



使用**Astra**資料儲存區

Astra Data Store

NetApp
July 19, 2022

目錄

使用Astra資料儲存區	1
使用kveccli命令管理Astra Data Store預覽資產	1
部署測試應用程式	7
管理叢集	11
利用功能表監控指標Cloud Insights	23
使用Prometheus和Grafana監控指標	35
設定及監控事件記錄	37
使用Astra Control Center搭配Astra Data Store預覽	37
使用自動指令碼解除安裝Astra Data Store預覽	38
無需指令碼即可解除安裝Astra Data Store預覽	44

使用Astra資料儲存區

使用kvecll命令管理Astra Data Store預覽資產

您可以使用kubectll命令和Kubernetes API擴充功能來管理Astra Data Store預覽資產。

若要瞭解如何部署範例應用程式、請參閱 ["部署測試應用程式"](#)。

如需下列叢集維護資訊、請參閱 ["管理叢集"](#)：

- 將節點置於維護模式
- 更換磁碟機
- 新增節點
- 更換節點

您需要的是 **#8217** ；需要的是什麼

- 您安裝的Astra Data Store預覽KECBECVL外掛程式 ["安裝Astra Data Store預覽"](#)

列出用於Astra Data Store預覽的Kubernetes自訂API資源

您可以使用Kubernetes內部的kubectll命令、與Astra Data Store預覽叢集互動並觀察其狀態。

「API-resources」命令中列出的每個項目都代表Kubernetes自訂資源定義（CRD）、Astra Data Store預覽可在內部用於管理叢集。

此清單特別有助於取得每個Astra Data Store預覽物件的快取名稱、以減少您的輸入、如稍後所示。

1. 顯示用於Astra Data Store預覽的Kubernetes自訂API資源清單：

```
kubectll api-resources --api-group astrads.netapp.io
```

回應：

NAME	SHORTNAMES	APIGROUP	NAMESPACED	KIND
astradsversions	adsve	astrads.netapp.io	true	
AstraDSVersion				
astradsclusters	adscl	astrads.netapp.io	true	
AstraDSCluster				
astradslicenses	adsli	astrads.netapp.io	true	
AstraDSLICENSE				
astradsnodeinfoes	adsni	astrads.netapp.io	true	
AstraDSNodeInfo				
astradsvolumes	adsvo	astrads.netapp.io	true	
AstraDSVolume				
astradsqospolicies	adsqp	astrads.netapp.io	true	
AstraDSQosPolicy				
astradsexportpolicies	adsep	astrads.netapp.io	true	
AstraDSEExportPolicy				
astradsvolumesnapshots	adsvs	astrads.netapp.io	true	
AstraDSVolumeSnapshot				
astradsvolumefiles	adsvf	astrads.netapp.io	true	
AstraDSVolumeFiles				
astradsautosupports	adsas	astrads.netapp.io	true	
AstraDSAutoSupport				
astradsfaileddrives	adsfd	astrads.netapp.io	true	
AstraDSFailedDrive				
astradsnodemanagements	adsnm	astrads.netapp.io	true	
AstraDSNodeManagement				

2. 若要在Kubernetes叢集中取得所有目前的Astra Data Store預覽物件、請使用「kubectl Get ads -As」命令：

```
kubectl get ads -A
```

回應：

```

NAMESPACE      NAME                                                    AGE
astrads-system  astradsqospolicy.astrads.netapp.io/bronze             45h
astrads-system  astradsqospolicy.astrads.netapp.io/gold               45h
astrads-system  astradsqospolicy.astrads.netapp.io/silver             45h

NAMESPACE      NAME
STATUS    VERSION      SERIAL NUMBER    MVIP                AGE
astrads-system  astradscluster.astrads.netapp.io/astrads-cluster-9f1
created      arda-9.11.1   e000000009       10.224.8.146        46h

NAMESPACE      NAME
```

```

AGE
astrads-system    astradsnodeinfo.astrads.netapp.io/englab.netapp.com
46h
astrads-system    astradsnodeinfo.astrads.netapp.io/englab.netapp.com
46h
astrads-system    astradsnodeinfo.astrads.netapp.io/englab.netapp.com
46h
astrads-system    astradsnodeinfo.astrads.netapp.io/englab.netapp.com
46h

NAMESPACE          NAME                                     AGE
astrads-system    astradsversion.astrads.netapp.io/astradsversion  46h

NAMESPACE          NAME                                     AGE
astrads-system    astradsvolumefiles.astrads.netapp.io/test23      27h
astrads-system    astradsvolumefiles.astrads.netapp.io/test234     27h
astrads-system    astradsvolumefiles.astrads.netapp.io/test2345    4h22m

NAMESPACE          NAME                                     SIZE   IP
CLUSTER            CREATED
astrads-system    astradsvolume.astrads.netapp.io/test234         21Gi
172.25.123.123    astrads-cluster-9f1 true
astrads-system    astradsvolume.astrads.netapp.io/test2345         21Gi
172.25.123.123    astrads-cluster-9f1 true

NAMESPACE          NAME
SEQUENCE COMPONENT      EVENT          TRIGGER    PRIORITY  SIZE
STATE
astrads-system    astradsautosupport.astrads.netapp.io/controlplane-
adsclustercreatesuccess-20211214t 9          controlplane
adsclustercreatesuccess k8sEvent  notice    0          uploaded
astrads-system    astradsautosupport.astrads.netapp.io/controlplane-
daily-20211215t0          15          controlplane  daily
periodic  notice    0          uploaded
astrads-system    astradsautosupport.astrads.netapp.io/controlplane-
daily-20211216t0          20          controlplane  daily
periodic  notice    0          uploaded
astrads-system    astradsautosupport.astrads.netapp.io/storage-
callhome.dbs.cluster.cannot.sync.blocks 10          storage
callhome.dbs.cluster.cannot.sync.blocks  firetapEvent  emergency  0
uploaded

NAMESPACE          NAME                                     ADSCLUSTER
VALID PRODUCT          EVALUATION ENDDATE    VALIDATED
astrads-system    astradslicense.astrads.netapp.io/e0    astrads-cluster-
9f1 true  Astra Data Store true          2022-02-07 2021-12-16T20:43:23Z

