



쿠버네티스

Data Infrastructure Insights

NetApp
January 10, 2025

목차

쿠버네티스	1
Kubernetes 클러스터 개요	1
NetApp Kubernetes 모니터링 운영자를 설치 또는 업그레이드하기 전에	2
Kubernetes Monitoring Operator 설치 및 구성	6
Kubernetes 모니터링 운영자 구성 옵션	21
Kubernetes 클러스터 세부 정보 페이지	34
Kubernetes 네트워크 성능 모니터링 및 맵	38
Kubernetes 변경 분석	46

쿠버네티스

Kubernetes 클러스터 개요

Data Infrastructure Insights Kubernetes Explorer는 Kubernetes 클러스터의 전체 상태 및 사용량을 보여주기 위한 강력한 툴이며 조사 영역을 쉽게 드릴다운할 수 있습니다.

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

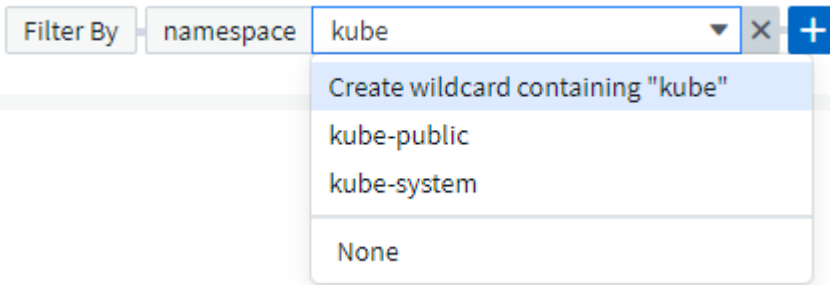
클러스터 목록

클러스터 목록에는 테넌트의 각 클러스터에 대한 다음 정보가 표시됩니다.

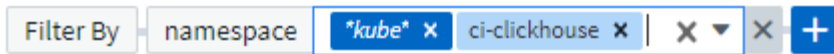
- 클러스터 * 이름 . 클러스터 이름을 클릭하면 해당 클러스터에 대한 가 열립니다" 상세 페이지 *".
- * 채도 * 백분율. 전체 채도란 CPU, 메모리 또는 저장소 채도가 가장 높은 것입니다.
- 클러스터의 * 노드 수 * 입니다. 이 번호를 클릭하면 노드 목록 페이지가 열립니다.
- 클러스터의 Pod 수 이 번호를 클릭하면 창 목록 페이지가 열립니다.
- 클러스터의 * 네임스페이스 * 수입니다. 이 번호를 클릭하면 네임스페이스 목록 페이지가 열립니다.
- 클러스터에 있는 * 워크로드 * 수. 이 번호를 클릭하면 워크로드 목록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

를 설치 또는 업그레이드하기 전에 이 정보를 "[Kubernetes 모니터링 운영자](#)" 읽으십시오.

구성 요소	요구 사항
Kubernetes 버전	Kubernetes v1.20 이상
Kubernetes 배포	AWS EKS(Elastic Kubernetes Service) AKS(Google Kubernetes Service) GKE(Google Kubernetes Engine) RKE(Red Hat OpenShift Rancher Kubernetes Engine) VMware Tanzu
리눅스 OS	Data Infrastructure Insights는 Arm64 아키텍처로 실행되는 노드를 지원하지 않습니다. 네트워크 모니터링: Linux 커널 버전 4.18.0 이상을 실행해야 합니다. Photon OS는 지원되지 않습니다.
라벨	Data Infrastructure Insights는 Kubernetes v1.20 이상 플랫폼에서 다음과 같은 Kubernetes 레이블을 찾는 Kubernetes 노드 선택기를 지정하여 Linux를 실행하는 Kubernetes 노드 모니터링을 지원합니다. Kubernetes.IO/OS=Linux Rancher+Cattle.IO as 오케스트레이션/Kubernetes 플랫폼: cool.IO/OS=Linux
명령	curl 및 kubectl 명령을 사용할 수 있어야 합니다. 최상의 결과를 얻으려면 이러한 명령을 경로에 추가하십시오.
연결성	kubettl CLI는 대상 K8s 클러스터와 통신하도록 구성되고 Data Infrastructure Insights 환경에 인터넷에 연결되어 있습니다. 설치하는 동안 프록시 뒤에 있는 경우 작업자 설치 섹션의 지침을 " 프록시 지원을 구성하는 중입니다 " 따르십시오. 정확한 감사 및 데이터 보고를 위해 NTP(Network Time Protocol) 또는 SNTP(Simple Network Time Protocol)를 사용하여 Agent 시스템의 시간을 동기화합니다.
기타	OpenShift 4.6 이상에서 실행 중인 경우 이 필수 구성 요소가 충족되는지 확인하는 것 외에도 를 따라야 " OpenShift 지침 "합니다.
API 토큰	Operator를 다시 배포하는 경우(예: 업데이트 또는 교체) 새 API 토큰을 생성할 필요가 없습니다. 이전 토큰을 다시 사용할 수 있습니다.

시작하기 전에 유의해야 할 중요 사항

를 프록시사용하거나사용자 지정 리포지토리, 를 사용하거나 를 사용하는 경우 OpenShift다음 섹션을 주의 깊게 읽어 보십시오.

에 대해서도 읽어 권한보세요.

프록시 지원을 구성하는 중입니다

NetApp Kubernetes 모니터링 오퍼레이터를 설치하기 위해 테넌트의 프록시를 사용할 수 있는 위치는 두 곳입니다. 이러한 시스템은 동일하거나 별도의 프록시 시스템일 수 있습니다.

- 설치 코드 스니펫을 실행하는 동안("curl" 사용) 프록시를 사용하여 스니펫이 실행되는 시스템을 Data Infrastructure Insights 환경에 연결해야 합니다
- Data Infrastructure Insights 환경과 통신하는 데 필요한 프록시입니다

둘 중 하나 또는 둘 모두에 프록시를 사용하는 경우 NetApp Kubernetes 운영 모니터를 설치하려면 먼저 프록시가 데이터 인프라 인사이트 환경과 제대로 통신할 수 있도록 구성되었는지 확인해야 합니다. 예를 들어, 운영자를 설치할 서버/VM에서 Data Infrastructure Insights에 액세스하고 Data Infrastructure 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 사용자에게 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_port>
```

Kubernetes 클러스터가 데이터 인프라 인사이트 환경과 통신하는 데 사용되는 프록시의 경우, 이 지침을 모두 읽은 후 NetApp Kubernetes 모니터링 운영자를 설치합니다.

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 Data Infrastructure 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 Data Infrastructure Insights 저장소에 로그인하고 운영자의 모든 이미지 종속성을 가져온 다음 Data Infrastructure Insights 저장소에서 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. 리포지토리에 있는 이러한 이미지에 대한 이미지 태그와 디렉터리 경로가 Data Infrastructure Insights 리포지토리에 있는 이미지와 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 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 이상에서 실행 중인 경우 _operator-config.yaml_에서 AgentConfiguration을 편집하여 _Privileged_setting_을 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 로그를 사용하여 이러한 리소스에 대한 액세스를 운영자에게 수동으로 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


Kubernetes Monitoring Operator 설치 및 구성

Data Infrastructure Insights는 Kubernetes 컬렉션을 위한 * Kubernetes Monitoring Operator * 를 제공합니다. Kubernetes > Collectors > + Kubernetes Collector * 로 이동하여 새 연산자를 구축합니다.

Kubernetes Monitoring Operator를 설치하기 전에

"필수 구성 요소"Kubernetes Monitoring Operator를 설치 또는 업그레이드하기 전에 설명서를 참조하십시오.

Kubernetes 모니터링 운영자 설치



kubernetes
Kubernetes

Deploy NetApp Monitoring Operator

Quickly install and configure a Kubernetes Operator to send cluster information to Cloud Insights.

Select existing API Access Token or create a new one

KEY2024 (...vw6NdM) ▾

+ API Access Token

Production Best Practices ?

Installation Instructions Need Help?

Please review the [pre-requisites](#) for installing the NetApp Kubernetes Monitoring Operator. To update an existing operator installation please follow [these steps](#).

- #### 1 Define Kubernetes cluster name and namespace

Provide the Kubernetes cluster name and specify a namespace for deploying the monitoring components.

Cluster	Namespace
clustername	netapp-monitoring
- #### 2 Download the operator YAML files

Execute the following download command in a *bash* prompt.

Copy Download Command Snippet

⊞
Reveal Download Command Snippet

This snippet includes a unique access key that is valid for 24 hours.

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 **Kubernetes Monitoring Operator** 에이전트를 설치하는 단계:

1. 고유한 클러스터 이름 및 네임스페이스를 입력합니다. 이전 Kubernetes 운영자의 경우 **업그레이드 중** 동일한 클러스터 이름과 네임스페이스를 사용합니다.
2. 이러한 내용을 입력하면 다운로드 명령 스크립트를 클립보드에 복사할 수 있습니다.
3. 스크립트를 `_bash_window`에 붙여 넣고 실행합니다. 오퍼레이터 설치 파일이 다운로드됩니다. 스크립트에는 고유한 키가 있으며 24시간 동안 유효합니다.
4. 사용자 지정 또는 개인 리포지토리가 있는 경우 선택적 이미지 풀 스크립트를 복사하여 `_bash_shell`에 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 대한 자세한 내용을 읽을 수 **"구성 옵션"** 있습니다.
6. 준비가 되면 `kubeck Apply` 스크립트를 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.

프록시를 사용하는 경우 정보를 **프록시를 구성하는 중**입니다 읽습니다.

사용자 지정 리포지토리가 있는 경우 정보를 **사용자 지정/프라이빗 Docker 저장소 사용** 읽습니다.

Kubernetes 모니터링 구성 요소

Data Infrastructure Insights Kubernetes Monitoring은 네 가지 모니터링 구성 요소로 이루어져 있습니다.


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

위의 옵션 구성 요소는 각 Kubernetes Collector에 대해 기본적으로 활성화되어 있습니다. 특정 수집기에 구성 요소가 필요하지 않다고 결정한 경우 * Kubernetes > Collectors * 로 이동한 후 화면 오른쪽의 수집기의 "세 가지 점" 메뉴에서 _Modify Deployment_를 선택하여 비활성화할 수 있습니다.

NetApp / Observability / Collectors

Cluster Name ↑	Status	Operator Version	Network Performance and Map	Change Analysis
au-pod	⚠ Outdated	1.1540.0	1.347.0	1.162.0
jks-troublemaker	Latest	1.1579.0	N/A	1.201.0
oom-test	⚠ Outdated	1.1555.0	N/A	1.101.0

화면에는 각 구성 요소의 현재 상태가 표시되며 필요한 경우 해당 Collector에 대한 구성 요소를 비활성화하거나 활성화할 수 있습니다.

 **kubernetes**
Kubernetes

Modify Deployment

Cluster Information

Kubernetes Cluster
ci-demo-01

Network Performance and Map
Enabled - Online

Event Logs
Enabled - Online

Change Analysis
Enabled - Online

Deployment Options

[Need Help?](#)

- Network Performance and Map
- Event Logs
- Change Analysis

Cancel

Complete Modification

최신 Kubernetes Monitoring Operator로 업그레이드하십시오

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 오퍼레이터
 - **최신 컨테이너 이미지를 가져오는 중입니다** 사용자 지정 리포지토리를 사용하고 있는지 확인합니다.

AgentConfiguration이 없는 경우:

- 데이터 인프라 인사이트 에서 인식된 클러스터 이름을 기록합니다(네임스페이스가 기본 NetApp-monitoring이 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-kubernetes-monitoring-operator, 설치 제거>> 기존 오퍼레이터
* <<installing-the-kubernetes-monitoring-operator, 설치합니다>> 최신 연산자.

- 동일한 클러스터 이름을 사용합니다.
- 최신 운영자 YAML 파일을 다운로드한 후 배포하기 전에 agent_backup.YAML에서 발견된 모든 사용자 정의를 다운로드한 operator-config.YAML에 이식하십시오.
- **최신 컨테이너 이미지를 가져오는 중입니다** 사용자 지정 리포지토리를 사용하고 있는지 확인합니다.

Kubernetes 모니터링 오퍼레이터 중지 및 시작

Kubernetes 모니터링 운영자를 중지하려면 다음을 따르십시오.

