



# Kubernetes

## Data Infrastructure Insights

NetApp  
January 17, 2025

# 目錄

Kubernetes .....	1
Kubernetes 叢集概觀 .....	1
安裝或升級 NetApp Kubernetes 監控操作員之前 .....	2
Kubernetes 監控營運商的安裝與組態 .....	6
Kubernetes 監控操作員組態選項 .....	21
Kubernetes 叢集詳細資料頁面 .....	33
Kubernetes 網路效能監控與地圖 .....	37
Kubernetes 變更分析 .....	44

# Kubernetes

## Kubernetes 叢集概觀

資料基礎架構 Insights Kubernetes Explorer 是一個強大的工具、可顯示 Kubernetes 叢集的整體健全狀況和使用情形、並可讓您輕鬆深入調查調查領域。

按一下\*儀表板> Kubernetes Explorer\*即可開啟Kubernetes叢集清單頁面。本概要頁面包含租戶上 Kubernetes 叢集的表格。



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

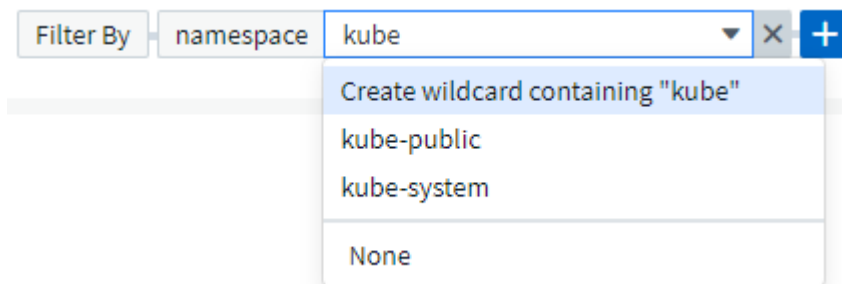
### 叢集清單

叢集清單會針對租戶上的每個叢集顯示下列資訊：

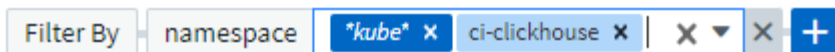
- 叢集\*名稱\*。按一下叢集名稱將會開啟該叢集["詳細資料頁"](#)的。
- \*飽和\*百分比。整體飽和是CPU、記憶體或儲存飽和的最高值。
- 叢集中\*節點\*的數目。按一下此編號將會開啟「節點」清單頁面。
- 叢集中的\* pod \*數。按一下此號碼將會開啟Pod清單頁面。
- 叢集中的\*命名空間\*數目。按一下此號碼將會開啟「命名空間」清單頁面。
- 叢集中\*工作負載\*的數目。按一下此數字將會開啟工作負載清單頁面。

### 精簡篩選條件

當您開始篩選時、您會看到根據目前文字建立\*萬用字元篩選器\*的選項。選取此選項會傳回符合萬用字元運算式的所有結果。您也可以使用Not或and建立\* Expressions \*、或是選取「無」選項來篩選欄位中的null值。



根據萬用字元或運算式（例如 不、和、「無」等）會在篩選欄位中以深藍色顯示。您直接從清單中選取的項目會以淺藍色顯示。



Kubernetes篩選器是關聯式的、也就是說、如果您在特定節點頁面上、pod\_name篩選器只會列出與該節點相關的Pod。此外、如果您套用特定命名空間的篩選器、則pod\_name篩選器只會在該節點\_and中列出該命名空間中的Pod。

請注意、萬用字元與運算式篩選功能可搭配文字或清單使用、但不能搭配數值、日期或布爾值使用。

## 安裝或升級 NetApp Kubernetes 監控操作員之前

安裝或升級之前，請先閱讀此資訊"[Kubernetes 監控營運者](#)"。

元件	需求
Kubernetes版本	Kubernetes v1.20 及更高版本。
Kubernetes 配送	AWS Elastic Kubernetes Service (EKS) Azure Kubernetes Service (aks) Google Kubernetes Engine (GKE) Red Hat OpenShift Rancher Kubernetes Engine (RKE) VMware Tanzu
Linux 作業系統	Data Infrastructure Insights 不支援以 Arm64 架構執行的節點。網路監控：必須執行 Linux 核心版本 4.18.0 或更新版本。不支援光子作業系統。
標籤	Data Infrastructure Insights 可指定 Kubernetes 節點選取器、在這些平台上尋找下列 Kubernetes 標籤、以支援對執行 Linux 的 Kubernetes 節點進行監控： Kubernetes v1.20 及更高版本：Kubernetes.IO/OS = Linux Rancher + Cattle .IO 做為協調 / Kubernetes 平台：Cattle .IO/OS = Linux
命令	curl 和 kubectl 命令必須可用；若要獲得最佳結果、請將這些命令新增至路徑。
連線能力	Kubectl CLI 已設定為與目標 K8s 叢集通訊、並可與您的 Data Infrastructure Insights 環境進行網際網路連線。如果您在安裝期間位於 Proxy 後方、請依照「操作員安裝」一節中的指示" <a href="#">設定Proxy支援</a> "進行。若要準確地進行稽核和資料報告、請使用網路時間傳輸協定 (NTP) 或簡易網路時間傳輸協定 (SNTP) 來同步代理機器上的時間。
其他	如果您是在 OpenShift 4.6 或更新版本上執行，則除了確保符合這些先決條件之外，還必須遵循" <a href="#">OpenShift指示</a> "。
API Token	如果您要重新部署操作員（亦即您正在更新或取代它）、則不需要建立新的 API 權杖；您可以重新使用先前的權杖。

### 開始之前要注意的重要事項

如果您是與一起執行Proxy，擁有自訂儲存庫或正在使用OpenShift，請仔細閱讀下列各節。

另請參閱[權限](#)。

## 設定Proxy支援

您可以在兩個地方使用租戶上的 Proxy 來安裝 NetApp Kubernetes 監控操作員。這些可能是相同或獨立的Proxy系統：

- 執行安裝程式碼片段（使用「cURL」）時需要 Proxy、以將執行程式碼片段的系統連線至您的 Data Infrastructure Insights 環境
- 目標 Kubernetes 叢集與您的 Data Infrastructure Insights 環境通訊所需的 Proxy

如果您使用上述任一或兩者的 Proxy、若要安裝 NetApp Kubernetes 作業系統、您必須先確定您的 Proxy 已設定為允許與您的 Data Infrastructure Insights 環境進行良好通訊。例如、從您想要安裝操作員的伺服器 / VM、您必須能夠存取 Data Infrastructure Insights、並從 Data Infrastructure Insights 下載二進位檔。

對於用來安裝NetApp Kubernetes作業監視器的Proxy、請先設定\_http代理伺服器/https代理伺服器環境變數、然後再安裝「運算子」。在某些Proxy環境中、您可能也需要設定\_no\_proxyEnvironments\_變數。

若要設定變數、請在系統\*安裝NetApp Kubernetes監控操作員之前\*執行下列步驟：

1. 為目前使用者設定\_https\_proxy\_和/或\_https\_proxy\_環境變數：
  - a. 如果正在設定的Proxy沒有驗證（使用者名稱/密碼）、請執行下列命令：

```
export https_proxy=<proxy_server>:<proxy_port>  
.. 如果正在設定的Proxy具有驗證（使用者名稱/密碼）、請執行下列命令：
```

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

對於 Kubernetes 叢集用來與資料基礎架構 Insights 環境通訊的 Proxy、請在閱讀完所有指示之後、安裝 NetApp Kubernetes Monitoring Operator。

在部署 NetApp Kubernetes Monitoring Operator 之前、請先在 operator-config.yaml 中設定 AgentConfiguration 的 Proxy 區段。

```

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
  ...
  ...

```

## 使用自訂或私有泊塢視窗儲存庫

根據預設、NetApp Kubernetes Monitoring Operator 會從 Data Infrastructure Insights 儲存庫擷取容器映像。如果您使用 Kubernetes 叢集做為監控目標、且該叢集設定為僅從自訂或私有 Docker 儲存庫或容器登錄中提取容器映像、則必須設定對 NetApp Kubernetes Monitoring Operator 所需容器的存取權。

從 NetApp Monitoring Operator 安裝方塊執行「影像提取片段」。此命令會登入 Data Infrastructure Insights 儲存庫、拉出操作員的所有影像相依性、然後登出 Data Infrastructure Insights 儲存庫。出現提示時、請輸入提供的儲存庫暫存密碼。此命令會下載操作員所使用的所有影像、包括選用功能。請參閱下方、瞭解這些影像的用途。

## 核心營運者功能與 Kubernetes 監控

- NetApp 監控
- Kube-RBAC 代理程式
- Kube-state 指標
- Telegraf
- 無 distrouse-root 使用者

## 事件記錄

- Fluent 位元
- Kubernetes-event-Exporter

## 網路效能與地圖

- CI-net-觀察者

根據您的企業原則、將「operator」泊塢視窗影像推送到您的「私有/本機/企業」泊塢視窗儲存庫。確保儲存庫中這些映像的映像標記和目錄路徑與 Data Infrastructure Insights 儲存庫中的映像標記和目錄路徑一致。

在 `operer-deployment.yaml` 中編輯監控營運者部署、並修改所有映像參照以使用您的私有 Docker 儲存庫。

```
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>
```