```

3. 使用其中一個簡短名稱來顯示叢集中磁碟區的目前狀態：

```
kubectl get adsvo -A
```

回應：

NAMESPACE	NAME	SIZE	IP	CLUSTER
astrads-system	test234	21Gi	172.25.138.109	astrads-cluster-9f1c99f
astrads-system	test2345	21Gi	172.25.138.111	astrads-cluster-9f1c99f

使用KECBECVL副檔名上的說明選項

「kubectl astrads」命令包含一個「-h」交換器、可提供使用方法和標記文件、方便您使用。

1. 顯示Astra Data Store preview kubectl副檔名中所有命令的說明：

```
kubectl astrads -h
```

回應：

```
A kubectl plugin for inspecting your AstraDS deployment

Usage:
  astrads [command]

Available Commands:
  asup          Manage AutoSupport
  clusters      Manage clusters
  drives        Manage drives in a cluster
  faileddrive   Manage drive replacement in a cluster
  help          Help about any command
  license       Manage license in the astrads cluster
  maintenance   Manage maintenance status of a node
  monitoring    Manage Monitoring Output
  nodes         Manage nodes in a cluster

Flags:
  --as string                Username to impersonate for the
operation
  --as-group stringArray     Group to impersonate for the
```

operation, this flag can be

groups.

`--cache-dir string`

cache")

`--certificate-authority string`
certificate authority

`--client-certificate string`
for TLS

`--client-key string`

`--cluster string`

cluster to use

`--context string`

context to use

`-h, --help`

`--insecure-skip-tls-verify`
certificate will not be checked

your HTTPS connections insecure

`--kubeconfig string`
use for CLI requests.

`-n, --namespace string`
for this CLI request

`--request-timeout string`
before giving up on a single

should contain a

1s, 2m, 3h).

timeout requests.

`-s, --server string`
Kubernetes API server

`--token string`
to the API server

`--user string`
to use

repeated to specify multiple

Default HTTP cache directory
(default "/u/arda/.kube/http-

Path to a cert file for the

Path to a client certificate file

Path to a client key file for TLS

The name of the kubeconfig

The name of the kubeconfig

help for astrads

If true, the server's

for validity. This will make

Path to the kubeconfig file to

If present, the namespace scope

The length of time to wait

server request. Non-zero values

corresponding time unit (e.g.

A value of zero means don't

(default "0")

The address and port of the

Bearer token for authentication

The name of the kubeconfig user

2. 如需命令的詳細資訊、請使用「astrads [command]-help」。

```
kubectl astrads asup collect --help
```

回應：

Collect the autosupport bundle by specifying the component to collect. It will default to manual event.

Usage:

```
astrads asup collect [flags]
```

Examples:

```
# Control plane collection
```

```
kubectl astrads collect --component controlplane example1
```

```
# Storage collection for single node
```

```
kubectl astrads collect --component storage --nodes node1 example2
```

```
# Storage collection for all nodes
```

```
kubectl astrads collect --component storage --nodes all example3
```

```
# Collect but don't upload to support
```

```
kubectl astrads collect --component controlplane --local example4
```

NOTE:

```
--component storage and --nodes <name> are mutually inclusive.
```

```
--component controlplane and --nodes <name> are mutually exclusive.
```

Flags:

<pre>-c, --component string</pre>	Specify the component to collect: [storage , controlplane , vasaprovider, all]
<pre>-d, --duration int</pre>	Duration is the duration in hours from the startTime for collection
<pre>-e, --event string</pre>	of AutoSupport. This should be a positive integer
<pre>(default "manual")</pre>	Specify the callhome event to trigger.
<pre>-f, --forceUpload</pre>	Configure an AutoSupport to upload if
<pre>it is in the compressed state</pre>	and not
	uploading because it was created with
<pre>the 'local' option or if</pre>	automatic uploads of AutoSupports is
<pre>disabled</pre>	at the cluster level.
<pre>-h, --help</pre>	help for collect
<pre>-l, --local</pre>	Only collect and compress the
<pre>autosupport bundle. Do not upload</pre>	to support.
	Use 'download' to copy the collected


```

bundle after it is in
    --nodes string          the 'compressed' state
                           Specify nodes to collect for storage
component. (default "all")
    -t, --startTime string  StartTime is the starting time for
collection of AutoSupport.
                           This should be in the ISO 8601 date
time format.
                           Example format accepted:
                           2021-01-01T15:20:25Z, 2021-01-
01T15:20:25-05:00
    -u, --usermessage string  UserMessage is the additional message
to include in the
                           AutoSupport subject.
                           (default "Manual event trigger from
CLI")