```
kubectl -n netapp-monitoring scale deploy monitoring-operator --replicas=0
```

Kubernetes 모니터링 운영자를 시작하려면 다음을 따르십시오.

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

제거 중

Kubernetes Monitoring Operator를 제거합니다

Kubernetes Monitoring Operator의 기본 네임스페이스는 "netapp-monitoring"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 Operator는 다른 인스턴스와 충돌을 피하기 위해 고유한 kube-state-metrics를 설치합니다.

Kube-State-Metrics에 대한 자세한 내용은 ["이 페이지"](#)참조하십시오.

오퍼레이터 구성/사용자 지정

이 섹션에서는 운영자 구성을 사용자 지정하거나, 프록시를 사용하거나, 사용자 지정 또는 개인 Docker 리포지토리를 사용하거나, OpenShift를 사용하는 방법에 대한 정보를 제공합니다.

설정 옵션

가장 일반적으로 수정된 설정은 `_AgentConfiguration_custom` 리소스에서 구성할 수 있습니다. 운영자를 배포하기 전에 `_operator-config.YAML_file`을 편집하여 이 리소스를 편집할 수 있습니다. 이 파일에는 주석 처리된 설정 예가 포함되어 있습니다. 최신 버전의 연산자는 의 목록을 ["사용 가능한 설정"](#)참조하십시오.

다음 명령을 사용하여 연산자를 배포한 후에 이 리소스를 편집할 수도 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

Kubernetes Monitoring Operator를 설치하기 위해 테넌트에서 프록시를 사용할 수 있는 두 위치가 있습니다. 이러한 시스템은 동일하거나 별도의 프록시 시스템일 수 있습니다.

- 설치 코드 스니펫을 실행하는 동안("curl" 사용) 프록시를 사용하여 스니펫이 실행되는 시스템을 Data Infrastructure Insights 환경에 연결해야 합니다
- Data Infrastructure Insights 환경과 통신하는 데 필요한 프록시입니다

이 중 하나 또는 둘 모두에 프록시를 사용하는 경우 Kubernetes 운영 모니터를 설치하려면 먼저 프록시가 Data Infrastructure Insights 환경과 통신할 수 있도록 구성되었는지 확인해야 합니다. 프록시가 있고 운영자를 설치하려는 서버/VM에서 Data Infrastructure Insights에 액세스할 수 있는 경우 프록시가 제대로 구성된 것일 수 있습니다.

Kubernetes 운영 모니터를 설치하는 데 사용되는 프록시의 경우 Operator를 설치하기 전에 `_http_proxy/https_proxy_environment` 변수를 설정합니다. 일부 프록시 환경에서는 `_no_proxy` 환경_변수를 설정해야 할 수도 있습니다.

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

1. 현재 사용자에게 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_port>
```

Kubernetes 클러스터가 Data Infrastructure Insights 환경과 통신하는 데 사용되는 프록시의 경우 이 모든 지침을 읽은 후 Kubernetes Monitoring Operator를 설치합니다.

Kubernetes 모니터링 연산자를 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 지정 또는 프라이빗 **Docker** 저장소 사용

기본적으로 Kubernetes Monitoring Operator는 Data Infrastructure Insights 저장소에서 컨테이너 이미지를 가져옵니다. Kubernetes 클러스터를 모니터링 대상으로 사용하고 있고 해당 클러스터가 사용자 지정 또는 프라이빗 Docker 저장소 또는 컨테이너 레지스트리에서 컨테이너 이미지만을 가져오도록 구성된 경우, Kubernetes Monitoring Operator가 필요한 컨테이너에 대한 액세스를 구성해야 합니다.

NetApp 모니터링 오퍼레이터 설치 타일에서 "이미지 풀 스니펫"을 실행합니다. 이 명령은 Data Infrastructure Insights 저장소에 로그인하고 운영자의 모든 이미지 종속성을 가져온 다음 Data Infrastructure Insights 저장소에서 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 - 텔레그래프
- distroless-root-user.(거리 없는 루트 사용자)

이벤트 로그

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

네트워크 성능 및 맵

- CI-NET-관찰자

회사 정책에 따라 운영 Docker 이미지를 프라이빗/로컬/엔터프라이즈 Docker 저장소로 밀어 넣습니다. 리포지토리에 있는 이러한 이미지에 대한 이미지 태그와 디렉터리 경로가 Data Infrastructure Insights 리포지토리에 있는 이미지와 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>/ci-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:
  xref:{relative_path}task_config_telegraf_agent_k8s.html#using-a-custom-or-
  private-docker-repository
  dockerRepo: your.docker.repo/long/path/to/test
  # Optional: A docker image pull secret that maybe needed for your
  private docker registry
  dockerImagePullSecret: docker-secret-name
```

OpenShift 지침

OpenShift 4.6 이상에서 실행 중인 경우 _operator-config.yaml_에서 AgentConfiguration을 편집하여

`_Privileged_setting`을 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.

톨러러스와 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. 운영자는 잘 알려진 일부 * 얼룩을 견딜 수 있도록 구성되었습니다. 노드에 사용자 지정 테인트를 구성하여 모든 노드에서 Pod가 실행되지 않도록 설정한 경우, 해당 테인트에 대한 * 톨러레이션 * 을 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 "오염과 내약입니다"자세히 알아보십시오.

로 돌아갑니다 "[* NetApp Kubernetes 모니터링 오퍼레이터 설치 * 페이지](#)"

비밀에 대한 참고 사항

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` 을 수정하여 변경 관리 섹션의 주석을 해제하고 변경 관리 섹션 아래에 `_kindsTolIgnoreFromWatch: "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 모니터링 운영자 이미지 서명 확인

운영자의 이미지 및 배포한 모든 관련 이미지는 NetApp에 의해 서명됩니다. Cosign 도구를 사용하여 설치하기 전에 이미지를 수동으로 확인하거나 Kubernetes 승인 컨트롤러를 구성할 수 있습니다. 자세한 내용은 ["Kubernetes 문서" 참조하십시오.](#)

이미지 서명을 확인하는 데 사용되는 공개 키는 `_선택 사항: 운영자 이미지를 개인 저장소에 업로드하십시오.` > 이미지 서명 공개 키 `_`의 모니터링 운영자 설치 파일에서 사용할 수 있습니다

이미지 서명을 수동으로 확인하려면 다음 단계를 수행하십시오.

1. Image Pull Snippet(이미지 가져오기 스크립트)을 복사하여 실행합니다
2. 메시지가 나타나면 리포지토리 암호를 복사하여 입력합니다
3. 이미지 서명 공개 키(예: `dii-image-signing.pub`)를 저장합니다.
4. 코사인을 사용하여 영상을 확인합니다. 다음 코사인 사용 예를 참조하십시오

```
$ 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 발생할 경우 시도해 볼 수 있는 몇 가지 사항은 다음과 같습니다.