在 `operer-config.yaml` 中編輯 `AgentConfiguration`、以反映新的泊塢視窗 repo 位置。為您的私有儲存庫建立新的 `imagePullSecret`、如需詳細資料、請參閱 <https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/>

```
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指示

如果您是在 OpenShift 4.6 或更新版本上執行、則必須在 `operer-config.yaml` 中編輯 `AgentConfiguration`、才能啟用 `runPrivileged` 設定：

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

OpenShift可能會實作額外的安全層級、以封鎖對某些Kubernetes元件的存取。

## 權限

如果您所監控的叢集包含沒有 `ClusterRole` 的自訂資源"要檢視的集合體"，您需要手動授予操作員這些資源的存取權，以便使用事件記錄來監控這些資源。

1. 在安裝之前或安裝之後、請先編輯 `operer-adder-permissions.yaml`、然後編輯資源 `<namespace>` 附加權限

2. 使用動詞 ["Get" 、 "watch " 、 "list" 建立所需的組和資源的新規則。請參閱 <https://kubernetes.io/docs/reference/access-authn-authz/rbac/>
3. 將變更套用至叢集


## Kubernetes 監控營運商的安裝與組態

Data Infrastructure Insights 為 Kubernetes 集合提供 \* Kubernetes Monitoring Operator\* 。瀏覽至 \* Kubernetes > Collectores > +Kubernetes Collector\* 以部署新的運算子。

### 安裝 Kubernetes Monitoring Operator 之前

安裝或升級 Kubernetes Monitoring Operator 之前，請先參閱"先決條件"文件。

### 安裝 Kubernetes Monitoring Operator

 **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
<input type="text" value="clustername"/>	<input type="text" value="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

在 **Kubernetes** 上安裝 **Kubernetes Monitoring Operator Agent** 的步驟：

1. 輸入唯一的叢集名稱和命名空間。如果您升級來自先前的 Kubernetes 運算子，請使用相同的叢集名稱和命名空間。
2. 一旦輸入這些指令碼、您就可以將 Download Command 片段複製到剪貼簿。
3. 將程式碼片段貼到 `_bash_` 視窗中並執行。將下載操作員安裝檔案。請注意、程式碼片段具有獨特的金鑰、有效時間為24小時。
4. 如果您有自訂或私有儲存庫、請複製選用的「影像」抽取片段、將其貼入 `bash Shell` 並加以執行。影像擷取完成後、請將其複製到您的私有儲存庫。請務必維持相同的標記和資料夾結構。更新 `operer-deployment.yaml` 中的路徑、以及 `operer-config.yaml` 中的泊塢視窗儲存庫設定。
5. 如有需要、請檢閱可用的組態選項、例如 Proxy 或私有儲存庫設定。您可以閱讀更多關於"[組態選項](#)"的資訊。
6. 準備好之後、請複製 KUBECTl 套用程式碼片段、下載並執行、以部署操作員。
7. 安裝會自動繼續進行。完成後、按一下 `_ 下一步 _` 按鈕。
8. 安裝完成後、按一下 `_ 下一步 _` 按鈕。請務必刪除或安全儲存 `operer-Secrets .yaml` 檔案。

如果您使用的是 Proxy ，請參閱關於[設定 Proxy](#)。

如果您有自訂儲存庫，請閱讀關於[使用自訂 / 私有泊塢視窗儲存庫](#)。

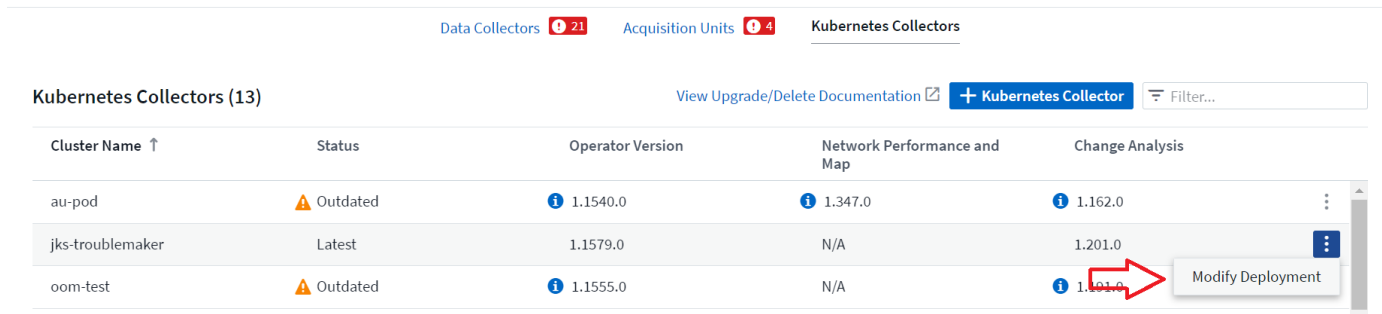
## Kubernetes 監控元件

資料基礎架構 Insights Kubernetes 監控由四個監控元件組成：

- 叢集度量
- 網路效能與地圖（選用）
- 事件記錄（選用）
- 變更分析（選用）

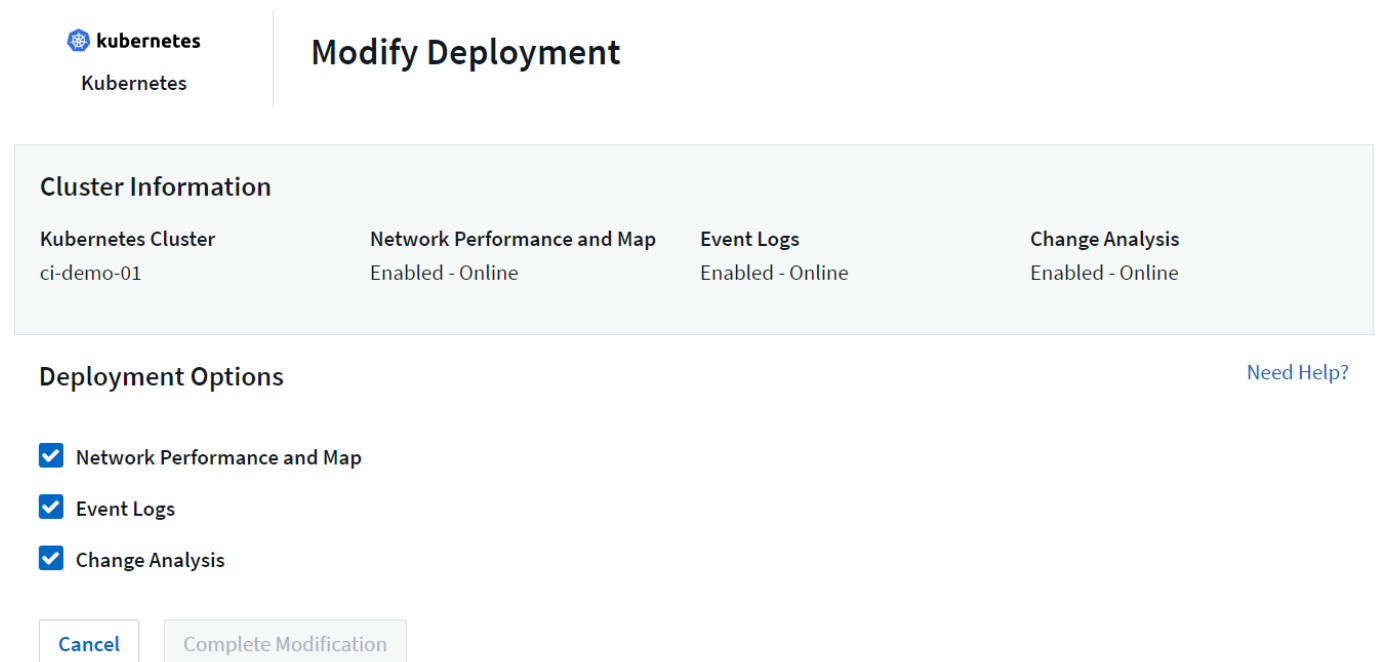
根據預設、每個 Kubernetes 收集器都會啟用上述選用元件；如果您決定不需要特定收集器的元件、您可以瀏覽至 **Kubernetes > Collectors**、然後從畫面右側的收集器「三點」功能表中選取「修改部署」、將其停用。

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	1.161.0	Modify Deployment

畫面會顯示每個元件的目前狀態、並可讓您視需要停用或啟用該收集器的元件。



**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](#)

## 升級至最新的 Kubernetes Monitoring Operator

判斷現有運算子是否存在 AgentConfiguration（如果您的命名空間不是預設的 `_NetApp-monitoring`、請改用適當的命名空間）：

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

如果存在 AgentConfiguration :

- 安裝現有運算子的最新運算子。
  - 確保您使用的是擷取最新的容器映像自訂儲存庫。

如果 AgentConfiguration 不存在：

- 請記下資料基礎架構洞見所識別的叢集名稱（如果您的命名空間不是預設的 NetApp 監控、請改用適當的命名空間）：

```
kubectl -n netapp-monitoring get agent -o jsonpath='{.items[0].spec.cluster-name}'
```

\* 建立現有運算子的備份（如果您的命名空間不是預設的 NetApp 監控功能、請改用適當的命名空間）：

```
kubectl -n netapp-monitoring get agent -o yaml > agent_backup.yaml
```

\* <<to-remove-the-kubernetes-monitoring-operator,解除安裝>>現有的運算子。  
\* <<installing-the-kubernetes-monitoring-operator,安裝>>最新的運算子。

- 請使用相同的叢集名稱。
- 下載最新的 Operator YAML 檔案之後、請先將 agent\_backup.yaml 中的任何自訂項目連接至下載的 operator-config.yaml、然後再進行部署。
- 確保您使用的是擷取最新的容器映像自訂儲存庫。

## 停止並啟動 Kubernetes 監控操作員

若要停止 Kubernetes 監控操作員：

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

若要啟動 Kubernetes Monitoring 運算子：

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

## 正在解除安裝

## 移除 Kubernetes Monitoring Operator

請注意、Kubernetes Monitoring Operator 的預設命名空間是「NetApp-Monitoring」。如果您已設定自己的命名空間、請在這些名稱空間以及所有後續命令和檔案中取代該命名空間。

可使用下列命令解除安裝較新版本的監控操作員：

```
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>
```

如果監控操作員部署在其專屬命名空間中、請刪除命名空間：

```
kubectl delete ns <NAMESPACE>
如果第一個命令傳回「找不到資源」、請依照下列指示解除安裝舊版監控操作員。
```

依序執行下列每個命令。視您目前的安裝情況而定、其中一些命令可能會傳回「找不到物件」訊息。這些訊息可能會被安全忽略。

```
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>
```

如果先前已建立安全性內容限制：

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

## 關於Kube-state指標

NetApp Kubernetes監控操作員會安裝自己的Kube-態 指標、以避免與任何其他執行個體發生衝突。

如需有關 Kube-State-Metrics 的資訊，請參閱["本頁"](#)。

## 設定 / 自訂操作員

這些區段包含自訂運算子組態、使用 Proxy、使用自訂或私有泊塢視窗儲存庫或使用 OpenShift 的相關資訊。

### 組態選項

最常修改的設定可在 *AgentConfiguration* 自訂資源中進行設定。您可以編輯 *operer-config.yaml* 檔案、在部署運算子之前編輯此資源。此檔案包含設定的註解範例。如需最新版的運算子，請參閱清單“[可用的設定](#)”。

您也可以使用下列命令在部署運算子之後編輯此資源：

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

若要判斷您部署的營運者版本是否支援 *AgentConfiguration*、請執行下列命令：

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

如果您看到「錯誤來自伺服器 ( NotFound )」訊息、則必須先升級您的營運商、才能使用 *AgentConfiguration*。

### 設定Proxy支援

您可以在兩個地方使用租戶上的 Proxy 來安裝 Kubernetes Monitoring Operator。這些可能是相同或獨立的Proxy系統：

- 執行安裝程式碼片段（使用「cURL」）時需要 Proxy、以將執行程式碼片段的系統連線至您的 Data Infrastructure Insights 環境
- 目標 Kubernetes 叢集與您的 Data Infrastructure Insights 環境通訊所需的 Proxy

如果您使用上述任一種或兩者的 Proxy、為了安裝 Kubernetes Operating Monitor、您必須先確定您的 Proxy 已設定為允許與 Data Infrastructure Insights 環境進行良好的通訊。如果您有代理伺服器、而且可以從想要安裝運算子的伺服器 / VM 存取 Data Infrastructure Insights、則您的 Proxy 可能已正確設定。

對於用於安裝 Kubernetes 作業系統的 Proxy、在安裝運算子之前、請先設定 `_http` 代理伺服器 / `https` 代理伺服器 `_` 環境變數。在某些Proxy環境中、您可能也需要設定 `_no_proxyEnvironments` 變數。

若要設定變數、請在系統 \* 上 \* 安裝 Kubernetes Monitoring 運算子之前 \* 執行下列步驟：

1. 為目前使用者設定 `_https_proxy` 和/或 `_https_proxy` 環境變數：
  - a. 如果正在設定的Proxy沒有驗證（使用者名稱/密碼）、請執行下列命令：

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

.. 如果正在設定的Proxy具有驗證（使用者名稱/密碼）、請執行下列命令：

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

對於 Kubernetes 叢集用來與 Data Infrastructure Insights 環境通訊的 Proxy、請在閱讀完所有指示之後、安裝 Kubernetes Monitoring Operator。

在部署 Kubernetes Monitoring Operator 之前、請先在 operator-config.yaml 中設定 AgentConfiguration 的 Proxy 區段。

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

### 使用自訂或私有泊塢視窗儲存庫

根據預設、Kubernetes Monitoring Operator 會從 Data Infrastructure Insights 儲存庫中擷取容器映像。如果您使用 Kubernetes 叢集做為監控目標、且該叢集設定為僅從自訂或私有 Docker 儲存庫或容器登錄中提取容器映像、則必須設定 Kubernetes Monitoring Operator 所需的容器存取權。

從 NetApp Monitoring Operator 安裝方塊執行「影像提取片段」。此命令會登入 Data Infrastructure Insights 儲存庫、拉出操作員的所有影像相依性、然後登出 Data Infrastructure Insights 儲存庫。出現提示時、請輸入提供的儲存庫暫存密碼。此命令會下載操作員所使用的所有影像、包括選用功能。請參閱下方、瞭解這些影像的用途。

### 核心營運者功能與 Kubernetes 監控

- NetApp 監控

- CI-KRBA-Proxy
- CI-ksm
- CI-Telegraf
- 無 distrouse-root 使用者

#### 事件記錄

- CI-Fluent 位元
- CI-Kubernetes-event-Exporter

#### 網路效能與地圖

- CI-net-觀察者

根據您的企業原則、將「operator」泊塢視窗影像推送到您的「私有/本機/企業」泊塢視窗儲存庫。確保儲存庫中這些映像的映像標記和目錄路徑與 Data Infrastructure Insights 儲存庫中的映像標記和目錄路徑一致。

在 `operer-deployment.yaml` 中編輯監控營運者部署、並修改所有映像參照以使用您的私有 Docker 儲存庫。