```

部署測試應用程式

以下是部署可搭配Astra Data Store預覽使用的測試應用程式的步驟。

在此範例中、我們使用Helm儲存庫部署Bitnami的MongoDB圖表。

您需要的是 **#8217** ；需要的是什麼

- Astra Data Store預覽叢集已部署及設定
- Trident安裝完成

步驟

1. 從Bitnami新增Helm repo：

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

2. 部署MongoDB：

```
helm install mongohelm4 --set persistence.storageClass=trident-csi
bitnami/mongodb --namespace=ns-mongodb --create-namespace
```

3. 檢查MongoDB Pod的狀態：

```

~% kubectl get pods -n ns-mongodb
NAME                                READY   STATUS    RESTARTS   AGE
mongodb-9846ff8b7-rfr4r            1/1     Running   0           67s

```

4. 驗證MongoDB使用的持續磁碟區宣告 (PVC)：

```
~% kubectl get pvc -n ns-mongodb
NAME          STATUS    VOLUME                                     CAPACITY   ACCESS MODES
STORAGECLASS  AGE
mongodb       Bound     pvc-1133453a-e2f5-48a5                   8Gi        RWO
trident-csi   97s
```

5. 使用kubectl命令「Get astradsvolume」列出磁碟區：

```
~% kubectl get astradsvolume pvc-1133453a-e2f5-48a5 -n astrads-system
NAME          SIZE          IP           CLUSTER    CREATED
pvc-1133453a-e2f5-48a5  8830116Ki    10.192.2.192  jai-ads    true
```

6. 使用kubectl命令「describe astradsvolume」來描述磁碟區：

```
~% kubectl describe astradsvolume pvc-1133453a-e2f5-48a5 -n astrads-system
Name:          pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07
Namespace:     astrads-system
Labels:        astrads.netapp.io/cluster=jai-ads
               astrads.netapp.io/mip=10.192.1.39
               astrads.netapp.io/volumeUUID=cf33fd38-a451-596c-b656-61b8270d2b5e
               trident.netapp.io/cloud=on-prem
               trident.netapp.io/creator=trident-dev
               trident.netapp.io/performance=premium
Annotations:   provisioning: {"provisioning":{"cloud":"on-prem","creator":"trident-dev","performance":"premium"}}
               trident:
                 {"trident":{"version":"21.10.0-test.jenkins-trident-stable-v21.10-2+e03219ce37294d9ba54ec476bbe788c1a7772548","backendUUID":"","platform":...
API Version:   astrads.netapp.io/v1alpha1
Kind:          AstraDSVolume
Metadata:
  Creation Timestamp:  2021-12-08T19:35:26Z
  Finalizers:
    trident.netapp.io/astradsvolume-finalizer
    astrads.netapp.io/astradsvolume-finalizer
  Generation:        1
  Managed Fields:
    API Version:      astrads.netapp.io/v1alpha1
```

```

Fields Type:  FieldsV1
fieldsV1:
  f:metadata:
    f:labels:
      f:astrads.netapp.io/cluster:
      f:astrads.netapp.io/mip:
      f:astrads.netapp.io/volumeUUID:
    f:status:
      .:
      f:cluster:
      f:conditions:
      f:created:
      f:displayName:
      f:exportAddress:
      f:internalName:
      f:mip:
      f:permissions:
      f:qosPolicy:
      f:requestedSize:
      f:restoreCacheSize:
      f:size:
      f:snapshotReservePercent:
      f:state:
      f:volumePath:
      f:volumeUUID:
Manager:      cluster-controller
Operation:    Update
Time:         2021-12-08T19:35:32Z
API Version:  astrads.netapp.io/v1alpha1
Fields Type:  FieldsV1
fieldsV1:
  f:status:
    f:exportPolicy:
Manager:      dms-controller
Operation:    Update
Subresource:  status
Time:         2021-12-08T19:35:32Z
API Version:  astrads.netapp.io/v1alpha1
Fields Type:  FieldsV1
fieldsV1:
  f:metadata:
    f:annotations:
      .:
      f:provisioning:
      f:trident:
    f:finalizers:

```

```

    v:"trident.netapp.io/astradsvolume-finalizer":
    f:labels:
      .:
      f:trident.netapp.io/cloud:
      f:trident.netapp.io/creator:
      f:trident.netapp.io/performance:
    f:spec:
      .:
      f:cluster:
      f:displayName:
      f:exportPolicy:
      f:noSnapDir:
      f:permissions:
      f:qosPolicy:
      f:size:
      f:snapshotReservePercent:
      f:type:
      f:volumePath:
    Manager:          trident_orchestrator
    Operation:        Update
    Time:             2021-12-08T19:35:34Z
    Resource Version: 12007115
    UID:              d522ae4f-e793-49ed-bbe0-9112d7f9167b
Spec:
  Cluster:          jai-ads
  Display Name:     pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07
  Export Policy:    pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07
  No Snap Dir:      true
  Permissions:      0777
  Qos Policy:       silver
  Size:             9042036412
  Snapshot Reserve Percent: 5
  Type:             ReadWrite
  Volume Path:      /pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07
Status:
  Cluster:  jai-ads
  Conditions:
    Last Transition Time: 2021-12-08T19:35:32Z
    Message:             Volume is online
    Reason:              VolumeOnline
    Status:              True
    Type:                AstraDSVolumeOnline
    Last Transition Time: 2021-12-08T19:35:32Z
    Message:             Volume creation request was successful
    Reason:              VolumeCreated
    Status:              True