<p>문제:</p>	<p>다음을 시도해 보십시오.</p>
<p>Kubernetes 영구 볼륨과 해당 백엔드 스토리지 장치 간의 하이퍼링크/연결이 표시되지 않습니다. 내 Kubernetes 영구 볼륨은 스토리지 서버의 호스트 이름을 사용하여 구성됩니다.</p>	<p>기존 Telegraf 에이전트를 제거한 다음 최신 Telegraf 에이전트를 다시 설치하는 단계를 따릅니다. Telegraf 버전 2.0 이상을 사용 중이고 Kubernetes 클러스터 스토리지를 Data Infrastructure Insights에서 능동적으로 모니터링해야 합니다.</p>
<p>E0901 15:21:39.962145 1 리플렉터.go:178]k8s.io/kube-state-metrics/internal/store/builder.go:352: * v1.목록에 실패했습니다. MutatingWebhookConfiguration: 서버에서 요청된 리소스 E0901 15:21:43.168352.kls.tu2o.tu2352: revm.u2352.u2352.u2o.testimeu2352.u2n.u2352.u2o.u2o.u2352.testime-ve-v</p>	<p>이러한 메시지는 Kubernetes 버전이 1.20 미만인 경우 kube-state-metrics 버전 2.0.0 이상을 실행하는 경우에 발생할 수 있습니다. 쿠버네티스 버전을 얻으려면: <code>_kubctl version_kubbe-state-metrics 버전: _kubectl deploy/kube-state-metrics-o jsonpath='{..image}'</code> 이러한 메시지가 발생하지 않도록 사용자는 kube-state-metrics 구축을 수정하여 다음 Lallase를 비활성화할 수 있습니다. <code>_muthookconfigurations_webhookconfigurrett al_configuredconfig_webvalidateCLI_webvalidateus etausetauseuseuseuseuseuse 리소스 = certificationesigningrequests, configmap, crontobs, demonset, 배포, 끝점, 수평 포드자동크기, 링스, 작업, reflodritranges, namespaces, networkpolicies, nistentvolumes, persistentpersistent volumes, podin예산, replicatingfasts, repliceters, replicatingreallets, replicets, replicets, repliceties, replicenets, replicatinggreenets, replicets, replicenets, replicets, service.networksets, service.sets, service.sets, vistenets, visteneties, replicaturies.networksets, service.sets, reseties.networksets, replicaturies, inations, replicaticaturies, replicaturies, replicaturies, inations validingwebhookconfigurations, volumeAttachments"</code></p>
<p>Telegraf의 오류 메시지는 다음과 유사하지만 Telegraf가 시작되고 실행됩니다. Oct 11 14:23:41 IP-172-31-39-47 시스템[1]: 플러그인 기반 서버 에이전트를 시작하여 메트릭을 영향력 있는 xDB에 보고합니다. 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.\n"fc="gosnowflake.(* defaultLogger).Errorf" file="log.go:120" 10월 11:23:41 IP-1118-117-117-117-118-117-117" 오류: 무시되었습니다. open /etc/Telegraf/.cache/snowflake/OCSP_response_cache.json: no that file or directory \n" fc="gosnowflake.(* defaultLogger).Errorf" file="log.Go:120" Oct 11:41 IP-31-39-11:21:114:118-427:114:117-118-114:118-427] 텔레그래프 1.19.3 시작</p>	<p>이는 알려진 문제입니다. "이 GitHub 기사를 참조하십시오" 자세한 내용은 을 참조하십시오. Telegraf가 실행 중인 경우 사용자는 이러한 오류 메시지를 무시할 수 있습니다.</p>

문제:	다음을 시도해 보십시오.
Kubernetes에서 Telegraf 포드가 "mountstats 정보 처리 중 오류: mountstats 파일을 열지 못했습니다. /hostfs/proc/1/mountstats, 오류: open/hostfs/proc/1/mountstats: 사용 권한이 거부되었습니다." 오류를 보고합니다.	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 "OpenShift 지침" 참조하십시오.
Kubernetes에서 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/cups/pi/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 ReplicaSet...kubtl을 편집하여 RS Telegraf-RS...를 편집하고 사양에 적절한 내약성을 추가합니다. 그런 다음 ReplicaSet 포드를 다시 시작합니다.
PSP/PSA 환경이 있습니다. 이 문제가 모니터링 오퍼레이터에게 영향을 미칩니까?	Kubernetes 클러스터가 PSP(Pod Security Policy) 또는 PSA(Pod Security Admission)를 적용한 상태에서 실행되는 경우 최신 Kubernetes Monitoring Operator로 업그레이드해야 합니다. 다음 단계에 따라 PSP/PSA를 지원하는 현재 오퍼레이터로 업그레이드합니다. 1. 설치 제거 이전 모니터링 운영자: kubectl delete agent-monitoring-NetApp NetApp-n NetApp-monitoring kubectl delete ns NetApp-monitoring kubectl delete clusterrole agent-manager-proxy-role-metrics-reader-role2 클러스터 바인딩 role2 설치합니다 모니터링 운영자의 최신 버전입니다.
오퍼레이터를 배포하는 데 문제가 발생했고 PSP/PSA를 사용하고 있습니다.	1. kubectl -n <name-space> edit agent 2 명령을 사용하여 에이전트를 편집합니다. '보안 - 정책 - 사용'을 '거짓'으로 표시합니다. 이렇게 하면 Pod 보안 정책 및 Pod 보안 입장이 비활성화되고 운영자가 배포할 수 있습니다. 다음 명령을 사용하여 확인합니다. kubctl get psp(Pod 보안 정책이 제거되었음을 표시해야 함) kubctl get all-n <namespace>
grep -i psp(아무 것도 찾을 수 없다는 것을 표시해야 함)	"ImagePullBackoff" 오류가 표시됩니다
이러한 오류는 사용자 지정 또는 전용 Docker 리포지토리가 있고 Kubernetes Monitoring Operator가 해당 리포지토리를 제대로 인식하도록 아직 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.

<p>문제:</p> <p>다음 명령의 출력을 캡처하거나 기록해 두고 기술 지원 팀에 문의하십시오.</p> <pre> kubect1 -n netapp-monitoring get all kubect1 -n netapp-monitoring describe all kubect1 -n netapp-monitoring logs <monitoring-operator-pod> --all -containers=true kubect1 -n netapp-monitoring logs <telegraf-pod> --all -containers=true </pre>	<p>다음을 시도해 보십시오.</p> <p>Operator 네임스페이스의 NET-observer(워크로드 맵) 포드는 CrashLoopBackOff에 있습니다</p>
<p>이러한 포드는 네트워크 관찰 가능성을 위한 워크로드 맵 데이터 수집기에 해당합니다. 다음을 시도해 보십시오. • 최소 커널 버전을 확인하려면 Pod 중 하나의 로그를 확인하십시오. 예:--{"CI-tenant-id":"your-tenant-id", "collector-cluster":"your-k8s-cluster-name", "environment":"prod", "level":"error", "msg":"가 검증에 실패했습니다. 이유: 커널 버전 3.10.0이 최소 커널 버전 4.18.0", "시간":"2022-11-09T08:23:08Z"}----- • Net-observer Pod는 Linux 커널 버전이 4.18.0 이상이어야 합니다. "uname -r" 명령을 사용하여 커널 버전을 확인하고 해당 버전이 4.18.0 이상인지 확인합니다</p>	<p>Pod는 운영자 네임스페이스(기본값: NetApp-모니터링)에서 실행되지만, 워크로드 맵 또는 Kubernetes 메트릭에 대한 UI에는 데이터가 표시되지 않습니다</p>
<p>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.</p>	<p>Operator 네임스페이스의 일부 NET-observer Pod가 Pending 상태입니다</p>
<p>Net-observer는 DemonSet로, k8s 클러스터의 각 노드에서 포드를 실행합니다. • Pending 상태인 POD를 확인하고 CPU 또는 메모리에 대한 리소스 문제가 발생하는지 확인합니다. 노드에서 필요한 메모리 및 CPU를 사용할 수 있는지 확인합니다.</p>	<p>Kubernetes 모니터링 운영자를 설치한 직후 [inputs.prometheus] 오류: 플러그인에서 http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics:get\http://kube-state-metrics.<namespace>.svc.cluster.local:8080/metrics:dial tcp:lookube-state-metrics.<namespace>.svc.no host cluster.local:no.local:no host.local:no.local:no</p>
<p>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.</p>	<p>클러스터에 존재하는 Kubernetes CronJobs에 대해 어떤 메트릭도 수집되지 않습니다.</p>

문제:	다음을 시도해 보십시오.
Kubernetes 버전을 확인합니다(예: <code>kubectl version</code>). v1.20.x 이하일 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. 따라서 kube-state-metrics는 cronjob 리소스를 찾을 수 없습니다.	운영자 설치 후, Telegraf-ds Pod는 CrashLoopBackOff로 진입하고 POD 로그는 "su:Authentication failure"를 나타낸다.
_AgentConfiguration_에서 Telegraf 섹션을 편집하고 _dockerMetricCollectionEnabled_를 false로 설정합니다. 자세한 내용은 조작자를 "구성 옵션" 참조하십시오. ... 사양 : ... 텔레그라프 : ... -name:docker run-mode: -DemonSet 대체:-key:docker_unix_sock_placeholder 값: unix://run/docker.sock... ..	내 Telegraf 로그에 다음과 유사한 오류 메시지가 반복 표시됩니다: E! [agent] 출력에 쓰는 중 오류가 발생했습니다. http: POST "https://<tenant_url>/REST/v1/lake/injest/influxdb": 컨텍스트 기한 초과(클라이언트. 헤더 대기 중 시간 초과)
_AgentConfiguration_에서 Telegraf 섹션을 편집하고 increase_outputTimeout_을 10초로 편집합니다. 자세한 내용은 조작자를 "구성 옵션" 참조하십시오.	일부 이벤트 로그에 대한 _divedobject_data_가 없습니다.
위 섹션의 단계를 수행했는지 "권한"확인하십시오.	두 개의 모니터링 운영자 Pod가 실행 중인 것을 볼 수 있는데, 하나는 netapp-ci-monitoring-operator- <pod>이고 다른 하나는 monitoring-operator- <pod>입니다.
2023년 10월 12일부터 Data Infrastructure Insights는 사용자에게 더 나은 서비스를 제공하기 위해 운영자를 재고했습니다. 변경 사항을 완전히 채택하려면 반드시 기존 연산자를 제거합니다 AND를 준수해야 합니다.새 장치를 장착하십시오	저의 Kubernetes 이벤트가 예기치 않게 Data Infrastructure Insights에 대한 보고를 중단했습니다.
이벤트 내보내기 포드의 이름을 검색합니다. <pre>`kubectl -n netapp-monitoring get pods`</pre>	grep event-exporter

