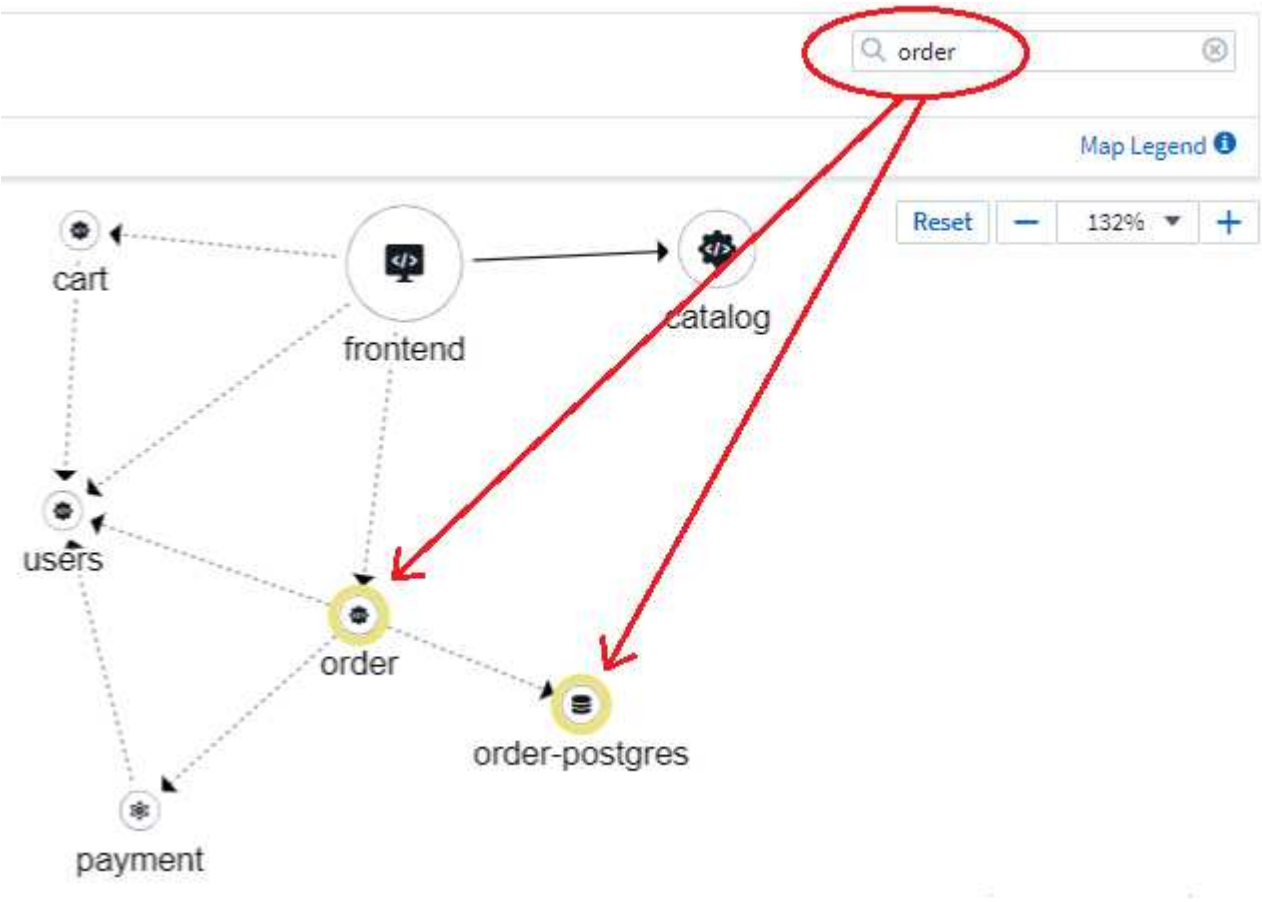


찾기 및 필터링

다른 Cloud Insights 기능과 마찬가지로 원하는 특정 오브젝트 또는 워크로드 특성에 초점을 맞춰 필터를 쉽게 설정할 수 있습니다.