```

```

Type: AstraDSVolumeCreated
Created: true
Display Name: pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07
Export Address: 10.192.2.192
Export Policy: pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07
Internal Name: pvc_1133453a_e2f5_48a5_a06c_d14b8aa7be07
Mip: 10.192.1.192
Permissions: 777
Qos Policy: silver
Requested Size: 9042036412
Restore Cache Size: 0
Size: 8830116Ki
Snapshot Reserve Percent: 5
State: online
Volume Path: /pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07
Volume UUID: cf33fd38-a451-596c-b656-61b8270d2b5e
Events:
  Type      Reason           Age   From                      Message
  ----      -
  Normal    VolumeCreated    3m9s  ADSClusterController     Volume creation
request was successful

```

管理叢集

您可以透過Astra Data Store預覽使用kubectl命令來管理叢集。

- [\[Add a node\]](#)
- [\[Place a node in maintenance mode\]](#)
- [\[Replace a node\]](#)
- [\[Replace a drive\]](#)

您需要的是 **#8217** ；需要的是什麼

- 安裝了kubectl和kubectl-astrads外掛程式的系統。請參閱 ["安裝Astra Data Store預覽"](#)。

新增節點

您要新增的節點應該是Kubernetes叢集的一部分、而且其組態應該類似於叢集中的其他節點。

步驟

1. 如果新節點的dataIP尚未納入ADSCluster CR、請執行下列步驟：
 - a. 編輯astradscluster CR、並在「ads Data Networks Addresses（廣告資料網路位址）」欄位中新增額外的dataIP：

```
~% kubectl edit astradscluster <cluster-name> -n astrads-system
```

回應：

```
adsDataNetworks:
  -addresses: dataIP1,dataIP2,dataIP3,dataIP4,*newdataIP*
```

- a. 儲存CR。
- b. 將節點新增至Astra Data Store預覽叢集：

```
~% kubectl astrads nodes add --cluster <cluster-name>
```

2. 否則、只要新增節點即可：

```
~% kubectl astrads nodes add --cluster <cluster-name>
```

3. 確認已新增節點：

```
~% kubectl astrads nodes list
```

將節點置於維護模式

當您需要執行主機維護或套件升級時、應將節點置於維護模式。



節點必須已是Astra Data Store預覽叢集的一部分。

當節點處於維護模式時、您無法將節點新增至叢集。在此範例中、我們會將節點「nhcitj1525」置於維護模式。

步驟

1. 顯示節點詳細資料：

```
~% kubectl get nodes
```

回應：

NAME	STATUS	ROLES	AGE	VERSION
nhcitjj1525	Ready	<none>	3d18h	v1.20.0
nhcitjj1526	Ready	<none>	3d18h	v1.20.0
nhcitjj1527	Ready	<none>	3d18h	v1.20.0
nhcitjj1528	Ready	<none>	3d18h	v1.20.0
scs000039783-1	Ready	control-plane,master	3d18h	v1.20.0

2. 確保節點尚未處於維護模式：

```
~% kubectl astrads maintenance list
```

回應（維護模式中沒有節點）：

NAME	NODE NAME	IN MAINTENANCE	MAINTENANCE STATE	MAINTENANCE VARIANT
------	-----------	----------------	-------------------	---------------------

3. 啟用維護模式。

```
~% kubectl astrads maintenance create <cr-name> --node-name=<<node  
-name>> --variant=Node
```

範例：

```
~% kubectl astrads maintenance create maint1 --node-name="nhcitjj1525"  
--variant=Node  
Maintenance mode astrads-system/maint1 created
```

4. 列出節點。

```
~% kubectl astrads nodes list
```

回應：

NODE NAME	NODE STATUS	CLUSTER NAME
nhcitjj1525	Added	ftap-astra-012
...		

5. 檢查維護模式的狀態：

```
~% kubectl astrads maintenance list
```

回應：

NAME	NODE NAME	IN MAINTENANCE	MAINTENANCE STATE
MAINTENANCE VARIANT			
node4	nhcitjj1525	true	ReadyForMaintenance Node

在「維護中」模式的開頭是「假」、並變更為「真」。「維護狀態」從「準備維護」改為「就緒維護」。

6. 節點維護完成後、請停用維護模式：

```
~% kubectl astrads maintenance update maint1 --node-name="nhcitjj1525"
--variant=None
```

7. 確保節點不再處於維護模式：

```
~% kubectl astrads maintenance list
```

更換節點

使用KECBECVL命令搭配Astra Data Store預覽、以取代叢集中故障的節點。

步驟

1. 列出所有節點：

```
~% kubectl astrads nodes list
```

回應：

NODE NAME	NODE STATUS	CLUSTER NAME
sti-rx2540-534d..	Added	cluster-multinodes-21209
sti-rx2540-535d...	Added	cluster-multinodes-21209
...		