문제:	다음을 시도해 보십시오.
<pre>awk '{print \$1}'</pre>	<pre>sed 's/event-exporter./event-exporter/' "netapp-ci-event-exporter" 또는 "event-exporter"여야 합니다. 그런 다음 모니터링 에이전트를 편집하고 kubectl -n netapp-monitoring edit agent 이전 단계에서 찾은 적절한 이벤트 내보내기 포드 이름을 반영하도록 log_file 의 값을 설정합니다. 보다 구체적으로 log_file을 "/var/log/containers/netapp-ci-event- exporter.log" 또는 "/var/log/containers/event-exporter .log"로 설정해야 합니다.</pre> <p>....</p> <p>fluent-bit:</p> <p>...</p> <p>- name: event-exporter-ci</p> <p>substitutions:</p> <p>- key: LOG_FILE</p> <p>values:</p> <p>- /var/log/containers/netapp-ci-event-exporter.log</p> <p>...</p> <p>....</p> <p>또는 상담원과 다시 설치합니다 함께 할 설치 제거 수도 있습니다.</p>
리소스 부족으로 인해 Kubernetes Monitoring Operator에 의해 구축된 Pod가 충돌하는 것을 볼 수 있습니다.	필요한 경우 Kubernetes 모니터링 운영자를 "구성 옵션" 참조하여 CPU 및/또는 메모리 한도를 늘리십시오.
이미지가 없거나 잘못된 구성으로 인해 NetApp-ci-kube-state 메트릭 Pod가 시작되지 않거나 준비되지 않았습니다. 이제 StatefulSet이 중단되고 구성 변경 사항이 NetApp-ci-kube-state-metrics Pod에 적용되지 않습니다.	StatefulSet 이 "고장" 상태입니다. 구성 문제를 해결한 후에는 NetApp-ci-kube-state-metrics Pod를 바운스하게 만들 수 있습니다.
Kubernetes Operator 업그레이드를 실행하고 ErrImagePull(이미지를 가져오지 못함)을 throw하는 후 NetApp-ci-kube-state-metrics Pod가 시작되지 않습니다.	포드 재설정을 수동으로 시도하십시오.
"이벤트가 maxEventAgeSeconds보다 오래된 것으로 폐기됨" 메시지가 Log Analysis에서 내 Kubernetes 클러스터에 대해 관찰되고 있습니다.	Operator_agentconfiguration 을 수정하고 <i>event-exporter-maxEventAgeSeconds</i> (예: 60s), <i>event-exporter-kubeQPS</i> (예: 100) 및 <i>event-exporter-kubeBurst</i> (예: 500)를 증가시킵니다. 이러한 구성 옵션에 대한 자세한 내용은 "구성 옵션" 페이지를 참조하십시오.
Telegraf는 잠금 가능한 메모리가 부족하다는 경고 또는 충돌에 대해 경고합니다.	기본 운영 체제/노드에서 Telegraf의 잠금 가능한 메모리 제한을 늘리십시오. 제한을 늘리는 것이 옵션이 아닌 경우 NKMO agentconfiguration과 <code>set_unprotected_to_true</code> 를 수정합니다. 그러면 Telegraf가 잠긴 메모리 페이지를 예약하지 않도록 지시합니다. 암호 해독된 암호가 디스크로 교체될 수 있기 때문에 보안 위험이 발생할 수 있지만 잠긴 메모리를 예약할 수 없는 환경에서 실행할 수 있습니다. <code>_unprotected_configuration</code> 옵션에 대한 자세한 내용은 "구성 옵션" 페이지를 참조하십시오.

<p>문제:</p>	<p>다음은 시도해 보십시오.</p>
<p>다음과 유사한 Telegraf의 경고 메시지가 표시됩니다. <code>_W!</code> <code>[input.diskio] "vdc"에 대한 디스크 이름을 수집할 수 없습니다. /dev/vdc 읽기 오류: 해당 파일 또는 디렉토리 _</code></p>	<p>Kubernetes 모니터링 운영자의 경우 이러한 경고 메시지는 양호하며 무시해도 됩니다. 또는 <code>AgentConfiguration</code>에서 Telegraf 섹션을 편집하고 <code>_runDsPrivileged_</code>를 <code>true</code>로 설정합니다. 자세한 내용은 "오퍼레이터 구성 옵션"참조하십시오.</p>
<p>내 fluent-bit POD가 다음 오류로 인해 실패하고 있습니다. <code>[2024/10/16 14:16:23] [ERROR] [/src/fluent-bit/plugins/in_tail/fs_fs_inotify.c:360 errno=24] 열려 있는 파일이 너무 많습니다. [ERROR</code></p>	<p>클러스터에서 <code>_fsnotify_settings</code>를 변경해 보십시오.</p> <pre> 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> </pre> <p>Fluent-bit을 다시 시작합니다.</p> <p>참고: 이러한 설정을 노드 재시작 시에도 계속 유지하려면 <code>_/etc/sysctl.conf_</code>에 다음 줄을 입력해야 합니다</p> <pre> fs.inotify.max_user_instances=<something larger than current setting> fs.inotify.max_user_watches=<something larger than current setting> </pre>

추가 정보는 페이지 또는 에서 찾을 ["지원" Data Collector 지원 매트릭스](#) 수 있습니다.

Kubernetes 모니터링 운영자 구성 옵션

이 ["Kubernetes 모니터링 운영자"](#) 구성은 사용자 정의할 수 있습니다.

아래 표에는 `_AgentConfiguration_` 파일에 사용할 수 있는 옵션이 나와 있습니다.

구성 요소	옵션을 선택합니다	설명
에이전트		운영자가 설치할 수 있는 모든 구성품에 공통으로 적용되는 구성 옵션. 이러한 옵션은 "글로벌" 옵션으로 간주할 수 있습니다.
	dockerRepo	DockerRepo는 Data Infrastructure Insights Docker 저장소와 비교하여 고객의 전용 Docker Repos에서 이미지를 가져오도록 재정의합니다. 기본값은 Data Infrastructure Insights Docker 저장소입니다
	dockerImagePullSecret	선택 사항: 고객의 비공개 리포의 비밀
	클러스터 이름	모든 고객 클러스터에서 클러스터를 고유하게 식별하는 자유 텍스트 필드입니다. Data Infrastructure Insights 테넌트 전체에서 고유해야 합니다. 기본값은 고객이 "클러스터 이름" 필드의 UI에 입력하는 것입니다
	프록시 형식: proxy:server:port:username:password:nProxy:isTelegyxyEnabled:isAuProxyEnabled:isFluentbitProxyEnabled:isCollectionorProxyEnabled:isCollectorProxyEnabled:	프록시를 설정하는 선택 사항입니다. 일반적으로 고객의 기업 대리인입니다.
텔레그래프		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
	runPrivileged(권한이	권한 부여 모드에서 Telegraf DemonSet's_Telegraf-mountstats-poller_container를 실행합니다. Kubernetes 노드에서 SELinux가 활성화된 경우 이 값을 true로 설정합니다.
	runDsPrivileged입니다	권한 있는 모드에서 Telegraf DemonSet의 Telegraf 컨테이너를 실행하려면 runDsPrivileged를 true로 설정합니다.
	BatchSize(부atchSize)	을 참조하십시오 "Telegraf 구성 문서"
	백혈구 한계	을 참조하십시오 "Telegraf 구성 문서"

구성 요소	옵션을 선택합니다	설명
	RoundInterval(라운드 간격)	을 참조하십시오 "Telegraf 구성 문서"
	콜렉션 지터	을 참조하십시오 "Telegraf 구성 문서"
	정밀도	을 참조하십시오 "Telegraf 구성 문서"
	FlushInterval(플러시 간격)	을 참조하십시오 "Telegraf 구성 문서"
	플러시 지터	을 참조하십시오 "Telegraf 구성 문서"
	출력 제한 시간	을 참조하십시오 "Telegraf 구성 문서"
	dsTolerations 를 선택합니다	텔레그라프 - DS 추가 허용.
	RsTolerations를 선택합니다	Telegraf-RS 추가 허용.
	SkipProcessorsAfterAggregators 를 참조하십시오	을 참조하십시오 "Telegraf 구성 문서"
	보호되지 않음	자세한 "알려진 Telegraf 문제입니다" 내용을 확인하십시오. setting_unprotected_는 Kubernetes 모니터링 운영자에게 플래그와 함께 Telegraf를 실행하도록 지시합니다. --unprotected
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 심표로 구분된 목록입니다. 예: cronjobs, demonsets, 배포, Ingresses, 작업, 네임스페이스, 노드, 영구 볼륨, 영구 볼륨, pods, replicaset, resourcequotas, 서비스, statefulsets
	공차	Kudbe-state-metrics 추가 공약입니다.
	라벨	kube-state-metrics가 캡처해야 하는 심표로 구분된 리소스 목록 + 예: crontobs=[], demonsets=[], 배포=[], ingports=[], 작업=[], 네임스페이스=[], 노드=[], persistentvolumeclaims=[], persistentvolumes=[], statestatestatestatesteasets=], stequets=*] *], statestequets=
로그		운영자의 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.

구성 요소	옵션을 선택합니다	설명
	이벤트-수출자-내약성	이벤트-수출자 추가 허용.
	event-exporter-maxEventAgeSeconds입니다	이벤트 내보내기 최대 이벤트 기간. 을 참조하십시오 https://github.com/jkroepke/resmoio-kubernetes-event-exporter
워크로드 맵		작업량 맵 수집 및 작업자의 설치를 사용자 지정할 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
	FailureDeclarationInterval Mins 를 참조하십시오	실패한 워크로드 배포가 실패로 표시되는 간격(분)입니다
	deployAggrIntervalSeconds입니다	작업 부하 배포 진행 중 이벤트가 전송되는 빈도입니다
	비작업 로드 AggrIntervalSeconds입니다	비워크로드 구축이 결합되고 전송되는 빈도입니다
	TERmsToRedact 를 참조하십시오	env 이름 및 데이터 맵에 사용되는 정규식 집합이며, 값이 "pwd", "password", "token", "apikey", "api-key", "jwt" 등의 용어가 교정됩니다.
	AdditionalKindsToWatch 를 참조하십시오	수집기에서 감시하는 기본 종류 집합에서 볼 수 있는 추가 종류의 심표로 구분된 목록
	KindsToIgnoreFromWatch 를 참조하십시오	수집기에서 감시하는 기본 종류의 집합에서 감시하는 것을 무시할 수 있는 심표로 구분된 종류의 목록입니다
	LogRecordAggrIntervalSeconds입니다	수집기에서 CI로 로그 레코드를 보내는 빈도입니다

구성 요소	옵션을 선택합니다	설명
	시계의 내약성	change-observer-watch-DS 추가 허용 오차. 축약된 단일선 형식만 해당. 예: '{key:taint1,operator:exists,effect:NoSchedule}, {key:taint2,operator:exists,effect:NoExecute}'

AgentConfiguration 파일 예

다음은 샘플_AgentConfiguration_파일입니다.

```

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

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

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

```

```

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

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

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

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

```

```

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

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

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

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

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

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

    # # Run the telegraf DaemonSet's telegraf-mountstats-poller container

```

in privileged mode. Set runPrivileged to true if SELinux is enabled on your Kubernetes nodes.

```
# runPrivileged: '{{
.Values.telegraf_installer.kubernetes.privileged_mode }}'
```

```
# # Set runDsPrivileged to true to run the telegraf DaemonSet's
telegraf container in privileged mode
```

```
# runDsPrivileged: '{{
.Values.telegraf_installer.kubernetes.ds.privileged_mode }}'
```

```
# # Collect container Block IO metrics.
```

```
# dsBlockIOEnabled: 'true'
```

```
# # Collect NFS IO metrics.
```

```
# dsNfsIOEnabled: 'true'
```