```
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>
```

在 `operer-config.yaml` 中編輯 `AgentConfiguration`、以反映新的泊塢視窗 repo 位置。為您的私有儲存庫建立新的 `imagePullSecret`、如需詳細資料、請參閱 <https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/>

```
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指示

如果您是在 OpenShift 4.6 或更新版本上執行、則必須在 `operer-config.yaml` 中編輯 `AgentConfiguration`、才能啟用 `runPrivileged` 設定：

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

OpenShift可能會實作額外的安全層級、以封鎖對某些Kubernetes元件的存取。

## 公差和污染

*NetApp-CI-telegraf-DS*、*NetApp-CI-Fluent-bit-DS* 和 *NetApp-CI-net-觀察者 -L4-DS* 示範必須在叢集中的每個節點上排程一個 Pod、以便正確收集所有節點上的資料。已將操作員配置為允許某些已知的 \* 污點 \*。如果在節點上配置了任何自定義污點，從而阻止 Pod 在每個節點上運行，則可以為這些污點創建 \*公差\* "在 [AgentConfiguration](#) 中"。如果您已將自訂污點套用至叢集中的所有節點、您也必須在操作員部署中新增必要的容錯功能、以便排程及執行操作員 Pod。

瞭解更多關於 Kubernetes "污染與容許"的資訊。

返回"[NetApp Kubernetes 監控操作員安裝](#) \* 頁面"

## 關於機密的備註

若要移除 Kubernetes Monitoring Operator 檢視整個叢集機密的權限、請在安裝之前、從 *operer-setup.yaml* 檔案中刪除下列資源：

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

如果是升級、也請從叢集中刪除資源：

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

如果啟用變更分析、請修改 *AgentConfiguration* 或 *operer-config.yaml* 以取消變更管理區段的註解、並在變更管理區段下包含 `_kindsToIgnoreFromWatch`：「Secrets」。請注意此行中單引號和雙引號的存在和位置。

```
# 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"
...
```



## 驗證 Kubernetes 監控操作員影像簽名

操作員的映像及其部署的所有相關映像均由 NetApp 簽署。您可以在安裝前使用 CO 簽署 工具手動驗證映像，或設定 Kubernetes 接入控制器。如需更多詳細資料"[Kubernetes文件](#)"，請參閱。

用於驗證映像簽名的公開金鑰可在監控操作員安裝方塊的 `_` 選擇性：將操作員映像上傳至您的私有儲存庫 > 映像簽章公開金鑰 `_` 下取得

若要手動驗證影像簽名，請執行下列步驟：

1. 複製並執行「影像擷取片段」
2. 出現提示時，請複製並輸入儲存庫密碼
3. 儲存影像簽名公開金鑰（範例中的 DII 映像簽署 .pub）
4. 使用 CO 簽署 來驗證影像。請參閱以下代碼使用範例

```
$ 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}]
```

## 疑難排解

如果您在設定 Kubernetes 監控操作員時遇到問題、請嘗試下列事項：

問題：	試用：
我看不到Kubernetes持續Volume與對應的後端儲存設備之間的超連結/連線。我的Kubernetes持續Volume是使用儲存伺服器的主機名稱來設定。	請依照步驟解除安裝現有的Telegraf代理程式、然後重新安裝最新的Telegraf代理程式。您必須使用 Telegraf 2.0 版或更新版本、且必須主動監控您的 Kubernetes 叢集儲存設備、以獲得 Data Infrastructure Insights 。

<p>問題：</p> <p>我在記錄中看到類似以下內容的訊息： E0901 15 : 21 : 39.962145 1 反射器。前往： 178 ) k8s.io/kube 狀態指標 / 內部 / 儲存 / 建置器。前往： 352 : 無法列出 * v1.MutatingWebhookbuilder 組態：伺服器找不到所要求的資源 E0901 : 21 : 43.16178 伺服器無法找到 IO 資源 v1.16178 v1/816v1.v1.v1.v1.v1.v1.v1.v2.1*</p>	<p>試用：</p> <p>如果您執行Kubernetes版本低於1.20的Kubernetes 2.0.0版或更新版本之Kube-state度量、則可能會出現這些訊息。若要取得Kubernetes版本：_kubectI版本_若要取得Kube-st態 度量版本：_kubectI Get Deploy / kube-state-metases -o jsonpath='{.image}'_若要避免發生這些訊息、使用者可以修改其kube-state-metases 部署、以停用下列Les: _mutatingwebhookwebhookvalidkap_props_enefrouop參數組態：resources=certificatesignquests、水平複製、組態、cronjobs、取消套用、部署、端點、橫向套用自動擴充、擷取、工作、限制範圍、命名空間、網路原則、節點、持續套用磁碟區、持續套用磁碟區、資源資源等、機密、服務、服務、網路套用原則、預設套用範圍、重複本、複本、複製、資源、套用、資源、限制、資源組、資源、資源組態、資源、儲存、預設值、資源、限制、資源、資源、儲存、組態設定、儲存、儲存、儲存、限制、資源、資源、資源、儲存區、限制、資源、資源、資源、資源、儲存區、資源、限制、資源、資源、資源、儲存區、限制、儲存區、資源組態設定、資源、儲存區、資源、資源、儲存區、資源、資源、資源、儲存區、儲存區、資源、資源、資源、資源、 驗證 webhookconfigurations、volume附件"</p>
<p>我看到 Telegraf 的錯誤訊息與下列類似，但 Telegraf 確實啟動並執行： 10 月 11 日 14 : 23 : 41 IP-172-31-39-47 systemd[1] : 啟動外掛程式導向的伺服器代理程式，將計量報告給影響資料庫。10 月 11 日 14 : 23 : 41 IP-172-31-39-47 Telegraf[1827] : Time="2021-10-11T14:23:41Z" Level=error msg="Failed to create cache directory./etc/telegraf/.cache/snowflake , err : mkdir /etc/telegraf/.cache : 權限遭拒。忽略 \n" func="govake.(*defaultLogger) .Errorf " file="log.go:120" Oct11 14 : 23 : 41 IP-172-39-47 Telegraf=18211-20211-20z"已忽略。開啟 /etc/telegraf/.cache/cloflake/occ_restore_cache.json : 無此類檔案或目錄 \n" func="govake.(*defaultLogger) .Errorf" file="log.go:120" 10 月 11 日 14 : 23 : 41 IP-172-39-47 telegraf[1821-2021:2012]啟動Telegraf 1.19.3</p>	<p>這是已知的問題。如"<a href="#">這篇GitHub文章</a>"需詳細資訊、請參閱。只要Telegraf已啟動且正在執行、使用者就可以忽略這些錯誤訊息。</p>
<p>在Kubernetes上、我的Telegraf pod報告下列錯誤：「處理mountstats資訊時發生錯誤：無法開啟mountstats檔案：/hostfs/proc/1/mountstats、錯誤：開啟/hostfs/proc/1/mountstats：權限遭拒」</p>	<p>如果啟用並強制執行 SELinux、則可能會阻止 Telegraf Pod 存取 Kubernetes 節點上的 /proc/1/mountstats 檔案。若要克服此限制、請編輯 agentconfiguration、然後啟用 RunPrivileged 設定。如需詳細資訊、請"<a href="#">OpenShift指示</a>"參閱。</p>