마찬가지로 *Find* 필드에 문자열을 입력하면 일치하는 워크로드가 강조 표시됩니다.



워크로드 레이블

지도에서 표시되는 워크로드 유형(예: 원 아이콘)을 식별하려면 워크로드 레이블이 필요합니다. 레이블은 다음과 같이 파생됩니다.

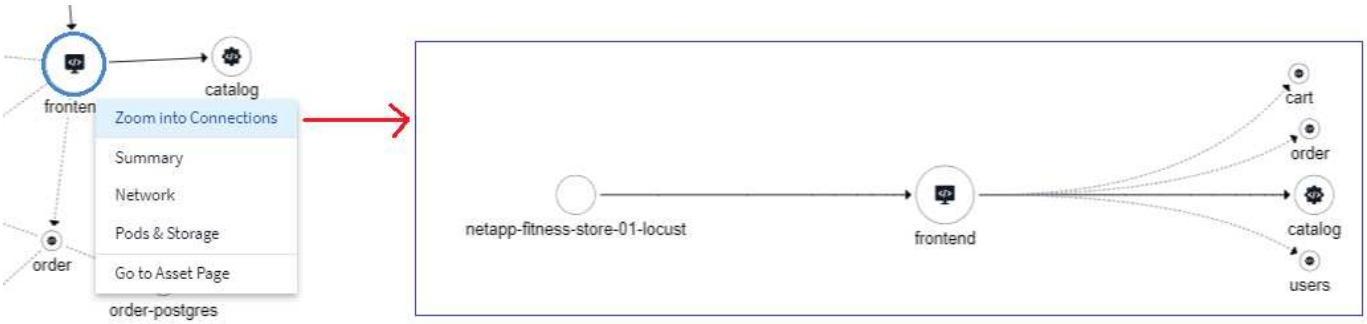
- 일반 용어로 실행 중인 서비스/애플리케이션의 이름입니다
- 소스가 POD인 경우:
  - 레이블은 POD의 워크로드 레이블에서 파생됩니다
  - 작업 부하에 대한 예상 레이블: app.Kubernetes.io/component
  - 라벨 이름 참조: https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels/
  - 권장 라벨:
    - 프런트 엔드

    - 백엔드
    - 데이터베이스
    - 캐시
    - 대기열
    - 카프카
- 소스가 Kubernetes 클러스터 외부에 있는 경우:
  - Cloud Insights는 DNS 확인된 이름을 구문 분석하여 서비스 유형을 추출하려고 시도합니다.