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

```
# managedK8sSystemMetricCollectionEnabled: 'false'
```

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

```
# podVolumeMetricCollectionEnabled: 'false'
```

```
# # Declare Rancher cluster as managed. Set this to true if your
Rancher cluster is managed as opposed to on-premise.
```

```
# isManagedRancher: 'false'
```

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

```
# rsHostEtcdCrt: ''
```

```
# rsHostEtcdKey: ''
```

```
# kube-state-metrics:
```

```
# # kube-state-metrics CPU/Mem limits and requests.
```

```
# cpuLimit: '500m'
```

```
# memLimit: '1Gi'
```

```
# cpuRequest: '100m'
```

```
# memRequest: '500Mi'
```

```
# # Comma-separated list of resources to enable.
```

```
# # See resources in https://github.com/kubernetes/kube-state-metrics/blob/main/docs/cli-arguments.md
```

```
# resources:
```

```
'cronjobs,daemonsets,deployments,ingresses,jobs,namespaces,nodes,persisten
```

```
tvolumeclaims,persistentvolumes,pods,replicasets,resourcequotas,services,statfulsets'
```

```
# # Comma-separated list of metrics to enable.
```

```
# # See metric-allowlist in https://github.com/kubernetes/kube-state-metrics/blob/main/docs/cli-arguments.md
```

```
# metrics:
```

```
'kube_cronjob_created,kube_cronjob_status_active,kube_cronjob_labels,kube_daemonset_created,kube_daemonset_status_current_number_scheduled,kube_daemonset_status_desired_number_scheduled,kube_daemonset_status_number_available,kube_daemonset_status_number_misscheduled,kube_daemonset_status_number_ready,kube_daemonset_status_number_unavailable,kube_daemonset_status_observed_generation,kube_daemonset_status_updated_number_scheduled,kube_daemonset_metadata_generation,kube_daemonset_labels,kube_deployment_status_replicas,kube_deployment_status_replicas_available,kube_deployment_status_replicas_unavailable,kube_deployment_status_replicas_updated,kube_deployment_status_observed_generation,kube_deployment_spec_replicas,kube_deployment_spec_paused,kube_deployment_spec_strategy_rollingupdate_max_unavailable,kube_deployment_spec_strategy_rollingupdate_max_surge,kube_deployment_metadata_generation,kube_deployment_labels,kube_deployment_created,kube_job_created,kube_job_owner,kube_job_status_active,kube_job_status_succeeded,kube_job_status_failed,kube_job_labels,kube_job_status_start_time,kube_job_status_completion_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_persistentvolume_info,kube_persistentvolume_claim_ref,kube_persistentvolumeclaim_access_mode,kube_persistentvolumeclaim_info,kube_persistentvolumeclaim_labels,kube_persistentvolumeclaim_resource_requests_storage_bytes,kube_persistentvolumeclaim_status_phase,kube_pod_info,kube_pod_start_time,kube_pod_completion_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_container_status_waiting,kube_pod_container_status_waiting_reason,kube_pod_container_status_running,kube_pod_container_state_started,kube_pod_container_status_terminated,kube_pod_container_status_terminated_reason,kube_pod_container_status_last_terminated_reason,kube_pod_container_status_ready,kube_pod_container_status_restarts_total,kube_pod_overhead_cpu_cores,kube_pod_overhead_memory_bytes,kube_pod_created,kube_pod_deletion_timestamp,kube_pod_init_container_info,kube_pod_init_container_status_waiting,kube_pod_init_container_status_waiting_reason,kube_pod_init_container_status_running,kube_pod_init_container_status_terminated,kube_pod_init_container_status_terminated_reason,kube_pod_init_container_status_last_terminated_reason,kube_pod_init_container_status_ready,kube_pod_init_container_status_restarts_total,kube_pod_status_scheduled_time,kube_pod_status_unschedulable,kube_pod_spec_volumes_persistentvolumeclaims_readonly,kube_pod_container_resource_requests_cpu_cores,kube_pod_container_resource_requests_memory_bytes,kube_
```

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

```
# # Comma-separated list of Kubernetes label keys that will be used in
the resources' labels metric.
```

```
# # See metric-labels-allowlist in https://github.com/kubernetes/kube-
state-metrics/blob/main/docs/cli-arguments.md
```

```
# labels:
```

```
'cronjobs=[*],daemonsets=[*],deployments=[*],ingresses=[*],jobs=[*],namesp
aces=[*],nodes=[*],persistentvolumeclaims=[*],persistentvolumes=[*],pods=[
*],replicasets=[*],resourcequotas=[*],services=[*],statefulsets=[*]'
```

```
# # kube-state-metrics additional tolerations. Use the following
abbreviated single line format only.
```

```
# # No tolerations are applied by default
```

```
# # Example: '{key: taint1, operator: Exists, effect:
NoSchedule},{key: taint2, operator: Exists, effect: NoExecute}'
```

```
# tolerations: ''
```

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

```
# shards: '2'
```

```
# # Settings for the Events Log feature.
```



```

# logs:
# # Set runPrivileged to true if Fluent Bit fails to start, trying to
open/create its database.
# runPrivileged: 'false'

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

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

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

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

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

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

# # event-exporter client-side throttling
# # Set kubeBurst to roughly match your events per minute and
kubeQPS=kubeBurst/5
# # See https://github.com/resmoio/kubernetes-event-
exporter#troubleshoot-events-discarded-warning
# event-exporter-kubeQPS: 20
# event-exporter-kubeBurst: 100

```

```

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

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

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

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

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

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

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

# change-management:
# # change-observer-watch-rs CPU/Mem limits and requests.
# # See https://kubernetes.io/docs/concepts/configuration/manage-
resources-containers/
# cpuLimit: '1'

```

```

# memLimit: '1Gi'
# cpuRequest: '500m'
# memRequest: '500Mi'

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

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

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

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

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

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

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

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

# # change-observer-watch-ds additional tolerations. Use the following
abbreviated single line format only.
# # Inspect change-observer-watch-ds to view tolerations which are

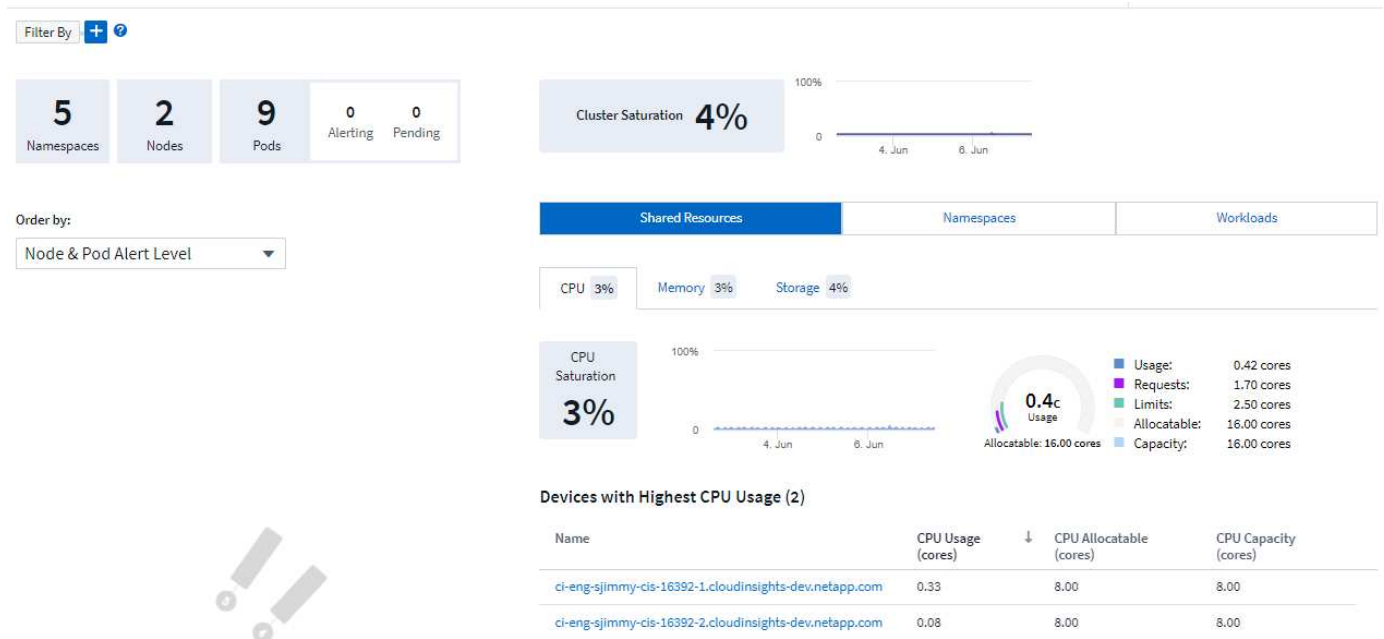
```

always present.

```
# # Example: '{key: taint1, operator: Exists, effect: NoSchedule},{key: taint2, operator: Exists, effect: NoExecute}'  
# watch-tolerations: ''
```

Kubernetes 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.