<p>問題：</p> <p>這些 Pod 對應於網路可觀察性的工作負載對應資料收集器。請嘗試下列方法：•檢查其中一個 Pod 的記錄、以確認最低核心版本。例如：--{"CI-租戶-id":"您的租戶-id"、"collector 叢集"："Your -k8s-cluster 名稱"、"Environment"："prod"、"Level"："error"、"msg"：驗證失敗。原因：核心版本 3.10.0 低於最低核心版本 4.18.0、"Time"："2022-11-09T08:23:08Z"} --- •Net 觀察者 Pod 要求 Linux 核心版本至少為 4.18.0。使用命令 "uname -r" 檢查核心版本、並確定它們 &gt;=4.18.0</p>	<p>試用：</p> <p>Pod 在 Operator 命名空間中執行（預設值：NetApp-Monitoring）、但查詢中的工作負載對應或 Kubernetes 度量、UI 中不會顯示任何資料</p>
<p>檢查 K8S 叢集節點上的時間設定。為了準確地進行稽核和資料報告、強烈建議您使用網路時間傳輸協定（NTP）或簡易網路時間傳輸協定（SNTP）、同步代理機器上的時間。</p>	<p>運算子命名空間中的某些網路觀察者 Pod 處於「擱置中」狀態</p>
<p>Net-觀察者 是一組示範集、在 k8s 叢集的每個節點上執行 Pod。•記下處於「擱置中」狀態的 Pod、並檢查它是否發生 CPU 或記憶體資源問題。確保節點中有可用的必要記憶體和 CPU。</p>	<p>安裝 Kubernetes 監控操作員之後、我會立即在記錄中看到下列內容：[外掛程式中出現 inputs.prometheus] 錯誤：向 http://kube-state-metrics 提出 HTTP 要求時發生錯誤。&lt;namespace&gt;。 <a href="http://kube-state-metrics">http://kube-state-metrics</a> svc.cluster。&lt;namespace&gt; 本機：dial tcp: LOOKUP kube-state -indformations.data.svc.cluster &lt;namespace&gt;。本機：無此類主機</p>
<p>此訊息通常只有在安裝新的營運者、且 Telefra-Rs pod 在_ksm_pod 啟動之前就已啟動時才會出現。所有 Pod 都在執行時、這些訊息應該會停止。</p>	<p>我沒有看到叢集中存在的 Kubernetes CronJobs 正在收集任何度量。</p>
<p>驗證 Kubernetes 版本（即 kubectl version）。如果是 v1.2.x 或更低版本、這是預期的限制。Kubernetes Monitoring Operator 部署的 kube-state 度量版本僅支援 v1.cronjob。使用 Kubernetes 1.2.x 及以下版本時、cronjob 資源為 v1beta。cronjob。因此、kube 狀態度量無法找到 cronjob 資源。</p>	<p>安裝操作員之後、Telegraf-DS Pod 會進入 CrashLoopBackOff、Pod 記錄會顯示「su：驗證失敗」。</p>
<p>編輯 AgentConfiguration 中的 Teledraf 區段、並將 dockerMetricCollectionEnabled 設為 false。有關詳細信息、請參閱操作員的"組態選項"。... 規格：... Telegraf：... -名稱：泊塢視窗 執行模式：... -示範設定替代項目：-索引鍵：泊塢視窗 _UNIX 襪子 _placeholder 值：UNIX：//run/泊塢視窗.sock .....</p>	<p>我在 Telegraf 記錄檔中看到類似以下內容的重複錯誤訊息：e ! [代理程式] 寫入 outputs.http：POST「\https://les/rest/v1/lake &lt;tenant_url&gt;/擷取/影響 xdb」時發生錯誤：內容已超過期限（用戶端。等待標頭時超過逾時）</p>
<p>編輯 AgentConfiguration 中的 Telegraf 區段、並將 outputTimeout 增加至 10s。有關詳細信息、請參閱操作員的"組態選項"。</p>	<p>我遺失某些事件記錄的 _參與物件_ 資料。</p>
<p>請務必遵循上述章節中的步驟"權限"。</p>	<p>為什麼我看到兩個監控營運商 Pod 正在執行、一個名為 NetApp-CI-monitoring、&lt;pod&gt;、另一個名為 monitoring、&lt;pod&gt;？</p>
<p>截至 2023 年 10 月 12 日、Data Infrastructure Insights 已重新考慮營運商、以更好地為使用者提供服務；若要充分採用這些變更 <b>移除舊的運算子安裝新的</b>、您必須和。</p>	<p>我的 Kubernetes 事件意外停止回報 Data Infrastructure Insights。</p>

問題：	試用：
擷取事件導出者 Pod 的名稱：	grep event-exporter
<pre>`kubect1 -n netapp-monitoring get pods</pre>	
awk '{print \$1}'	<pre>sed 's/event-exporter./event-exporter/'</pre> <p>應為「NetApp-CI-EVENT - Exporter」或「EVENT - Exporter」。接下來，編輯監控代理程式 <code>kubect1 -n netapp-monitoring edit agent</code>，並設定 <code>log_file</code> 的值，以反映上一步中找到的適當事件導出器 Pod 名稱。更具體地說、<code>log_file</code> 應該設定為「<code>/var/log/container/NetApp-CI-event-Exporter.log</code>」或「<code>/var/log/containers/event-Exporter.log</code>」</p> <pre>.... fluent-bit: ... - name: event-exporter-ci substitutions: - key: LOG_FILE values: - /var/log/containers/netapp-ci-event-exporter*.log ... ....</pre> <p>或者，您也可以<a href="#">解除安裝</a>和<a href="#">重新安裝</a>值機員一起使用。</p>
我看到 Kubernetes Monitoring Operator 部署的 Pod 因為資源不足而當機。	請參閱 Kubernetes Monitoring Operator <a href="#">"組態選項"</a> ，視需要增加 CPU 和 / 或記憶體限制。
遺失影像或組態無效、導致 NetApp-CI-kube 狀態指標 Pod 無法啟動或準備就緒。目前狀況集已停滯、組態變更並未套用至 NetApp-CI-kube 狀態指標群。	StateSetSet 處於某種 <a href="#">"毀損"</a> 狀態。在修正任何組態問題之後、NetApp-CI-kube 狀態指標 Pod 就會恢復運作。
NetApp-CI-kube-state 指標 Pod 在執行 Kubernetes 操作員升級後無法啟動、造成勘誤影像資料（無法擷取影像）。	請嘗試手動重設 Pod。
我的 Kubernetes 叢集在記錄分析下會觀察到「事件已捨棄為較舊、然後是 <code>maxEventAgeSeconds</code> 」的訊息。	修改運算子 <code>agentconfigured</code> 、並增加 <code>event-aler-maxEventAgeSeconds</code> （即增加至 60s）、 <code>event-kaler-kubeQPS</code> （即增加至 100）、以及 <code>_event-kaler-kubebBurst</code> （即增加至 500）。如需這些組態選項的詳細資訊，請參閱 <a href="#">"組態選項"</a> 頁面。
Telegraf 會警告或因為可鎖定的記憶體不足而當機。	嘗試增加基礎作業系統 / 節點中 Telegraf 可鎖定記憶體的限制。如果增加限制並非選項，請修改 NKMO 代理組態，並將 <code>NOISS</code> 設為 <code>true</code> 。這將指示 Telegraf 不要嘗試保留鎖定的記憶體頁面。雖然這可能會造成安全風險、因為解密的機密資料可能會被換出到磁碟、但它允許在無法保留鎖定記憶體的環境中執行。如需 <code>inbizon</code> 組態選項的詳細資訊，請參閱 <a href="#">"組態選項"</a> 頁面。

<p>問題：</p> <p>我看到 Telegraf 發出的警告訊息類似以下內容：_W! [Inputs.DiskIO] 無法收集「VDC」的磁碟名稱：讀取 /dev/VDC 時發生錯誤：沒有此類檔案或目錄_</p>	<p>試用：</p> <p>對於 Kubernetes Monitoring 操作員而言、這些警告訊息是良性的、可以安全地忽略。 或者、在 AgentConfiguration 中編輯 Telegraf 區段、並將 _runDsPrivileged 設為 true 。如需詳細資訊、請<a href="#">"駕駛員的組態選項"</a>參閱。</p>
<p>我的 Fluent 位元 Pod 發生故障，並出現下列錯誤： [2024/10/16 14 : 16 : 23][ 錯誤 ] [/src/flue- bit/plugins/in_tail/fs_inos_inoseture.c:360 errno=24 ] 太多開啟的檔案 [2024/10/16 14 : 16 : 23][ 錯誤 ] 無法初始化輸入 tail.0[2024/16 : 10/16 : 16 : 16 錯 誤 [ 錯誤</p>	<p>嘗試變更叢集中的 _fsnotify_settings :</p> <pre>sudo sysctl fs.inotify.max_user_instances (take note of setting)  sudo sysctl fs.inotify.max_user_instances=&lt;som ething larger than current setting&gt;  sudo sysctl fs.inotify.max_user_watches (take note of setting)  sudo sysctl fs.inotify.max_user_watches=&lt;some thing larger than current setting&gt;</pre> <p>重新啟動 Fluent 位元。</p> <p>附註：若要在重新啟動節點時持續執行這些設定、您需要將下列行放入 /etc/sysctl.conf</p> <pre>fs.inotify.max_user_instances=&lt;so mething larger than current setting&gt; fs.inotify.max_user_watches=&lt;some thing larger than current setting&gt;</pre>

問題：	試用：
由於無法驗證 TLS 憑證，因此 Telegraf DS Pod 回報 Kubernetes 輸入外掛程式無法發出 HTTP 要求的錯誤。例如：e ! 外掛程式中的 [Inputs.Kubernetes] 錯誤：發出取得 " <a "="" &amp;lt;kubelet_ip&amp;gt;="" &amp;lt;kubelet_ip&amp;gt;:10250="" &gt;https:="" :="" a&gt;="" a&gt;：驗證憑證失敗：x509：無法驗證="" class="bare" href="https://&amp;lt;kubelet_IP&amp;gt;:10250/stats/summary" http="" ip="" san<="" stats="" summary":&lt;="" td="" tls="" 的="" 的憑證，因為它不包含任何="" 要求時發生錯誤"&lt;a=""> <td>如果 kubelet 使用自我簽署的憑證，且 / 或指定的憑證未在憑證_主體替代名稱_清單中包含 &lt;kubelet_IP&gt;，就會發生這種情況。若要解決此問題"代理程式組態"，使用者可以修改，並將 <i>telegraf:insecureK8sSkipVerify</i> 設為 <i>true</i>。這會將 Telegraf 輸入外掛程式設定為略過驗證。或者，使用者也可以設定的 kubelet "<i>serverTLSBootstrap</i>"，以觸發來自「certifications.k8s.io」API 的憑證要求。</td> </a>	如果 kubelet 使用自我簽署的憑證，且 / 或指定的憑證未在憑證_主體替代名稱_清單中包含 <kubelet_IP>，就會發生這種情況。若要解決此問題"代理程式組態"，使用者可以修改，並將 <i>telegraf:insecureK8sSkipVerify</i> 設為 <i>true</i> 。這會將 Telegraf 輸入外掛程式設定為略過驗證。或者，使用者也可以設定的 kubelet " <i>serverTLSBootstrap</i> "，以觸發來自「certifications.k8s.io」API 的憑證要求。

您可以在頁面或中找到其他"支援"資料收集器支援對照表"資訊。

## Kubernetes 監控操作員組態選項

"Kubernetes 監控營運者"您可以自訂組態。

下表列出 *AgentConfiguration* 檔案的可能選項：

元件	選項	說明
代理程式		操作員可以安裝的所有元件通用的組態選項。這些選項可視為「整體」選項。
	dockerRepo	相較於 Data Infrastructure Insights 泊塢視窗 repo、dockerRepo 會置換以從「客戶」的「私有」泊塢視窗中提取影像。預設為 Data Infrastructure Insights 泊塢視窗 repo
	dockerImagePullSecret	選用：客戶的秘密私人回購
	叢集名稱	可唯一識別所有客戶叢集的任意文字欄位。這在 Data Infrastructure Insights 租戶中應該是獨一無二的。預設是客戶在 UI 中輸入的「叢集名稱」欄位
	Proxy 格式：Proxy：伺服器：連接埠：使用者名稱：密碼：NoProxy：啟用 ISTELEgrafProxy: 啟用 isAuProxy: 啟用 isFluentbitProxy: 啟用 isCollectorProxy: 啟用 isCollectorProxy:	可選擇設定 Proxy。這通常是客戶的公司代理。
Telegraf		可自訂電信業者安裝的組態選項
	CollectionInterval	度量收集時間間隔（以秒為單位）（最大 = 60 秒）
	dsCpuLimit	Telegraf DS 的 CPU 限制
	dsMemLimit	Telegraf DS 的記憶體限制
	dsCpuRequest	對 Telegraf DS 的 CPU 要求