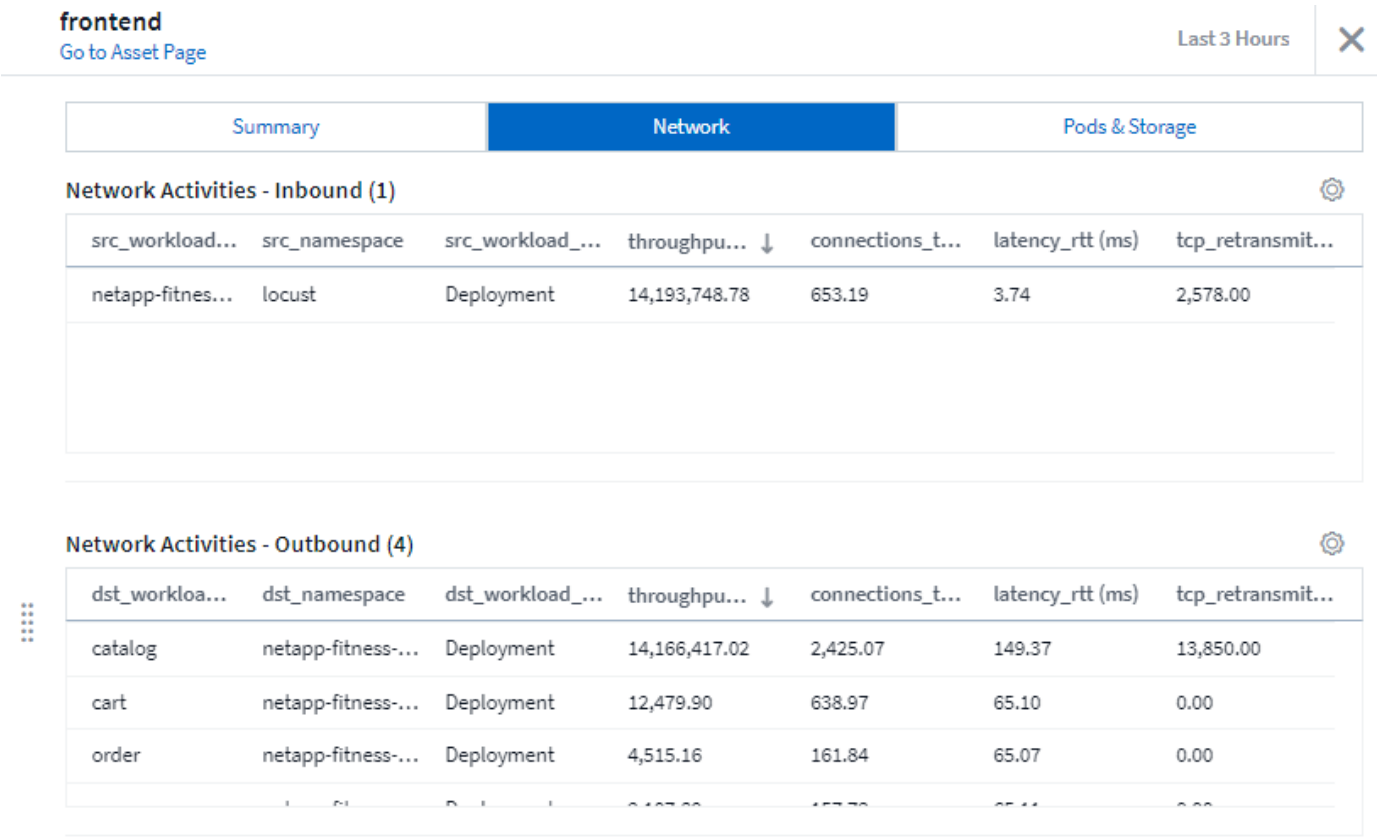
    예를 들어, DNS Resolved 이름이 _s3.eu-north-1.amazonaws.com _ 인 경우 Resolved 이름은 서비스 유형으로 get_s3_로 구문 분석됩니다.