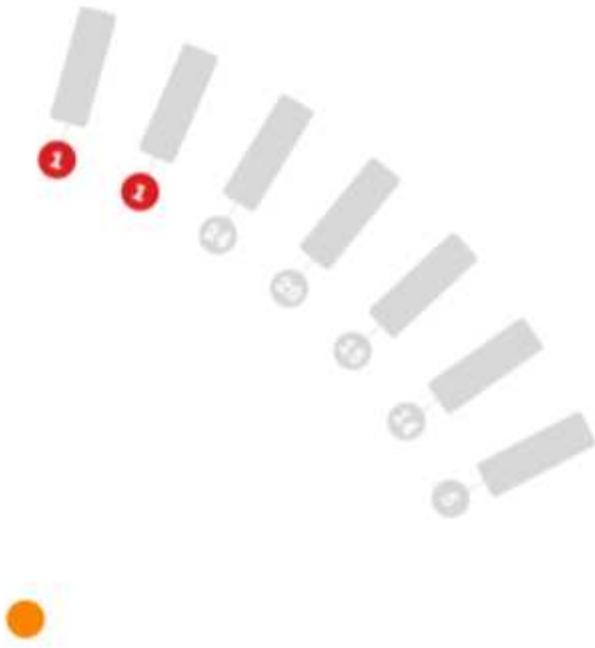
Shared Resources	Namespaces	Workloads	
Namespaces (5)			
Name ↓	CPU Usage (cores)	Memory Usage (GiB)	Workload Count
netapp-monitoring	0.25	0.38	4
kube-system	0.01	0.03	3
kube-public	0.00	0.00	0
kube-node-lease	0.00	0.00	0
default	0.00	<0.01	1

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 "힐"



UNSCHEDULED 1

ALERTING PODS 2 NODES 7

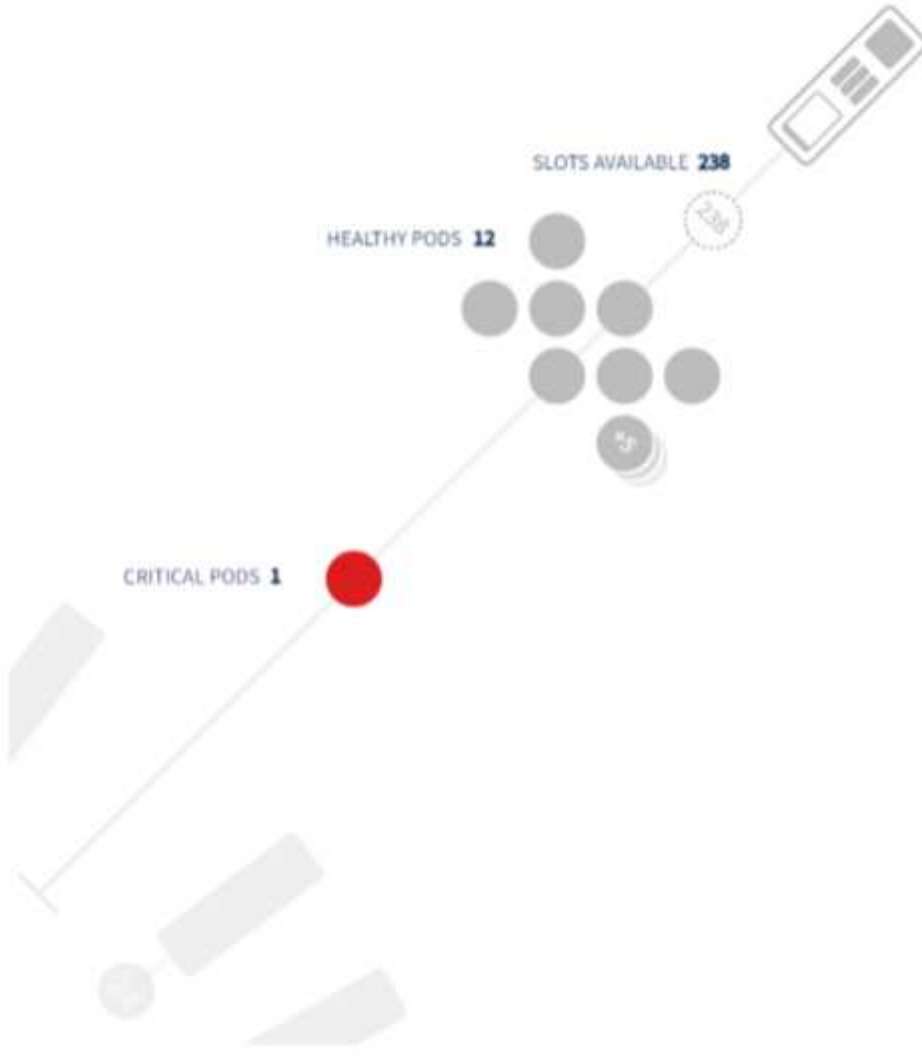
클러스터 "Wheel" 섹션은 노드 및 POD 상태를 한눈에 파악할 수 있도록 하여 자세한 정보를 확인할 수 있습니다. 클러스터에 페이지의 이 영역에 표시할 수 있는 것보다 많은 노드가 포함된 경우 사용 가능한 버튼을 사용하여 휠을 돌릴 수 있습니다.

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

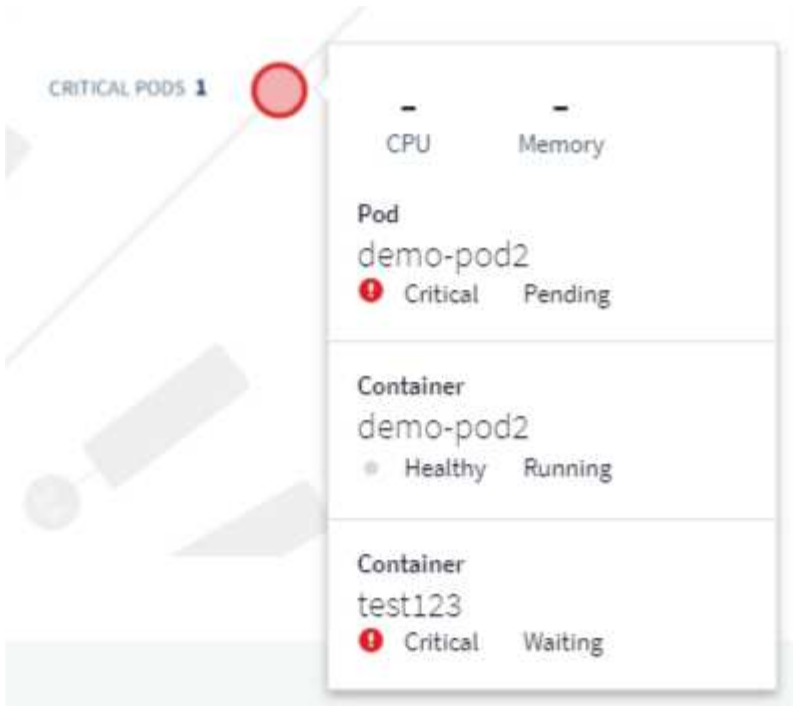
POD(원) 또는 Node(막대) 위로 마우스를 이동하면 노드의 보기가 확장됩니다.



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.

- 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 네트워크 성능 모니터링 및 맵을 사용하려면 먼저 이 옵션을 사용하도록 클러스터를 구성해야 ["NetApp Kubernetes 모니터링 운영자"](#)합니다. 오퍼레이터 배포 중에 "네트워크 성능 및 맵" 확인란을 선택하여 활성화합니다. Kubernetes 랜딩 페이지로 이동하여 "배포 수정"을 선택하여 이 옵션을 활성화할 수도 있습니다.

 **kubernetes**
Kubernetes

Configure Data Acquisition

Review Kubernetes cluster information and choose additional data to collect.

Cluster Information		
Kubernetes Cluster stream8	Network Performance and Map Disabled	Events Log Disabled

Deployment Options [Need Help?](#)

- Network Performance and Map
- Events Log

[Complete Setup](#)

모니터

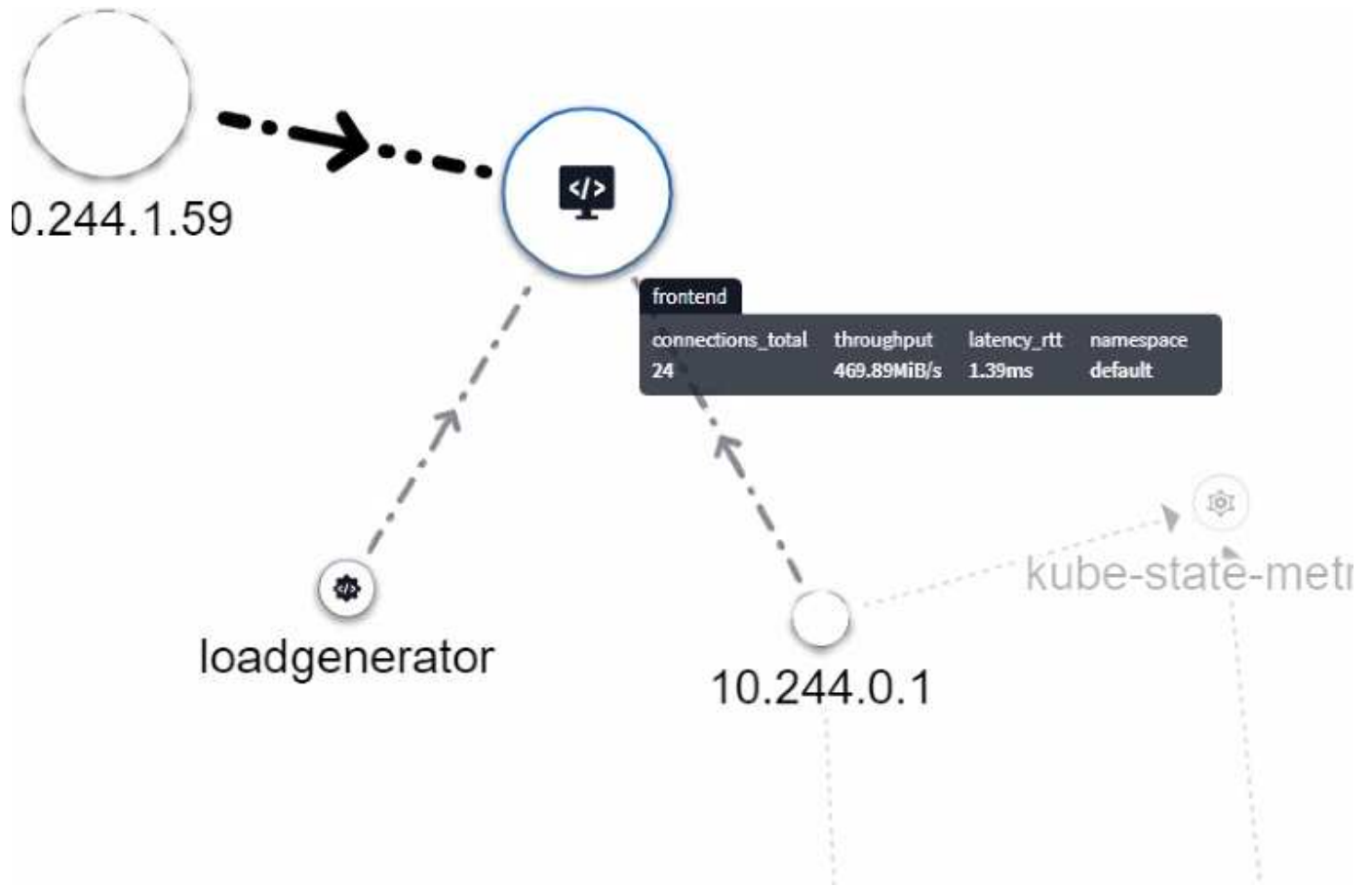
워크로드 맵을 사용하여 "모니터"정보를 파생시킵니다. Data Infrastructure Insights는 다양한 기본 Kubernetes Monitor를 제공합니다(이는 기본적으로 `_ Paused _`일 수 있습니다. 원하는 모니터를 *Resume*(예: 활성화)하거나, Kubernetes 객체에 대한 사용자 지정 모니터를 생성할 수 있습니다. 이 경우 워크로드 맵도 사용됩니다.