元件	選項	說明
	dsMemRequest	對 Telegraf DS 的記憶體要求
	rsCpuLimit	Telegraf RS 的 CPU 限制
	rsMemLimit	Telegraf RS 的記憶體限制
	rsCpuRequest	適用於 Telegraf RS 的 CPU 要求
	rsMemRequest	對 Telegraf RS 的記憶體要求
	RunPrivileged	在特權模式下運行 Telegraf demonSet 的 _telegraf-mountstats-poller-Container。如果在 Kubernetes 節點上啟用 SELinux、請將此設定為 True。
	RunDsPrivileged	將 RunDsPrivileged 設為 true、以特權模式執行 Telegraf demonSet 的 Telgraf 容器。
	批次大小	請參閱 <a href="#">"Telegraf 組態文件"</a>
	bufferLimit	請參閱 <a href="#">"Telegraf 組態文件"</a>
	圓週期間隔	請參閱 <a href="#">"Telegraf 組態文件"</a>
	CollectionJitter	請參閱 <a href="#">"Telegraf 組態文件"</a>
	精度	請參閱 <a href="#">"Telegraf 組態文件"</a>
	FlushInterval	請參閱 <a href="#">"Telegraf 組態文件"</a>
	FlushJitter	請參閱 <a href="#">"Telegraf 組態文件"</a>
	輸出逾時	請參閱 <a href="#">"Telegraf 組態文件"</a>
	dsTolerations	Telegraf-DS 額外的容忍度。
	RsTolerations	Telegraf-RS 額外容忍度。
	skipProcessorsAfterAggregators	請參閱 <a href="#">"Telegraf 組態文件"</a>
	未受保護	EE 本 <a href="#">"已知的 Telegraf 問題"</a> 。設定 <code>_無保護_</code> 會指示 Kubernetes 監控營運者以旗標執行 Telegraf。 <code>--unprotected</code>
	insecureK8sSkipVerify	如果由於缺少 IP SAN 而導致 Telegraf 無法驗證憑證，請嘗試啟用驗證跳過
Kube-state 指標		可自訂操作員的 kbe 狀態度量安裝的組態選項
	cpuLimit	kube 狀態度量部署的 CPU 限制
	MemLimit	kube 狀態度量部署的記憶體限制
	cpuRequest	CPU 要求進行 kube 狀態指標部署
	MemRequest	kube 狀態指標部署的記憶體要求
	資源	要擷取的資源清單以逗號分隔。例如：cronjobs，daemonsets，部署，擷取，工作，命名空間，節點，持續磁碟區，持續磁碟區，群組，複本集，資源等量，服務，完整狀態集
	公差	Kube-state - 衡量其他容忍度。



元件	選項	說明
	標籤	以逗號分隔的資源清單應擷取 + 範例： cronjobs=[],daemonsets=[],plasing=[],inges=[],jobs=[ ],vedemas=[],nodes=[],sterentvolumes=[,equpos=[, equas*[*
記錄		可自訂記錄收集和安裝操作員的組態選項
	readFromHead	是非題、應能流暢地從標頭讀取記錄
	逾時	逾時、以秒為單位
	dnsMode	TCP/UDP、DNS 模式
	流暢的位元容忍度	Fluent-bit-DS 額外公差。
	事件導出者容忍度	事件導出者額外容忍度。
	事件導出者 -maxEventAgeSeconds	事件導出者最大事件發生時間。請參閱 <a href="https://github.com/jkroepke/resmoio-kubernetes-event-exporter">https://github.com/jkroepke/resmoio-kubernetes-event-exporter</a>
工作負載對應		可自訂工作負載對應集合及安裝 Operator 的組態選項。
	cpuLimit	Net 觀察者 DS 的 CPU 上限
	MemLimit	net 觀察者 DS 的記憶體限制
	cpuRequest	CPU 要求取得 Net 觀察者 DS
	MemRequest	net 觀察者 DS 的記憶體要求
	MetricAggregationInterval.	度量集合時間間隔 (以秒為單位)
	bpfPollInterval.	BPF 輪詢時間間隔 (秒)
	enabledDNSlookup	是非題、啟用 DNS 查詢
	L4-公差	net-觀察者 -L4-DS 額外容忍度。
	RunPrivileged	是非題：如果在 Kubernetes 節點上啟用 SELinux、請將 RunPrivileged 設為 true。
變更管理		Kubernetes 變更管理與分析的組態選項
	cpuLimit	change-觀察者 water-RS 的 CPU 上限
	MemLimit	change-觀察者 water-RS 的記憶體限制
	cpuRequest	CPU 要求變更觀察者手錶 -RS
	MemRequest	mem 要求 change-觀察者 water-RS
	Failure宣言 InternalMins	未成功部署工作負載的時間間隔 (以分鐘為單位) 將標示為失敗
	deployAggrIntervalSeconds	工作負載部署進行中事件的傳送頻率
	NonWorkloadAggrIntervalSeconds	非工作負載部署的組合與傳送頻率

元件	選項	說明
	termsToRedact	env 名稱和資料對應中使用的一組規則運算式，其值將會被編修範例詞彙： "pwd" ， "password" ， "toke" ， "apikey" ， "api-key" ， "JWT"
	其他 KindsToWatch	以逗號分隔的其他種類清單、可從收集器所監控的預設種類集觀看
	KindsToIgnoreFromWatch	從收集器所監控的預設種類集中、忽略的種類清單、以逗號分隔
	LogRecordAggrIntervalSecs	從收集器傳送記錄至 CI 的頻率
	監看容忍度	change-觀察者 water-DS 額外容忍度。僅限精簡單行格式。範例： ' { key : tint1 、 operator : Exists 、 effect : NoSchedule } 、 { key : tint2 、 operator : Exists 、 effect : NoExecute } '

## AgentConfiguration 檔案範例

以下是 *AgentConfiguration* 檔案範例。

```

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,persistentvolumeclaims,persistentvolumes,pods,replicasets,resourcequotas,services,storageclasses'

# # 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,kub