깊이 잠들어 보세요

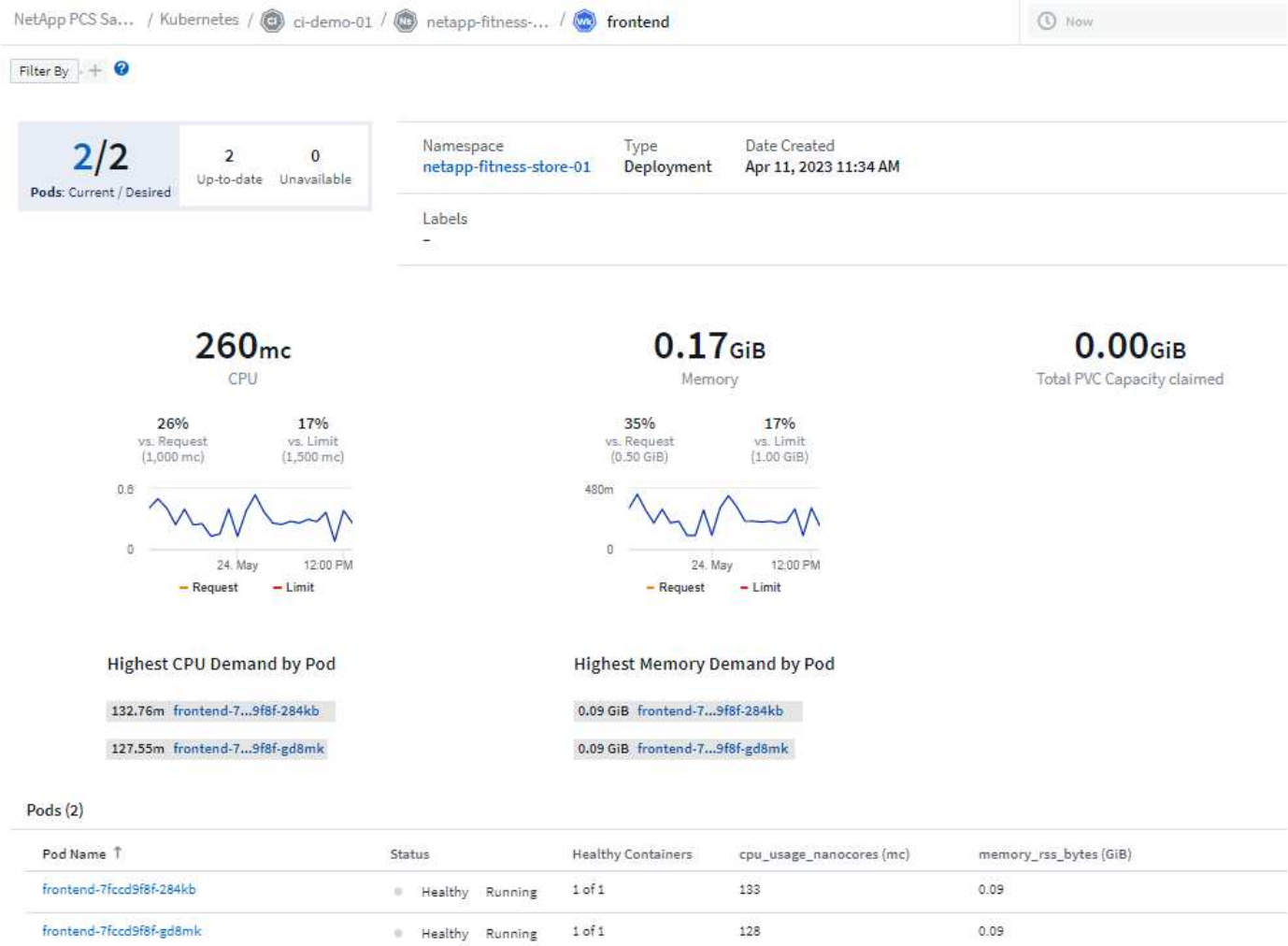
워크로드를 마우스 오른쪽 버튼으로 클릭하면 더 자세히 살펴볼 수 있는 추가 옵션이 제공됩니다. 예를 들어, 여기에서 해당 워크로드의 연결을 확대하여 볼 수 있습니다.



또는 세부 정보 슬라이드 아웃 패널을 열어 *Summary*, *Network* 또는 *Pod & Storage* 탭을 직접 볼 수 있습니다.



마지막으로, _자산 페이지로 이동 _ 을(를) 선택하면 워크로드에 대한 상세 자산 랜딩 페이지가 열립니다.