2. 描述叢集：

```
~% kubectl astrads clusters list
```


回應：

CLUSTER NAME	CLUSTER STATUS	NODE COUNT
cluster-multinodes-21209	created	4

3. 驗證故障節點上的「Node HA (節點HA)」是否標記為「假」：

```
~% kubectl describe astradscluster -n astrads-system
```

回應：

```
Name:          cluster-multinodes-21209
Namespace:     astrads-system
Labels:        <none>
Annotations:   kubectl.kubernetes.io/last-applied-configuration:

{"apiVersion":"astrads.netapp.io/v1alpha1","kind":"AstraDSCluster","meta
data":{"annotations":{},"name":"cluster-multinodes-21209","namespa...
API Version:   astrads.netapp.io/v1alpha1
Kind:          AstraDSCluster

State:         Disabled
Variant:       None
Node HA:       false
Node ID:       4
Node Is Reachable: false
Node Management IP: 172.21.192.192
Node Name:     sti-rx2540-532d.ctl.gdl.englab.netapp.com
Node Role:     Storage
Node UUID:     6f6b88f3-8411-56e5-b1f0-a8e8d0c946db
Node Version:  12.75.0.6167444
Status:       Added
```

4. 修改astradscluster cr,將「AdsNode Count」的值減至3、以移除故障節點：

```
cat manifests/astradscluster.yaml
```

回應：

```
apiVersion: astrads.netapp.io/v1alpha1
kind: AstraDSCluster
metadata:
```

```

name: cluster-multinodes-21209
namespace: astrads-system
spec:
  # ADS Node Configuration per node settings
  adsNodeConfig:
    # Specify CPU limit for ADS components
    # Supported value: 9
    cpu: 9
    # Specify Memory Limit in GiB for ADS Components.
    # Your kubernetes worker nodes need to have at least this much RAM
    free
    # for ADS to function correctly
    # Supported value: 34
    memory: 34
    # [Optional] Specify raw storage consumption limit. The operator
    # will only select drives for a node up to this limit
    capacity: 600
    # [Optional] Set a cache device if you do not want auto detection
    # e.g. /dev/sdb
    # cacheDevice: ""
    # Set this regex filter to select drives for ADS cluster
    # drivesFilter: ".*"

    # [Optional] Specify node selector labels to select the nodes for
    # creating ADS cluster
    # adsNodeSelector:
    #   matchLabels:
    #     customLabelKey: customLabelValue

    # Specify the number of nodes that should be used for creating ADS
    # cluster
    adsNodeCount: 3

    # Specify the IP address of a floating management IP routable from any
    # worker node in the cluster
    mvip: "172..."

    # Comma separated list of floating IP addresses routable from any host
    # where you intend to mount a NetApp Volume
    # at least one per node must be specified
    # addresses: 10.0.0.1,10.0.0.2,10.0.0.3,10.0.0.4,10.0.0.5
    # netmask: 255.255.255.0
    adsDataNetworks:
      - addresses: "172..."
        netmask: 255.255.252.0

```

```

# [Optional] Provide a k8s label key that defines which protection
domain a node belongs to
# adsProtectionDomainKey: ""

# [Optional] Provide a monitoring config to be used to setup/configure
a monitoring agent.
monitoringConfig:
  namespace: "netapp-monitoring"
  repo: "docker.repo.eng.netapp.com/global/astra"

autoSupportConfig:
  # AutoUpload defines the flag to enable or disable AutoSupport
  upload in the cluster (true/false)
  autoUpload: true
  # Enabled defines the flag to enable or disable automatic
  AutoSupport collection.
  # When set to false, periodic and event driven AutoSupport
  collection would be disabled.
  # It is still possible to trigger an AutoSupport manually while
  AutoSupport is disabled
  # enabled: true
  # CoredumpUpload defines the flag to enable or disable the upload of
  coredumps for this ADS Cluster
  # coredumpUpload: false
  # HistoryRetentionCount defines the number of local (not uploaded)
  AutoSupport Custom Resources to retain in the cluster before deletion
  historyRetentionCount: 25
  # DestinationURL defines the endpoint to transfer the AutoSupport
  bundle collection
  destinationURL: "https://testbed.netapp.com/put/AsupPut"
  # ProxyURL defines the URL of the proxy with port to be used for
  AutoSupport bundle transfer
  # proxyURL:
  # Periodic defines the config for periodic/scheduled AutoSupport
  objects
  periodic:
    # Schedule defines the Kubernetes Cronjob schedule
    - schedule: "0 0 * * *"
    # PeriodicConfig defines the fields needed to create the
    Periodic AutoSupports
    periodicconfig:
      - component:
          name: storage
          event: dailyMonitoring
          userMessage: Daily Monitoring Storage AutoSupport bundle
          nodes: all

```

```
- component:
  name: controlplane
  event: daily
  userMessage: Daily Control Plane AutoSupport bundle
```

5. 驗證節點是否已從叢集移除：

```
~% kubectl get nodes --show-labels
```

回應：

NAME	STATUS	ROLES
sti-astramaster-237	Ready	control-plane,master
sti-rx2540-532d	Ready	<none>
sti-rx2540-533d	Ready	<none>

```
~% kubectl astrads nodes list
```

回應：

NODE NAME	NODE STATUS	CLUSTER NAME
sti-rx2540-534d	Added	cluster-multinodes-21209
sti-rx2540-535d	Added	cluster-multinodes-21209
sti-rx2540-536d	Added	cluster-multinodes-21209

```
~% kubectl get nodes --show-labels
```

回應：

NAME	STATUS	ROLES	AGE	VERSION
sti-astramaster-237	Ready	control-plane,master	24h	v1.20.0
sti-rx2540-532d	Ready	<none>	24h	v1.20.0

```
~% kubectl describe astradscluster -n astrads-system
```

回應：

```
Name:          cluster-multinodes-21209
Namespace:     astrads-system
Labels:        <none>
Kind:          AstraDSCluster
Metadata:
...
```