```

```
e_pod_container_status_restarts_total,kube_pod_overhead_cpu_cores,kube_pod_overhead_memory_bytes,kube_pod_created,kube_pod_deletion_timestamp,kube_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_cores,kube_pod_container_resource_limits_memory_bytes,kube_pod_container_resource_limits_storage_bytes,kube_pod_container_resource_limits_ephemeral_storage_bytes,kube_pod_init_container_resource_limits_cpu_cores,kube_pod_init_container_resource_limits_memory_bytes,kube_pod_init_container_resource_limits_storage_bytes,kube_pod_init_container_resource_limits_ephemeral_storage_bytes,kube_pod_init_container_resource_requests_cpu_cores,kube_pod_init_container_resource_requests_memory_bytes,kube_pod_init_container_resource_requests_storage_bytes,kube_pod_init_container_resource_requests_ephemeral_storage_bytes,kube_replicaset_status_replicas,kube_replicaset_status_ready_replicas,kube_replicaset_status_observed_generation,kube_replicaset_spec_replicas,kube_replicaset_metadata_generation,kube_replicaset_labels,kube_replicaset_created,kube_replicaset_owner,kube_resourcequota,kube_resourcequota_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_statefulset_status_replicas_updated,kube_statefulset_status_observed_generation,kube_statefulset_replicas,kube_statefulset_metadata_generation,kube_statefulset_created,kube_statefulset_labels,kube_statefulset_status_current_revision,kube_statefulset_status_update_revision,kube_node_status_capacity,kube_node_status_allocatable,kube_node_status_condition,kube_pod_container_resource_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=[*],namespaces=[*],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: ''

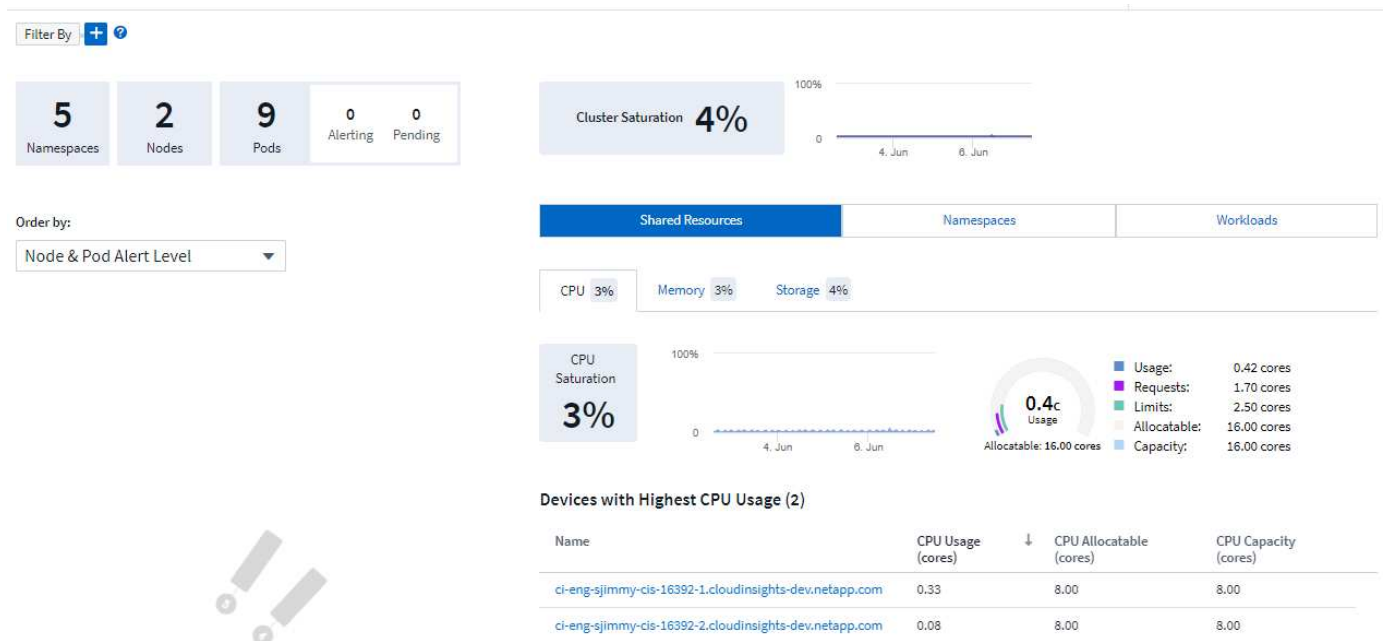
# # Frequency with which log records are sent to CI from the collector
# logRecordAggrIntervalSeconds: '20'

# # 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 叢集詳細資料頁面

Kubernetes 叢集詳細資料頁面會顯示 Kubernetes 叢集的詳細總覽。



### 命名空間、節點和Pod數

頁面頂端的計數會顯示叢集中的命名空間、節點和Pod總數、以及目前警示和擱置中的快顯數目。

### 共享資源與配置

在詳細資料頁面右上角、您的叢集飽和程度為目前百分比、以及顯示一段時間內最近趨勢的圖表。叢集飽和是指每個時間點的CPU、記憶體或儲存設備飽和程度最高。

下方的頁面預設會顯示\*共享資源\*使用量、其中有CPU、記憶體和儲存設備的索引標籤。每個索引標籤都會顯示

一段時間內的飽和百分比和趨勢、以及其他使用詳細資料。對於儲存設備而言、所顯示的值是後端和檔案系統飽和程度的較高值、而這是獨立計算出來的值。

使用率最高的裝置會顯示在底部的表格中。按一下任一連結即可瀏覽這些裝置。

## 命名空間

「命名空間」索引標籤會顯示Kubernetes環境中所有命名空間的清單、顯示CPU和記憶體使用量、以及每個命名空間中的工作負載計數。按一下「Name (名稱)」連結以瀏覽每個命名空間。

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