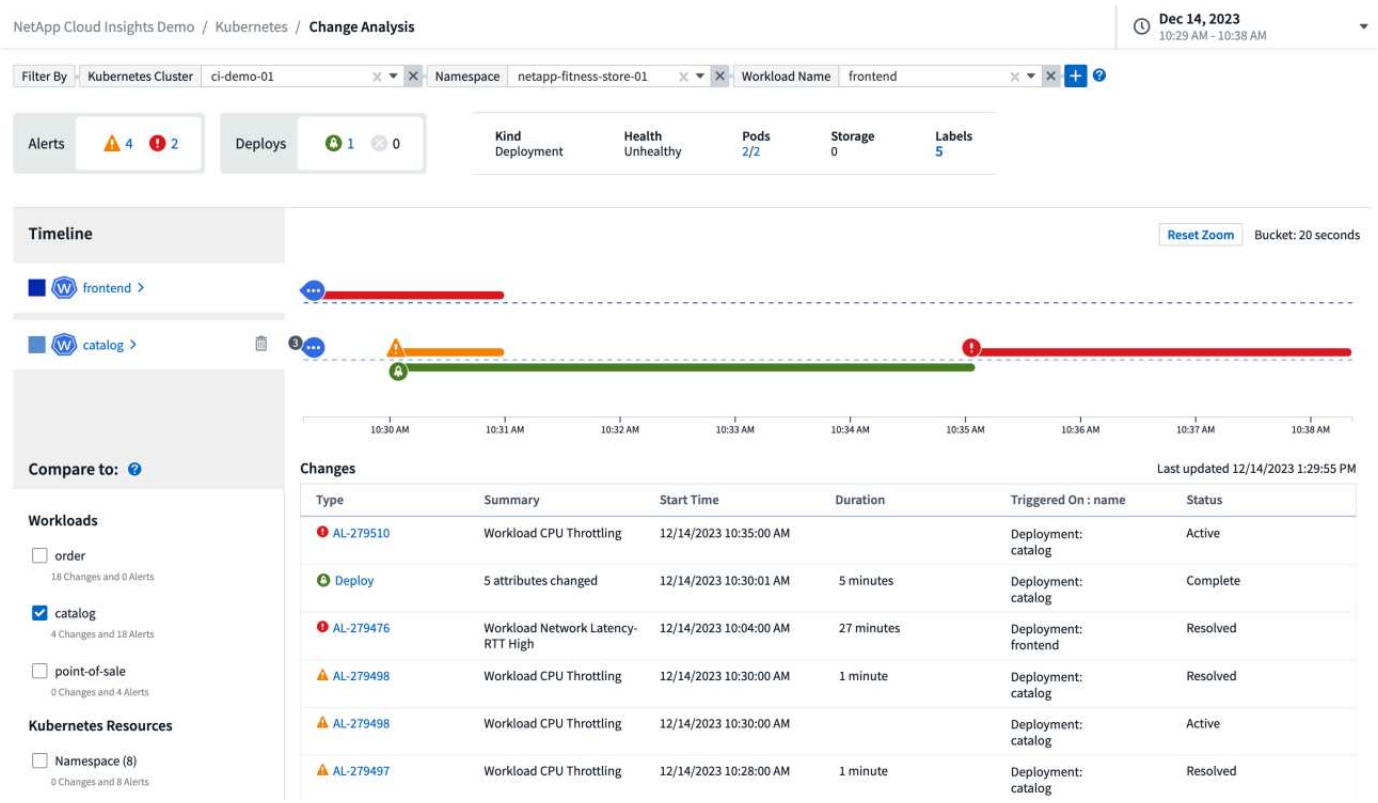
Kubernetes 변경 분석

Kubernetes Change Analytics는 Kubernetes 환경의 최근 변경에 대한 올인원 뷰를 제공합니다. 알림 및 배포 상태를 즉시 확인할 수 있습니다. 변경 분석을 사용하여 모든 배포 및 구성 변경을 추적하고 Kubernetes 서비스, 인프라 및 클러스터의 상태 및 성능과 상호 연관시킬 수 있습니다.

다음 사항에 유의하십시오.

- 멀티 테넌트 환경에서는 잘못 구성된 변경으로 인해 중단이 발생할 수 있습니다. 매우 동적인 환경에서는 Cloud Insights 변경 분석이 모든 변경 사항을 제대로 추적하지 못할 수 있습니다.
- Change Analytics는 워크로드 상태와 구성 변경의 상태를 보고 상호 연관시킬 수 있는 단일 창을 제공합니다. 이는 동적 환경의 문제를 해결하는 데 도움이 될 수 있습니다.