6. 修改叢集CR、將節點新增至叢集以進行更換。節點數會遞增至4。確認已挑選新節點進行新增。

```
rvi manifests/astradscluster.yaml
cat manifests/astradscluster.yaml
apiVersion: astrads.netapp.io/v1alpha1
kind: AstraDSCluster
metadata:
  name: cluster-multinodes-21209
  namespace: astrads-system
```

```
~% kubectl apply -f manifests/astradscluster.yaml
```

回應：

```
astradscluster.astrads.netapp.io/cluster-multinodes-21209 configured
```

```
~% kubectl get pods -n astrads-system
```

回應：

NAME	READY	STATUS	RESTARTS	AGE
astrads-cluster-controller...	1/1	Running	1	24h
astrads-deployment-support...	3/3	Running	0	24h
astrads-ds-cluster-multinodes-21209	1/1	Running		

```
~% kubectl astrads nodes list
```

回應：

NODE NAME	NODE STATUS	CLUSTER NAME
sti-rx2540-534d...	Added	cluster-multinodes-21209
sti-rx2540-535d...	Added	cluster-multinodes-21209

```
~% kubectl astrads clusters list
```

回應：

CLUSTER NAME	CLUSTER STATUS	NODE COUNT
cluster-multinodes-21209	created	4

```
~% kubectl astrads drives list
```

回應：

DRIVE NAME	DRIVE ID	DRIVE STATUS	NODE NAME	CLUSTER NAME
scsi-36000...	c3e197f2...	Active	sti-rx2540...	cluster-multinodes-21209

更換磁碟機

當叢集中的磁碟機故障時、必須儘快更換磁碟機、以確保資料完整性。當磁碟機故障時、您會在叢集CR節點狀態、叢集健全狀況資訊和度量端點中看到故障磁碟機資訊。

顯示**nodeStatuses.driveStatuses**中故障磁碟機的叢集範例

```
$ kubectl get adscl -A -o yaml
```

回應：

```

...
apiVersion: astrads.netapp.io/v1alpha1
kind: AstraDSCluster
...
nodeStatuses:
  - driveStatuses:
    - driveID: 31205e51-f592-59e3-b6ec-185fd25888fa
      driveName: scsi-36000c290ace209465271ed6b8589b494
      drivesStatus: Failed
    - driveID: 3b515b09-3e95-5d25-a583-bee531ff3f31
      driveName: scsi-36000c290ef2632627cb167a03b431a5f
      drivesStatus: Active
    - driveID: 0807fa06-35ce-5a46-9c25-f1669def8c8e
      driveName: scsi-36000c292c8fc037c9f7e97a49e3e2708
      drivesStatus: Active
  ...

```

故障磁碟機CR會在叢集中自動建立、名稱對應於故障磁碟機的UUID。

```
$ kubectl get adsfd -A -o yaml
```

回應：

```

...
apiVersion: astrads.netapp.io/v1alpha1
kind: AstraDSFailedDrive
metadata:
  name: c290a-5000-4652c-9b494
  namespace: astrads-system
spec:
  executeReplace: false
  replaceWith: ""
status:
  cluster: arda-6e4b4af
  failedDriveInfo:
    failureReason: AdminFailed
    inUse: false
    name: scsi-36000c290ace209465271ed6b8589b494
    path: /dev/disk/by-id/scsi-36000c290ace209465271ed6b8589b494
    present: true
    serial: 6000c290ace209465271ed6b8589b494
    node: sti-rx2540-300b.ctl.gdl.englab.netapp.com
  state: ReadyToReplace

```

```
~% kubectl astrads faileddrive list --cluster arda-6e4b4af
```

回應：

NAME	NODE	CLUSTER	STATE
AGE			
6000c290	sti-rx2540-300b.lab.netapp.com	ard-6e4b4af	ReadyToReplace
13m			

步驟

1. 使用「`kubectl astrads show-replacement`」命令列出可能的更換磁碟機、該命令可篩選符合更換限制的磁碟機（未在叢集內使用、未掛載、無分割區、等於或大於故障磁碟機）。

若要列出所有磁碟機而不篩選可能的更換磁碟機、請在「`show -replacement`」命令中新增「`-all`」。

```
~% kubectl astrads faileddrive show-replacements --cluster ard-6e4b4af  
--name 6000c290
```

回應：

NAME	IDPATH	SERIAL	PARTITIONCOUNT	MOUNTED	SIZE
sdh	/scsi-36000c29417	45000c	0	false	100GB

2. 使用「放置」命令、以通過的序號取代磁碟機。命令會完成替換、如果經過「`-wait`」時間、則會失敗。

```
~% kubectl astrads faileddrive replace --cluster arda-6e4b4af --name  
6000c290 --replaceWith 45000c --wait  
Drive replacement completed successfully
```



如果使用不適當的「`-replaceWith`」序號來執行「`kubectl astrads`故障磁碟機更換」、則會出現類似以下的錯誤：


```
~% kubectl astrads replacedrive replace --cluster astrads-cluster-
f51b10a --name 6000c2927 --replaceWith BAD_SERIAL_NUMBER
Drive 6000c2927 replacement started
Failed drive 6000c2927 has been set to use BAD_SERIAL_NUMBER as a
replacement
...
Drive replacement didn't complete within 25 seconds
Current status: {FailedDriveInfo:{InUse:false Present:true Name:scsi-
36000c2 FiretapUUID:444a5468 Serial:6000c Path:/scsi-36000c
FailureReason:AdminFailed Node:sti-b200-0214a.lab.netapp.com}
Cluster:astrads-cluster-f51b10a State:ReadyToReplace
Conditions:[{Message: "Replacement drive serial specified doesn't
exist", Reason: "DriveSelectionFailed", Status: False, Type:' Done'}]}
```