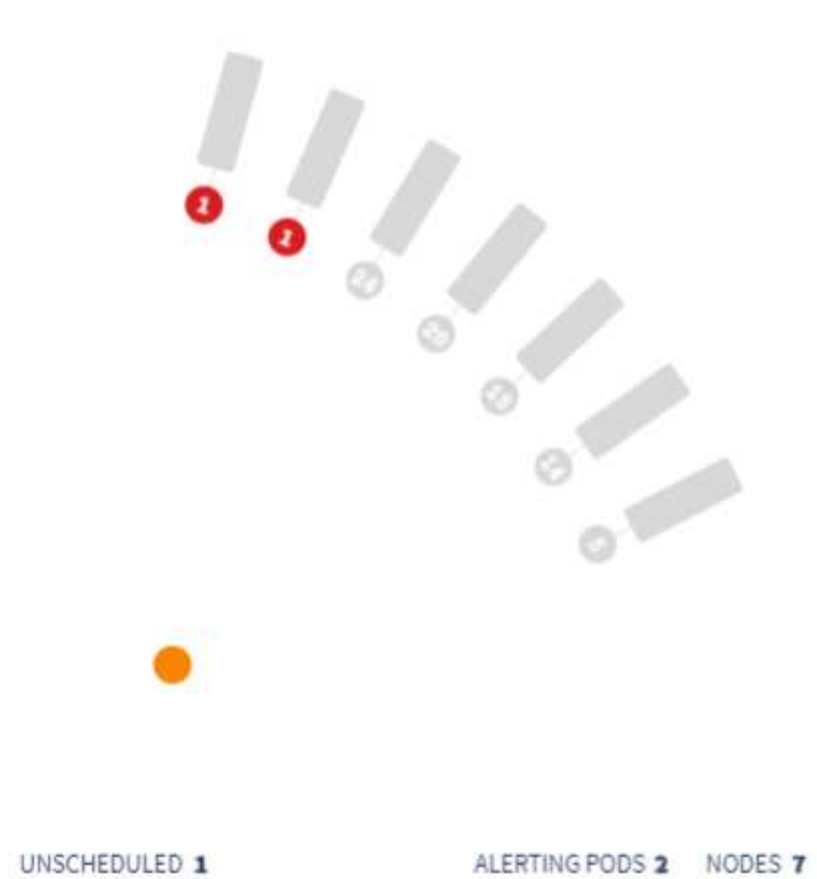
## 工作負載

同樣地、「工作負載」索引標籤會顯示每個命名空間中的工作負載清單、再次顯示CPU和記憶體使用量。按一下「命名空間」連結、即可深入瞭解每個項目。

### 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

### 叢集「輪式」



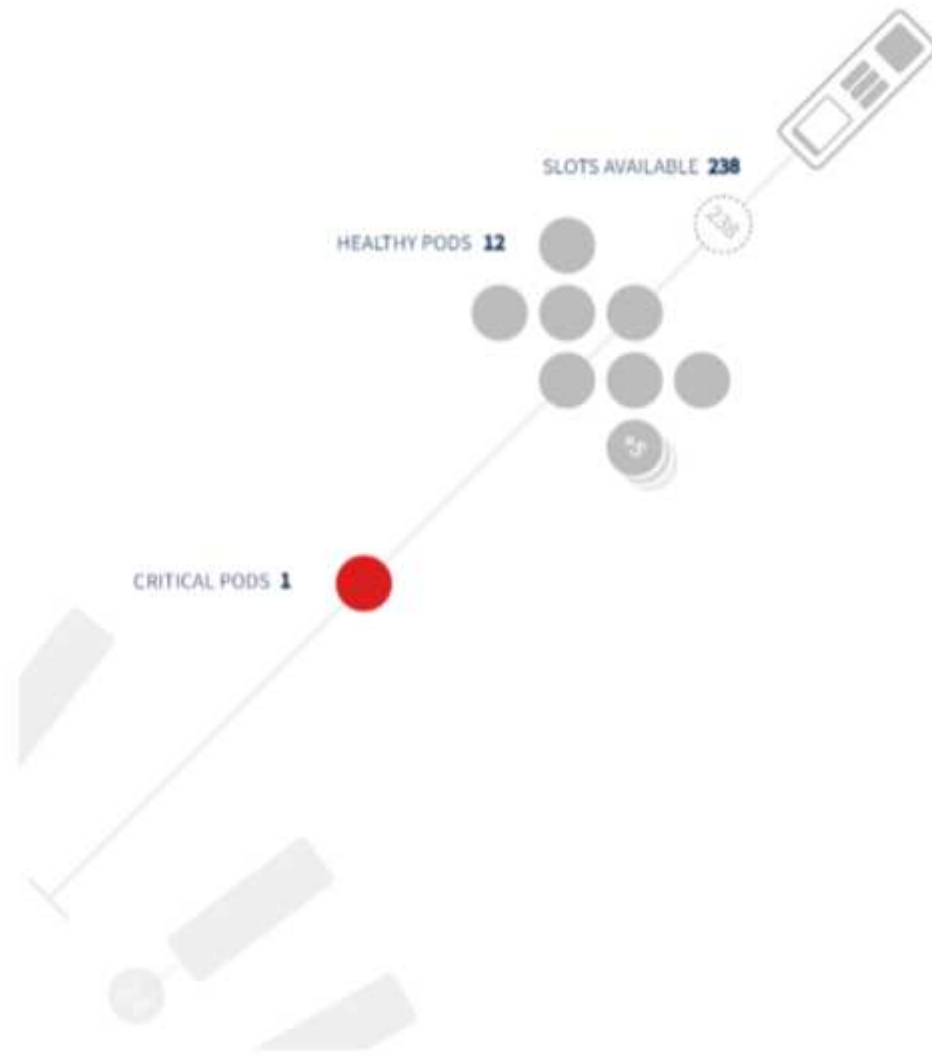
叢集「輪式」區段提供節點和pod健全狀況一覽、您可以深入瞭解更多資訊。如果叢集包含的節點數量超過頁面此區域所能顯示的數量、您就能使用可用的按鈕來轉動定位輪。

警示Pod或節點會以紅色顯示。「警告」區域會以橘色顯示。非排程的Pod（即未附加的）會顯示在叢集「輪式」的下角。

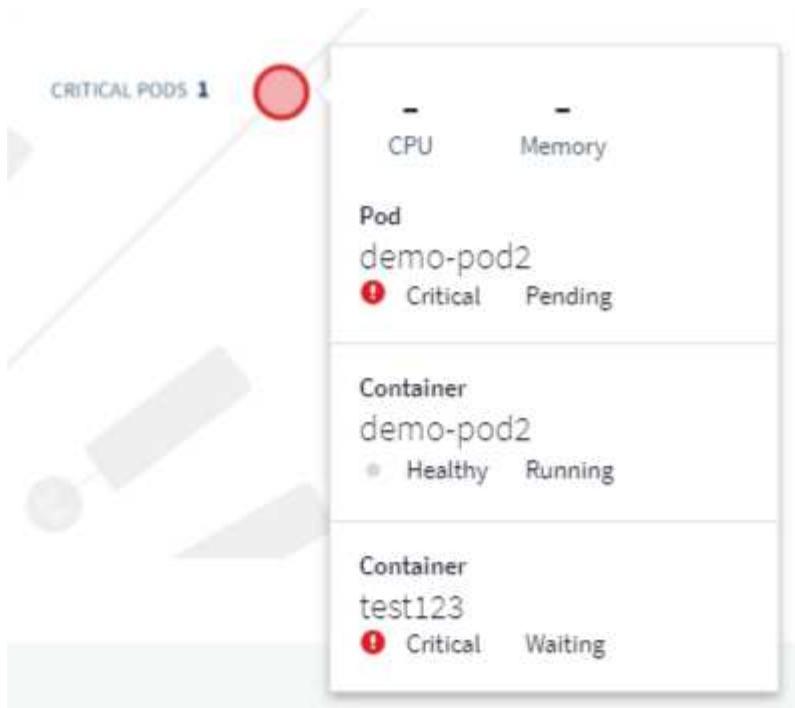
將游標移到Pod（圓圈）或節點（長條）上、將會延伸節點的檢視範圍。



按一下該檢視中的Pod或節點、即可放大展開的節點檢視。



您可在此處將游標暫留在某個元素上、以顯示該元素的詳細資料。例如、將游標移到關鍵Pod上、就會顯示該Pod的詳細資料。



您可以將游標移到Node元素上、以檢視Filesystem、Memory和CPU資訊。



### 量表注意事項

記憶體和CPU量測計顯示三種顏色、因為它們顯示的是與\_allocatable capacity\_和\_total capacity\_相關的\_二者\_。


## Kubernetes 網路效能監控與地圖

Kubernetes 網路效能監控與地圖功能可對應服務之間的相依性（也稱為工作負載）、藉此簡化疑難排解程序、並即時掌握網路效能延遲和異常狀況、以便在影響使用者之前先識別效能問題。此功能可分析及稽核 Kubernetes 流量、協助組織降低整體成本。

主要功能：  
 • 工作負載對應表呈現 Kubernetes 工作負載相依性和流程、並強調網路和效能問題。  
 • 監控 Kubernetes Pod、工作負載和節點之間的網路流量、找出流量和延遲問題的來源。  
 • 分析入口、出口、跨區域和跨區域網路流量、藉此降低整體成本。

## 先決條件

在使用 Kubernetes 網路效能監控和地圖之前，您必須先設定"[NetApp Kubernetes監控操作員](#)"以啟用此選項。在部署操作員期間、選取「網路效能與地圖」核取方塊以啟用。您也可以瀏覽至 Kubernetes 登陸頁面、然後選取「修改部署」來啟用此選項。

 **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](#)



## 監控

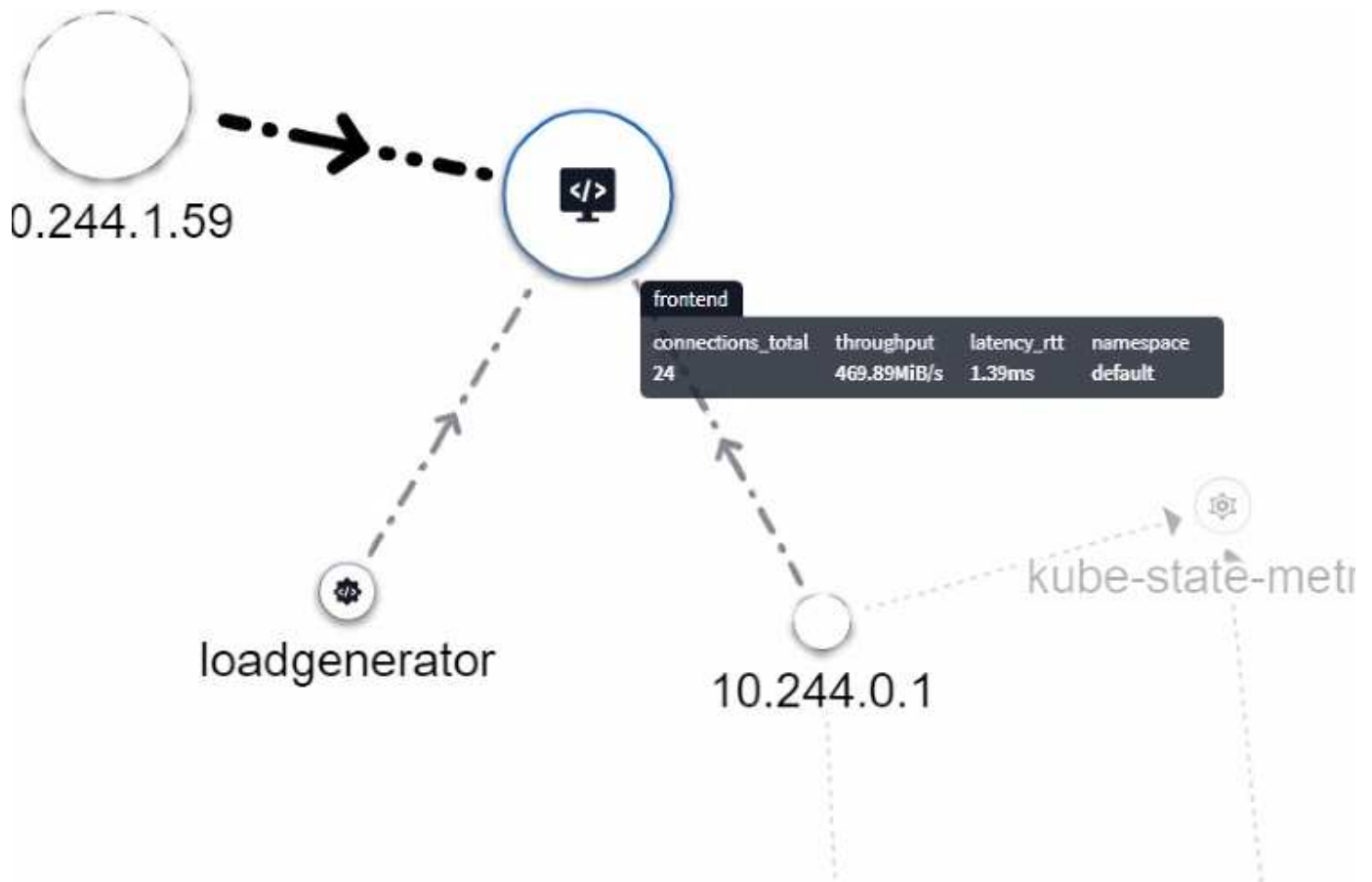
工作負載對應會使用"監控"來衍生資訊。Data Infrastructure Insights 提供許多預設的 Kubernetes 監視器（請注意、這些監視器預設可能為 `_Paused`（暫停））。您可以 `_恢復_`（即啟用）您想要的監視器）、或是為 Kubernetes 物件建立自訂監視器、工作負載對應也會使用這些監視器。

您可以針對下列任何物件類型建立 Data Infrastructure Insights 度量警告。請確定資料已依預設物件類型分組。

- Kubernetes.Workload
- Kubernetes.daemonset
- kubernetes.deployment
- Kubernetes.cronjob
- Kubernetes.job
- Kubernetes.Replicaset
- Kubernetes.statefset
- Kubernetes.pod
- kubernetes.network\_traffic\_l4

## 地圖

地圖會顯示服務 / 工作負載及其彼此之間的關係。箭頭顯示交通路況方向。將游標移至工作負載上方會顯示該工作負載的摘要資訊、如以下範例所示：

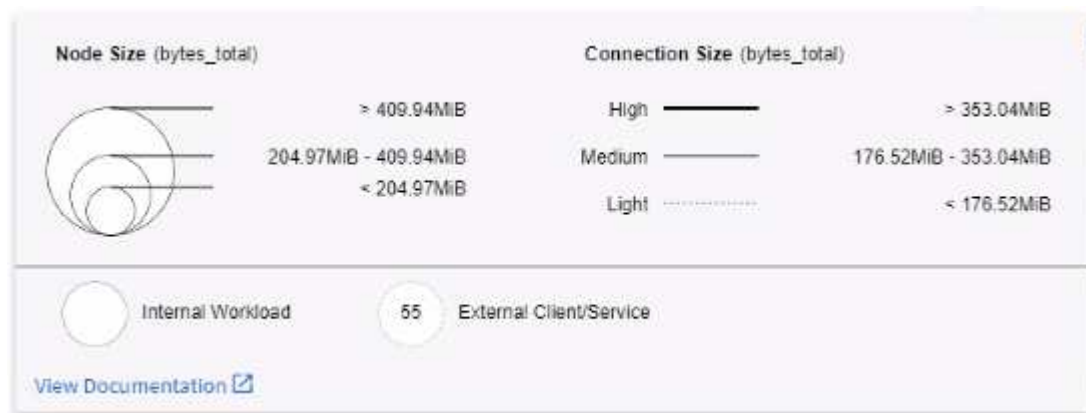


圓圈內的圖示代表不同的服務類型。請注意，只有在基礎物件具有標籤時，才會顯示圖示。



每個圓圈的大小代表節點大小。請注意、這些大小是相對的、您的瀏覽器縮放等級或螢幕大小可能會影響實際的圓圈大小。同樣地、交通路況線條樣式也能讓您一目瞭然地瞭解連線大小、粗體實線是高流量、而輕點虛線則是較低的流量。

圓圈內的數字是服務目前正在處理的外部連線數量。



## 工作負載詳細資料和警示

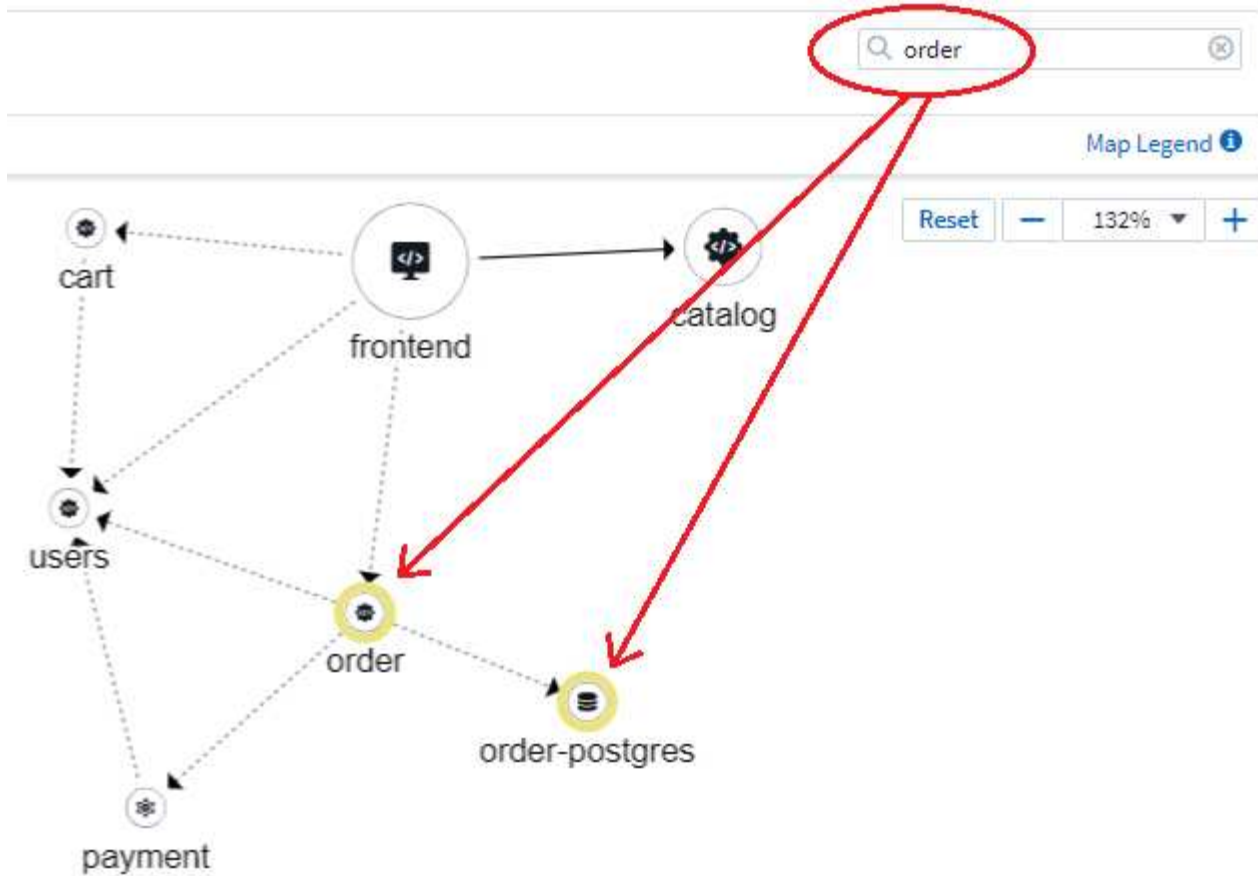
以色彩顯示的圓圈表示工作負載的警示或嚴重層級警示。將游標移到圓圈上方以取得問題摘要、或按一下圓圈以開啟詳細的滑出面板。

## 尋找及篩選

如同其他 Data Infrastructure Insights 功能、您可以輕鬆設定篩選條件、將重點放在您想要的特定物件或工作負載屬性上。

### AQA / Workload Map

同樣地、在 *Find* 欄位中鍵入字串、也會醒目提示相符的工作負載。



## 工作負載標籤

如果您希望地圖識別所顯示的工作負載類型（例如圓圈圖示）、則工作負載標籤是必要的。標籤的衍生方式如下：

- 以一般術語執行的服務 / 應用程式名稱
- 如果來源是 Pod：
  - 標籤衍生自 Pod 的工作負載標籤