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



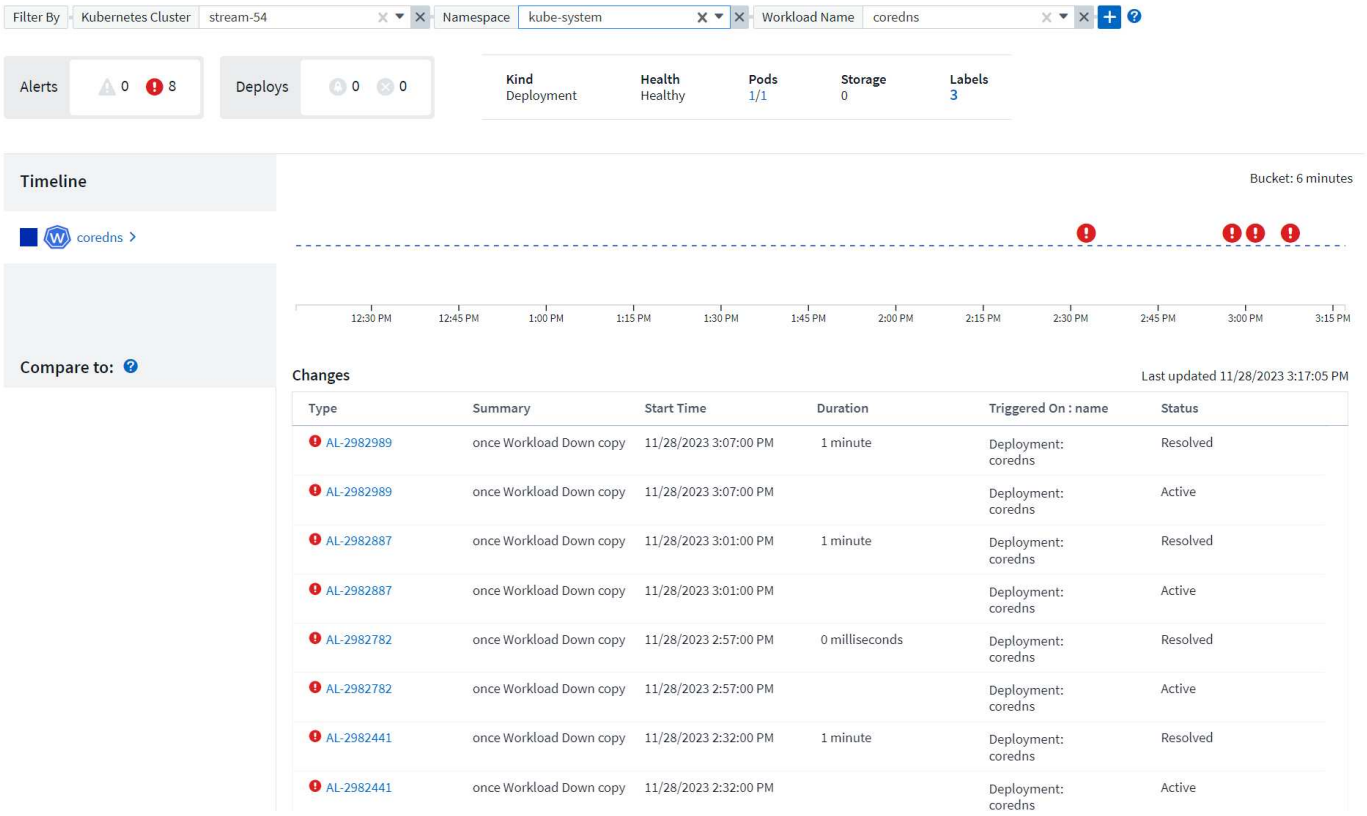
.

현재 선택한 Cloud Insights 시간 범위에 따라 페이지가 자동으로 새로 고쳐집니다. 시간 범위가 작을수록 화면 새로 고침이 더 자주 발생합니다.

필터링

Cloud Insights의 모든 기능과 마찬가지로 변경 목록을 필터링하는 것도 직관적입니다. 페이지 상단에서 Kubernetes 클러스터, 네임스페이스 또는 워크로드의 값을 입력하거나 선택하거나 {+] 버튼을 선택하여 고유한 필터를 추가할 수 있습니다.

특정 클러스터, 네임스페이스 및 워크로드(사용자가 설정한 다른 필터 포함)로 필터링하면 해당 클러스터의 해당 네임스페이스에 있는 해당 워크로드에 대한 배포 타임라인과 경고가 표시됩니다. 그래프를 클릭하고 끌어서 보다 구체적인 시간 범위에 초점을 맞춰 확대합니다.