아래의 모든 오브젝트 유형에 대해 Data Infrastructure Insights 메트릭 경고를 생성할 수 있습니다. 데이터가 기본 개체 유형별로 그룹화되어 있는지 확인합니다.

- Kubernetes 워크로드
- Kubernetes 발병
- kubernetes.deployment
- 쿠버네티스.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.

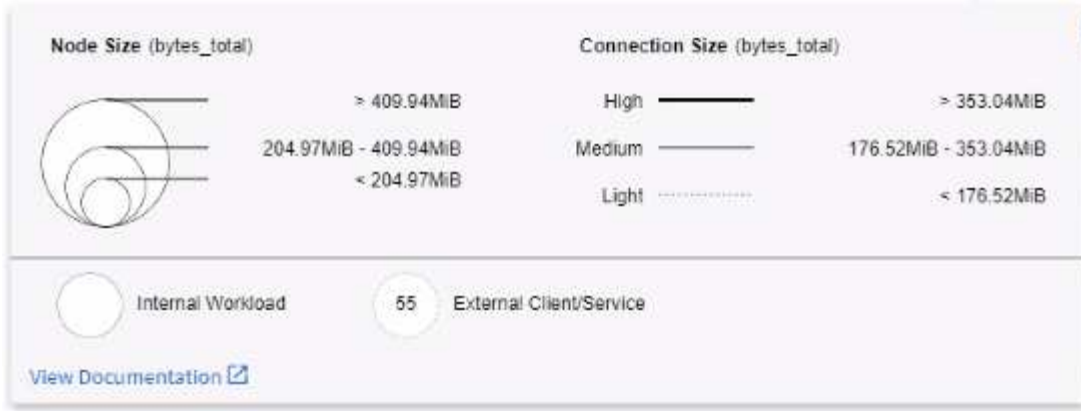


원 안에 있는 아이콘은 다양한 서비스 유형을 나타냅니다. 아이콘은 기본 개체에 가 있는 경우에만 라벨 표시됩니다.



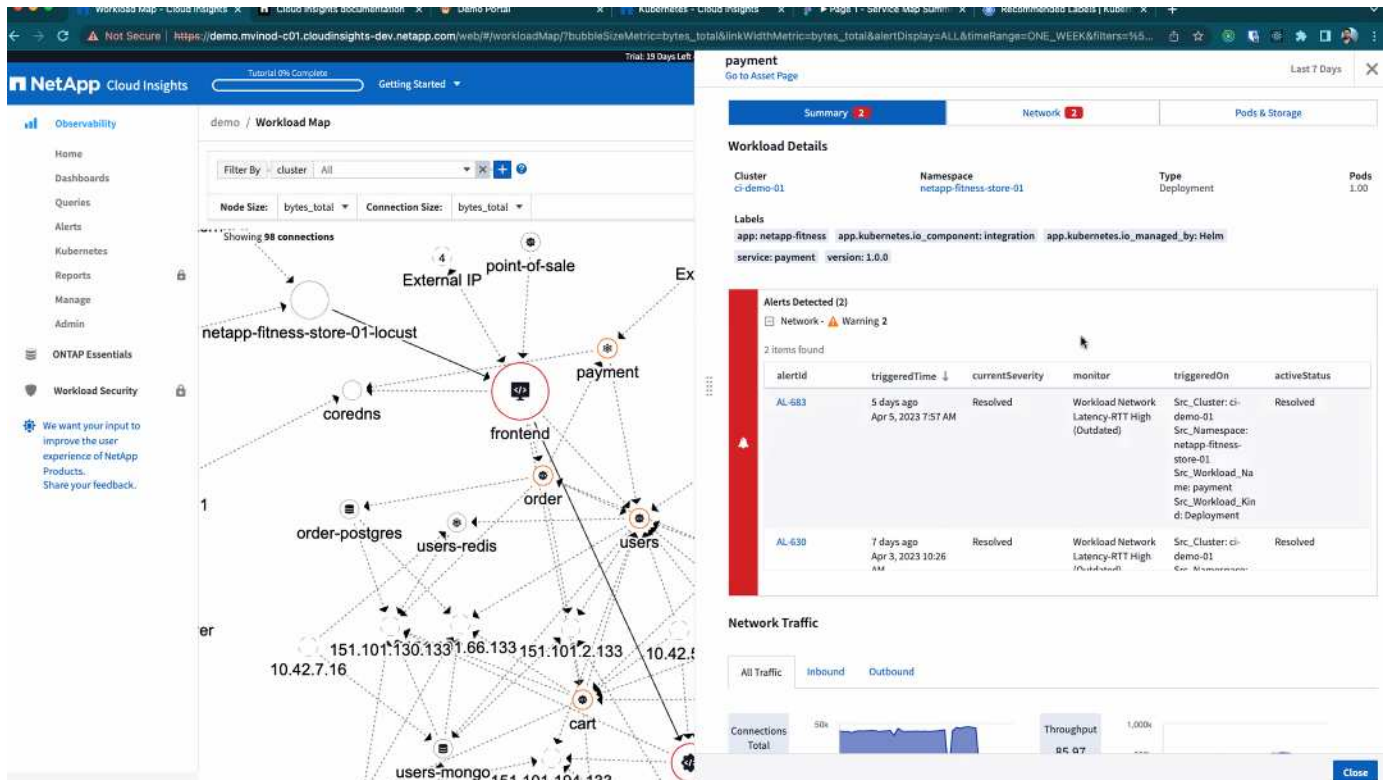
각 원의 크기는 노드 크기를 나타냅니다. 이러한 크기는 상대적이며 브라우저 확대/축소 수준 또는 화면 크기가 실제 원형 크기에 영향을 줄 수 있습니다. 같은 방법으로, 트래픽 회선 스타일은 연결 크기를 한 눈에 볼 수 있게 합니다. 굵은 실선은 트래픽이 높고 밝은 점선은 트래픽이 적습니다.

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



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.



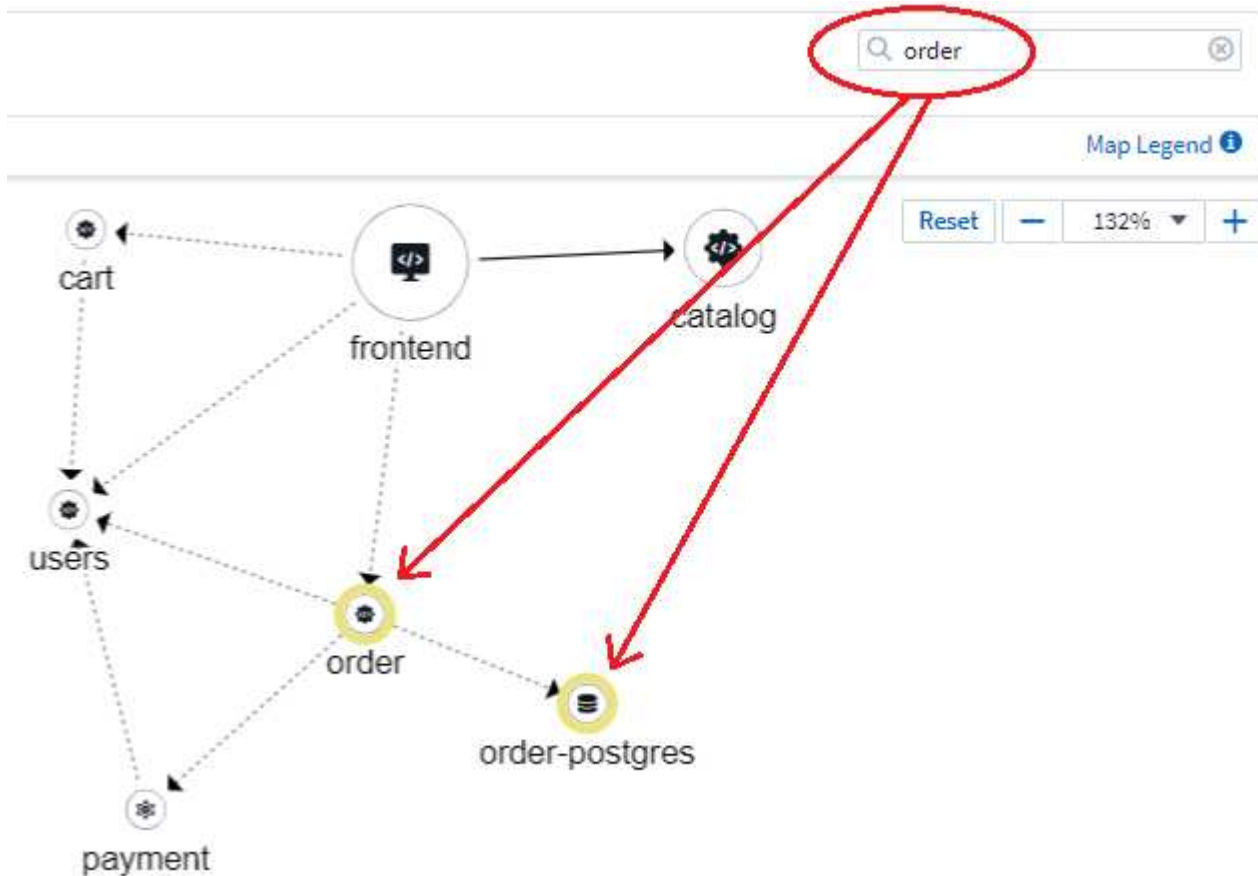
찾기 및 필터링

다른 Data Infrastructure Insights 기능과 마찬가지로 필터를 쉽게 설정하여 원하는 특정 오브젝트 또는 워크로드 특성에 집중할 수 있습니다.

Filter By: cluster All scope_cluster All

Node Size: bytes_total Connection Size: bytes_total

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`
 - 레이블 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:

- Data Infrastructure Insights는 DNS 확인된 이름을 구문 분석하여 서비스 유형을 추출합니다.

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.