  - 工作負載上的預期標籤：app.Kubernetes.IO/ 元件
  - 標籤名稱參照：<https://kubernetes.io/docs/concepts/overview/working-with-objects/common-labels/>
  - 建議標籤：
    - 前端
    - 後端
    - 資料庫
    - 快取
    - 佇列
    - 卡夫卡
- 如果來源位於 Kubernetes 叢集外部：
  - Data Infrastructure Insights 會嘗試剖析 DNS 解析名稱、以擷取服務類型。

例如、使用 DNS 解析名稱 `s3.eu-north-1.amazonaws.com`、解析解析名稱以取得 `_S_` 作為服務類型。

## 深入探索

在工作負載上按一下滑鼠右鍵、即可提供其他選項、讓您進一步探索。例如、您可以從這裡放大檢視該工作負載的連線。



或者、您也可以開啟「詳細資料」滑出面板、直接檢視「Summary」、「Network」或「Pod & Storage」標籤。

**frontend** Last 3 Hours ✕

[Go to Asset Page](#)

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

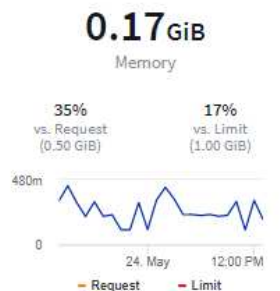
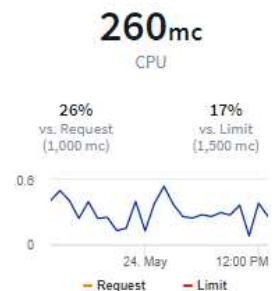
最後、選取 `_移至資產頁面_` 將會開啟工作負載的詳細資產登陸頁面。

Filter By + ?

**2/2**  
Pods: Current / Desired

2 Up-to-date    0 Unavailable

Namespace <b>netapp-fitness-store-01</b>	Type <b>Deployment</b>	Date Created <b>Apr 11, 2023 11:34 AM</b>
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

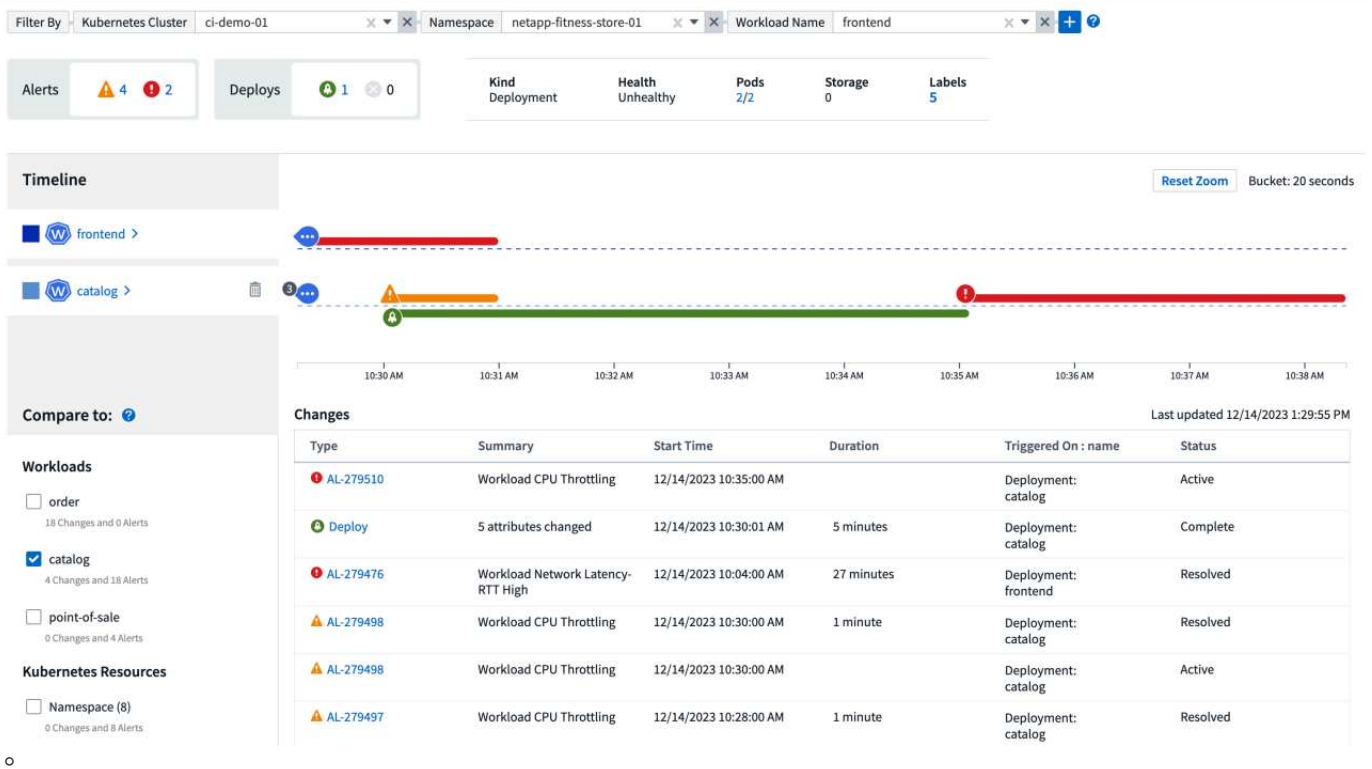
## Kubernetes 變更分析

Kubernetes Change Analytics 提供您 K8s 環境最近變更的全方位檢視。警示和部署狀態盡在您的掌握之中。利用 Change Analytics、您可以追蹤每個部署和組態變更、並將其與 K8s 服務、基礎架構和叢集的健全狀況和效能建立關聯。

變更分析有何幫助？

- 在多租戶 Kubernetes 環境中、可能會因為設定不當而導致停機。變更分析可提供單一窗格來檢視及關聯工作負載和組態變更的健全狀況、藉此協助解決此問題。這有助於疑難排解動態 Kubernetes 環境。

若要檢視 Kubernetes 變更分析、請瀏覽至 [\\* Kubernetes > 變更分析 \\*](#)。

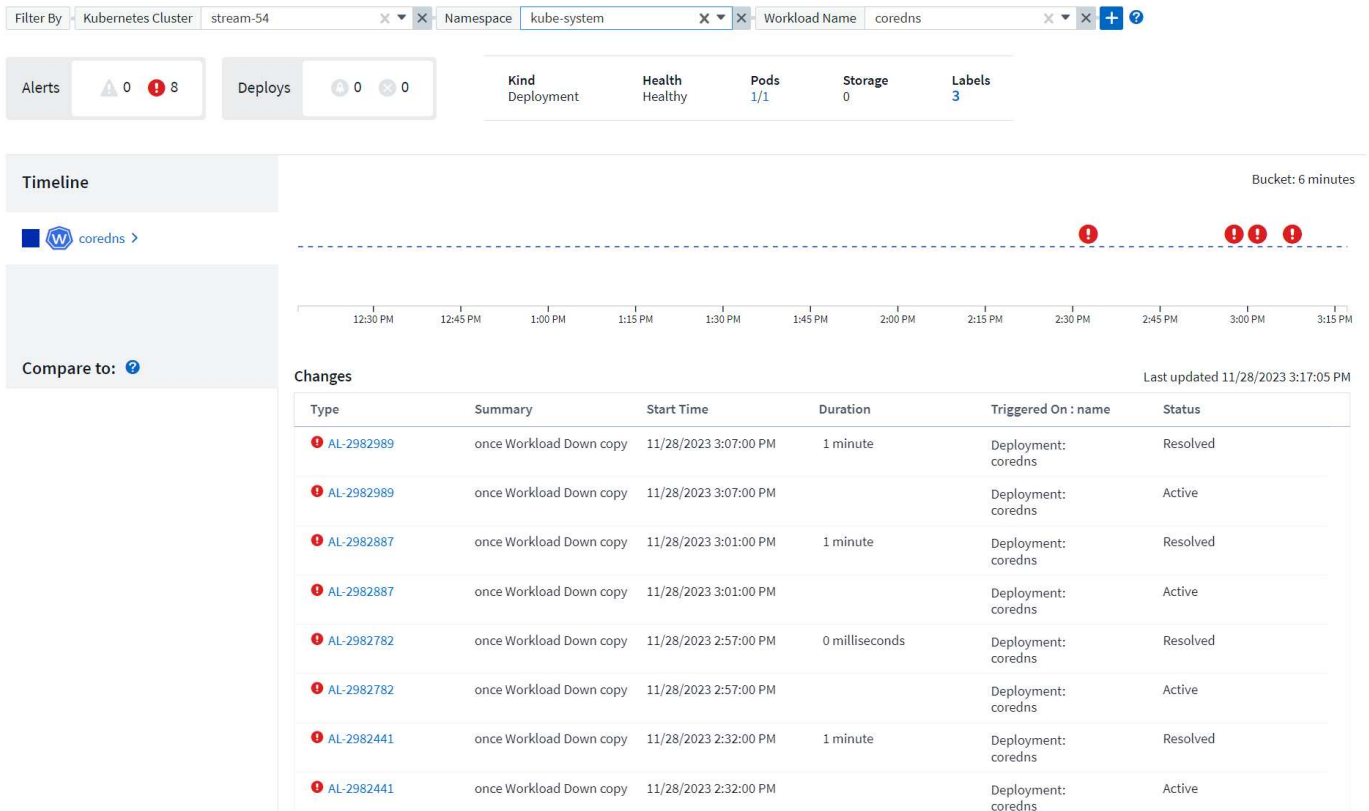


頁面會根據目前選取的 Data Infrastructure Insights 時間範圍自動重新整理。時間範圍越短、畫面更新就越頻繁。

## 篩選

與 Data Infrastructure Insights 的所有功能一樣、篩選變更清單是直覺式的：在頁面頂端、輸入或選取 Kubernetes 叢集、命名空間或工作負載的值、或是選取 { + } 按鈕來新增自己的篩選器。

當您篩選到特定叢集、命名空間和工作負載（以及您設定的任何其他篩選器）時、會在該叢集的該命名空間中顯示該工作負載的部署和警示時間表。在圖表中按一下並拖曳、即可進一步放大、以專注於更特定的時間範圍。



## 快速狀態

篩選區域下方有許多高階指標。左側是警示數量（警告和嚴重）。這個數字包括 *Active* 和 *Resolved* 警示。若要僅查看 *\_Active* 警示、請設定「狀態」篩選條件、然後選擇「作用中」。

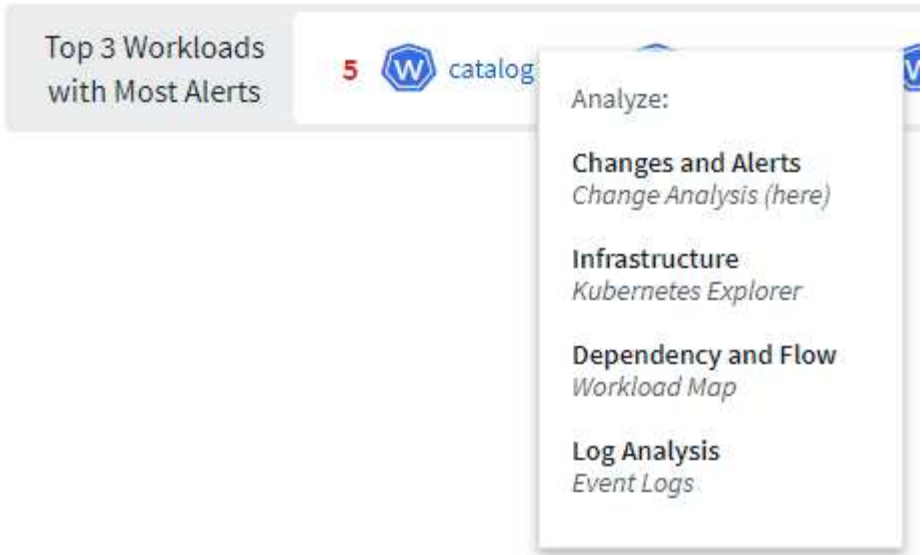


此處也會顯示部署狀態。同樣地、預設值是顯示 *started*、*Completed* 和 *Failed* 部署的計數。若要僅查看 *Failed* 部署、請設定「狀態」篩選條件、然後選取「失敗」。



接下來是警示最多的前 3 大工作負載。每個工作負載旁的紅色數字代表與該工作負載相關的警示數量。按一下工作負載連結、瀏覽基礎架構（Kubernetes Explorer）、相依性（工作負載對應）或記錄分析（事件記錄）。





### 詳細資料面板

在清單中選取變更會開啟一個面板、以更詳細地說明變更。例如、選取失敗的部署會顯示部署摘要、包括開始和結束時間、持續時間、以及觸發部署的位置、以及探索這些資源的連結。它也會顯示故障原因、任何相關變更、以及任何相關事件。

## 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

同樣地、選取警示也會提供警示的詳細資料、包括觸發警示的監視器、以及顯示警示視覺時間表的圖表。

## 版權資訊

Copyright © 2025 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

## 商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。