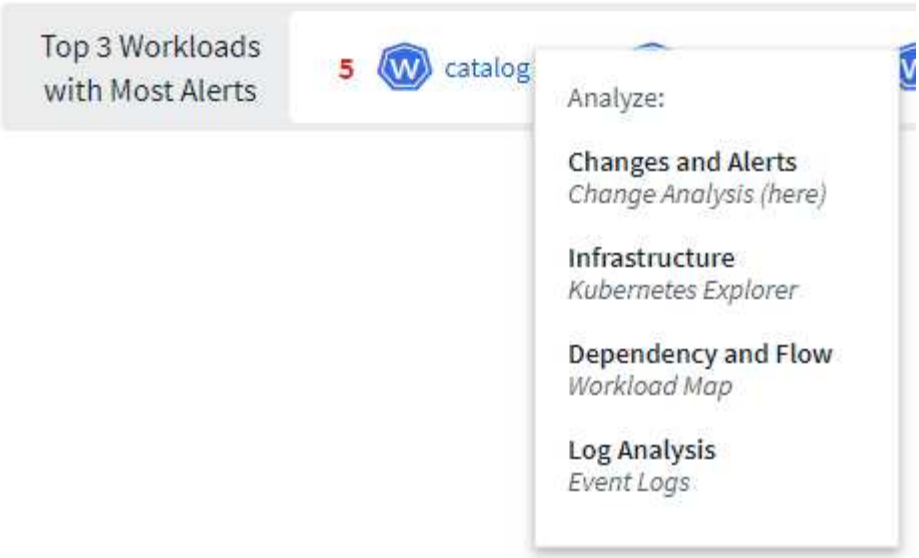


빠른 상태

필터링 영역 아래에는 여러 개의 상위 수준 표시기가 있습니다. 왼쪽에는 경고(Warning) 및 위험(Critical)의 수가 표시됩니다. 이 수에는 _Active_와 _Resolved_alerts 가 포함됩니다. Active_alerts만 보려면 "Status"에 대한 필터를 설정하고 "Active"를 선택합니다.

배포 상태도 여기에 표시됩니다. 기본적으로 *Started*,*Complete* 및 _Failed_deployments의 개수가 표시됩니다. Only_Failed_Deployments를 보려면 "Status"에 필터를 설정하고 "Failed"를 선택합니다.

다음 워크로드로는 경고가 가장 많은 상위 3개 워크로드가 있습니다. 각 워크로드 옆의 빨간색 숫자는 해당 워크로드와 관련된 알림 수를 나타냅니다. 워크로드 링크를 클릭하여 인프라(Kubernetes Explorer), 종속성(워크로드 맵) 또는 로그 분석(이벤트 로그)을 탐색합니다.



세부 정보 패널

목록에서 변경을 선택하면 변경 사항을 자세히 설명하는 패널이 열립니다. 예를 들어 실패한 배포를 선택하면 배포 요약, 시작 및 종료 시간, 기간, 배포가 트리거된 위치, 해당 리소스를 탐색할 수 있는 링크가 표시됩니다. 또한 실패 이유, 관련 변경 사항 및 관련 이벤트도 표시합니다.

Deploy Failed

Summary

Start Time
10/18/2023 2:40:01 PM

End Time
10/18/2023 2:50:02 PM

Duration
10 minutes

Triggered On
ci-demo-01 >
netapp-fitness-store-01 >
billing-accounts >

Triggered On : kind
Deployment

Failure Detail

Reason For Failure
ProgressDeadlineExceeded - ReplicaSet "billing-accounts-6ddc7df546" has timed out progressing.

Message
Failed deploy

Changes (2)

Attribute Name	Previous	New
spec.template.spec.containers[0].image	210811600188.dkr.ecr.us-east-1.amazonaws.com/sm-billing-accounts-apis:1.0.0	210811600188.dkr.ecr.us-east-1.amazonaws.com/sm-billing-accounts-apis:1.0.09
metadata.annotations.deployment.kubernetes.io/revision	2964	2965

All Changes Diff

Associated Events

Event Logs

Close

마찬가지로 경고를 선택하면 경고를 트리거한 모니터와 경고에 대한 시각적 타임라인을 보여 주는 차트를 비롯하여 알림에 대한 세부 정보가 제공됩니다.