3. 若要重新執行磁碟機更換、請使用之前的命令「-force」：

```
~% kubectl astrads replacedrive replace --cluster astrads-cluster-
f51b10a --name 6000c2927 --replaceWith VALID_SERIAL_NUMBER --force
```

以取得更多資訊

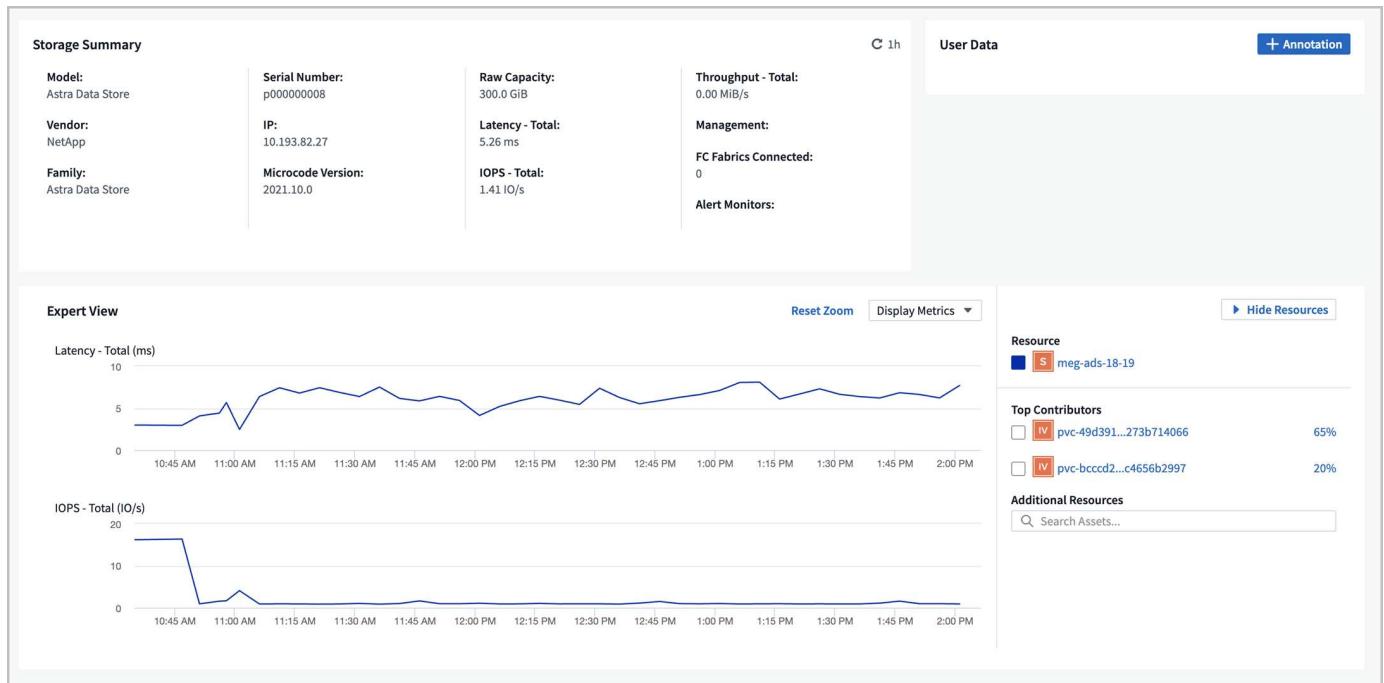
- ["使用kveccli命令管理Astra Data Store預覽資產"](#)

利用功能表監控指標Cloud Insights

您可以使用Cloud Insights 支援功能來監控Astra Data Store預覽指標。

- [\[Complete Cloud Insights connection prerequisite tasks\]](#)
- [\[Acquisition Unit storage\]](#)
- [\[Download and run the installation script\]](#)
- [\[Edit the Cloud Insights connection\]](#)
- [\[Disconnect from Cloud Insights\]](#)

以下是顯示在畫面上的Astra Data Store預覽指標範例Cloud Insights
：



您也可以使用、顯示Astra Data Store預覽中產生的度量清單 [\[Open Metrics API help\]](#)。

完成Cloud Insights 連線先決條件工作

在將Astra Data Store與Cloud Insights 支援功能整線之前、您必須先完成下列工作：

- "安裝Astra Data Store監控操作員" 這是Astra Data Store預覽安裝說明的一部分。
- "安裝kubectl-astrads二進位檔" 這是Astra Data Store預覽安裝說明的一部分。
- "建立Cloud Insights 一個不一樣的帳戶"。
- 請確認下列命令可用：「awk、curl、grep」和「jq」

收集下列資訊：

- * Cloud Insights 具備類別讀寫權限的API存取權杖*：擷取單元、資料收集、資料擷取和記錄擷取。這將用於讀取/寫入作業、設定擷取單位、以及設定資料擷取程序。
- * Kubernetes API伺服器IP位址和連接埠*。這是用來監控Astra Data Store預覽叢集。
- * Kubernetes API權杖*。這是用來呼叫Kubernetes API。
- 持續磁碟區組態。有關如何配置持續磁碟區的資訊。

擷取單元儲存

擷取單元需要三個持續磁碟區來儲存安裝檔案、組態資料和記錄。監控操作員使用預設儲存類別來建立持續的Volume宣告。您可以在執行安裝程式指令碼時、使用「-s」選項來指定不同的儲存類別名稱。