Summary	Network	Pods & Storage
---------	----------------	----------------

Network Activities - Inbound (1) ⚙

src_workload...	src_namespace	src_workload_...	throughpu... ↓	connections_t...	latency_rtt (ms)	tcp_retransmit...
netapp-fitness...	locust	Deployment	14,193,748.78	653.19	3.74	2,578.00

Network Activities - Outbound (4) ⚙

dst_workloa...	dst_namespace	dst_workload_...	throughpu... ↓	connections_t...	latency_rtt (ms)	tcp_retransmit...
catalog	netapp-fitness-...	Deployment	14,166,417.02	2,425.07	149.37	13,850.00
cart	netapp-fitness-...	Deployment	12,479.90	638.97	65.10	0.00
order	netapp-fitness-...	Deployment	4,515.16	161.84	65.07	0.00

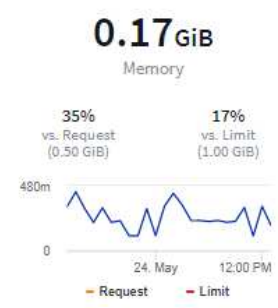
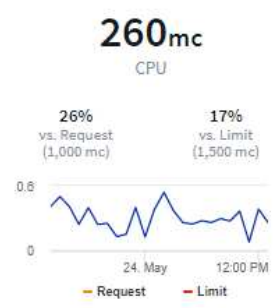
마지막으로, _자산 페이지로 이동_ 을(를) 선택하면 워크로드에 대한 상세 자산 랜딩 페이지가 열립니다.

Filter By + ?

2/2
Pods: Current / Desired

2 Up-to-date 0 Unavailable

Namespace netapp-fitness-store-01	Type Deployment	Date Created Apr 11, 2023 11:34 AM
Labels -		



0.00GiB
Total PVC Capacity claimed

Highest CPU Demand by Pod

- 132.76m frontend-7...9f8f-284kb
- 127.55m frontend-7...9f8f-gd8mk

Highest Memory Demand by Pod

- 0.09 GiB frontend-7...9f8f-284kb
- 0.09 GiB frontend-7...9f8f-gd8mk

Pods (2)

Pod Name ↑	Status	Healthy Containers	cpu_usage_nanocores (mc)	memory_rss_bytes (GiB)
frontend-7fccd9f8f-284kb	● Healthy Running	1 of 1	133	0.09
frontend-7fccd9f8f-gd8mk	● Healthy Running	1 of 1	128	0.09

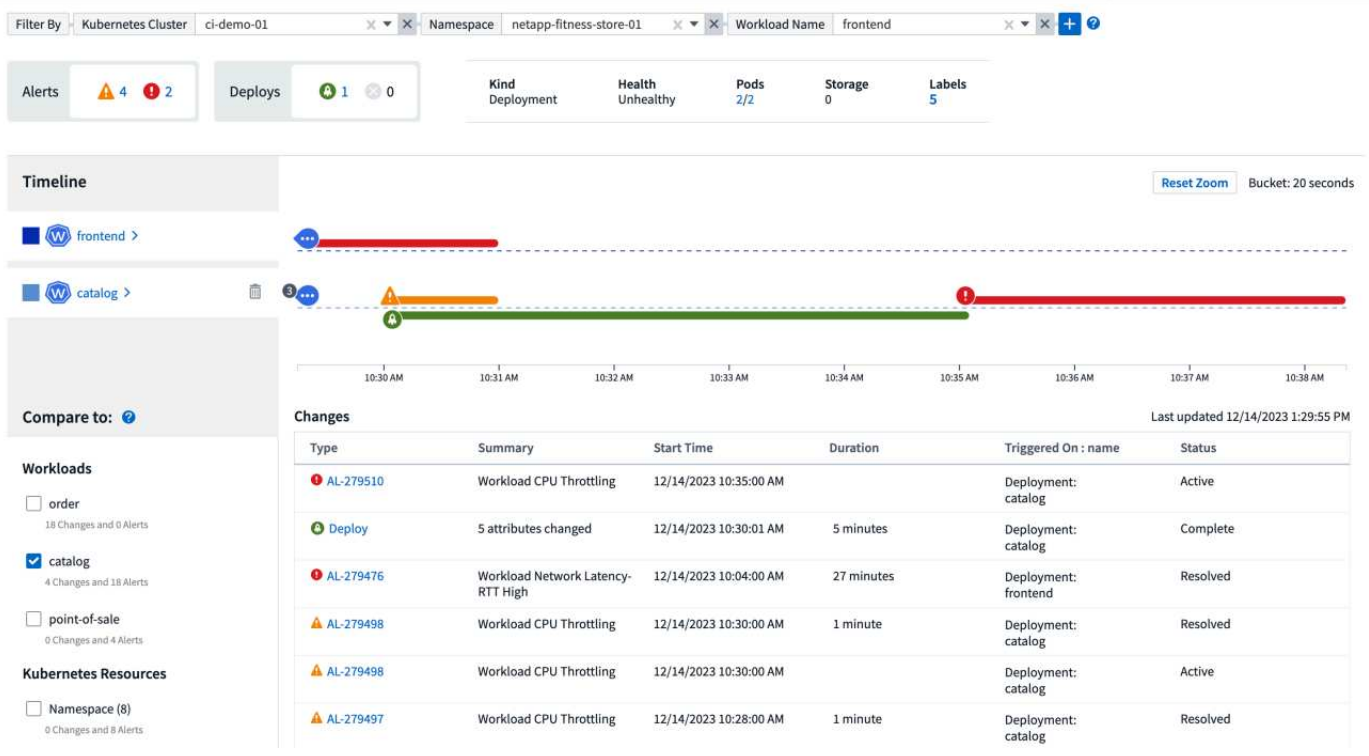
Kubernetes 변경 분석

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.

변경 분석은 어떻게 도움이 됩니까?

- 멀티 테넌트 Kubernetes 환경에서는 잘못 구성된 변경으로 인해 중단이 발생할 수 있습니다. Change Analytics는 워크로드 상태와 구성 변경의 상태를 보고 상호 연관시킬 수 있는 단일 창을 제공하여 이러한 작업을 지원합니다. 이는 동적 Kubernetes 환경 문제를 해결하는 데 도움이 될 수 있습니다.

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



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

필터링

Data Infrastructure Insights의 모든 기능과 마찬가지로 변경 목록을 필터링하는 것도 직관적입니다. 페이지 상단에서 Kubernetes 클러스터, 네임스페이스 또는 워크로드의 값을 입력하거나 선택하거나 [+] 버튼을 선택하여 자체 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.

Filter By: Kubernetes Cluster stream-54 | Namespace: kube-system | Workload Name: coredns

Alerts: 0 Warning, 8 Critical | Deploys: 0 Success, 0 Failed

Kind: Deployment | Health: Healthy | Pods: 1/1 | Storage: 0 | Labels: 3

Timeline: Bucket: 6 minutes

Timeline visualization showing alerts for coredns workload.

Compare to: ?

Type	Summary	Start Time	Duration	Triggered On : name	Status
AL-2982989	once Workload Down copy	11/28/2023 3:07:00 PM	1 minute	Deployment: coredns	Resolved
AL-2982989	once Workload Down copy	11/28/2023 3:07:00 PM		Deployment: coredns	Active
AL-2982887	once Workload Down copy	11/28/2023 3:01:00 PM	1 minute	Deployment: coredns	Resolved
AL-2982887	once Workload Down copy	11/28/2023 3:01:00 PM		Deployment: coredns	Active
AL-2982782	once Workload Down copy	11/28/2023 2:57:00 PM	0 milliseconds	Deployment: coredns	Resolved
AL-2982782	once Workload Down copy	11/28/2023 2:57:00 PM		Deployment: coredns	Active
AL-2982441	once Workload Down copy	11/28/2023 2:32:00 PM	1 minute	Deployment: coredns	Resolved
AL-2982441	once Workload Down copy	11/28/2023 2:32:00 PM		Deployment: coredns	Active

빠른 상태

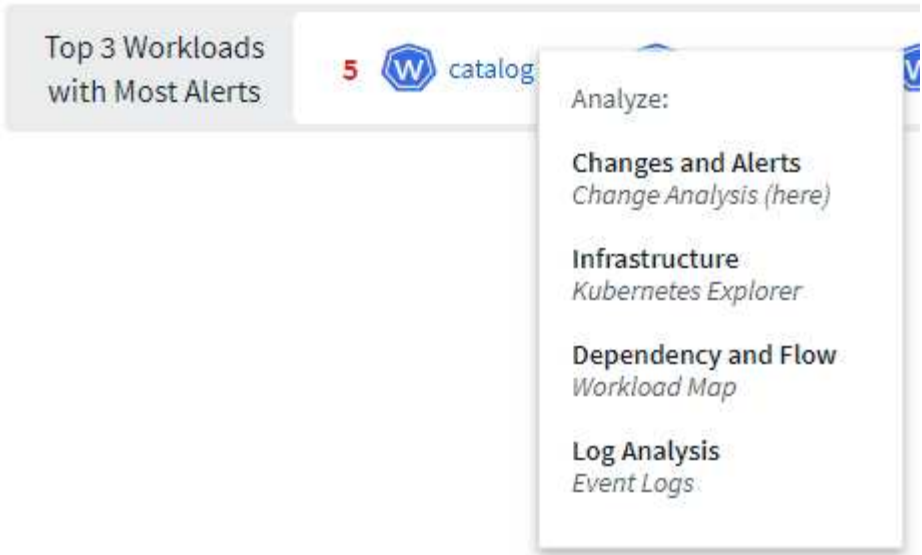
필터링 영역 아래에는 여러 개의 상위 수준 표시기가 있습니다. 왼쪽에는 경고(Warning) 및 위험(Critical)의 수가 표시됩니다. 이 수에는 `_Active_` 와 `_Resolved_alerts_` 가 포함됩니다. `Active_alerts`만 보려면 "Status"에 대한 필터를 설정하고 "Active"를 선택합니다.

Alerts: 6 Warning, 17 Critical

배포 상태도 여기에 표시됩니다. 기본적으로 `Started, Complete` 및 `_Failed_deployments_`의 개수가 표시됩니다. `Only_Failed_Deployments`를 보려면 "Status"에 필터를 설정하고 "Failed"를 선택합니다.

Deploys: 36 Success, 4 Failed

다음 워크로드로는 경고가 가장 많은 상위 3개 워크로드가 있습니다. 각 워크로드 옆의 빨간색 숫자는 해당 워크로드와 관련된 알람 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.

저작권 정보

Copyright © 2025 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.