如果您的Kubernetes叢集沒有儲存資源配置程式（例如NetApp Trident）、您可以在執行安裝程式指令碼時、使用「-r」選項來提供本機檔案系統路徑。設定「-r」選項時、安裝程式指令碼會在所提供的目錄內建立三個持續磁碟區。此目錄需要至少150 GB的可用空間。

下載並執行安裝指令碼

提供Bash指令碼、可透過監控操作員啟用Astra Data Store預覽監控功能。Cloud Insights安裝指令碼會安裝擷取單元、其中含有Astra Data Store Collector、Telegraf代理程式及Fluent位元代理程式。

下載時、將會在安裝程式指令碼中內嵌選定的「更新網域名稱」和「選定的更新API存取權杖」Cloud Insights Cloud Insights。

然後、會以下列方式傳送指標：

- Telegraf會將指標傳送到Cloud Insights 這個數據湖。
- Fluent位元會將記錄傳送至記錄擷取服務。

顯示安裝程式指令碼說明

安裝程式指令碼的完整說明文字如下所示：

顯示安裝程式指令碼說明文字：

```
./cloudinsights-ads-monitoring.sh -h
```

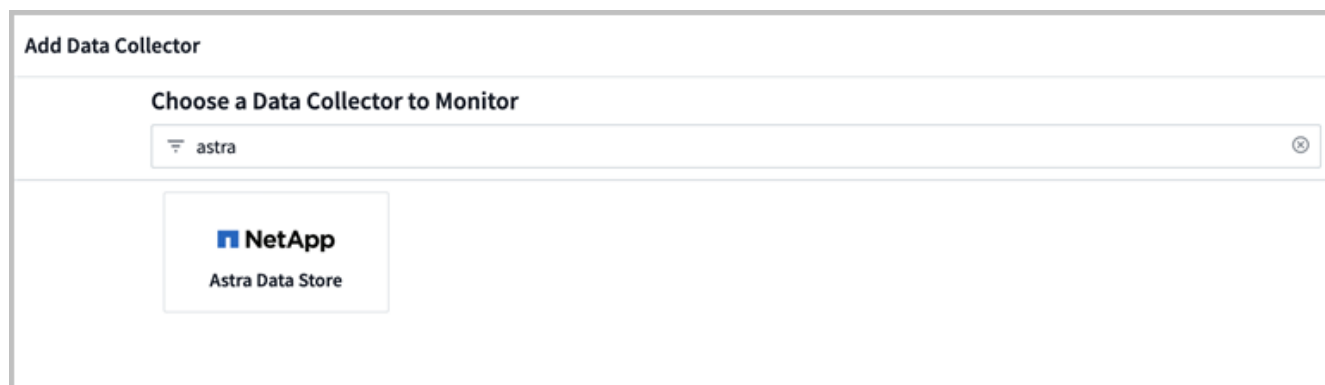
回應：

```
USAGE: cloudinsights-ads-monitoring.sh [OPTIONS]
Configure monitoring of Astra Data Store by Cloud Insights.
OPTIONS:
  -h                                Display this help message.
  -d ci_domain_name                 Cloud Insights tenant domain name.
  -i kubernetes_ip                 Kubernetes API server IP address.
  -k ci_api_key                     Cloud Insights API Access Token.
  -n namespace                      Namespace for monitoring components. (default:
netapp-monitoring)
  -p kubernetes_port                Kubernetes API server port. (default: 6443)
  -r root_pv_dir                    Create 3 Persistent Volumes in this directory
for the Acquisition Unit.
                                   Only specify this option if there is no Storage
Provisioner installed and the PVs do not already exist.
  -s storage_class                  Storage Class name for provisioning Acquisition
Unit PVs. If not specified, the default storage class will be used.
  -t kubernetes_token              Kubernetes API server token.
```

執行安裝指令碼


1. 如果Cloud Insights 您還沒有這個帳戶、請建立一個這個帳戶。
2. 登入Cloud Insights 到

3. 在「支援資料」功能表中、按一下「管理」>「資料收集器」Cloud Insights。
4. 按一下「+資料收集器」以新增收集器。




5. 按一下「* Astra Data Store*」方塊。
6. 選取正確Cloud Insights 的「循環API」存取權杖、或建立新的權杖。
7. 請依照指示下載安裝程式指令碼、更新權限、然後執行指令碼。


此指令碼包含Cloud Insights 您的URL、以及所選Cloud Insights 的循環API存取權杖。



Select a Data Collector



Configure Collector



NetApp
Astra Data Store

Configure Collector

Configure Kubernetes Operator to monitor NetApp Astra Data Store (ADS).

What Operating System or Platform Are You Using?

Kubernetes

Select existing API Access Token or create a new one

default_ads_api_key1 (...d0gHof)

+ API Access Token

Production Best Practices ?

Configure Astra Data Store Need Help?

- 1

The commands *awk*, *curl*, *grep*, *jq*, *kubect*l and the *kubect*l-*astrads* plugin must be installed where the installer script is run. You will need the Kubernetes API server IP address and a Kubernetes API token to run this install script. See the documentation if you need help finding this information.
- 2

Copy Installer Script

☐ Reveal Installer Script

#!/usr/bin/env bash
SCRIPT='basename \$0'

CI_DOMAIN_NAME="f49uaky.gstabler-ads.cloudinsights-dev.netapp.com"
CI_API_KEY="eyJraWQ1OiI5OTk5IiwidHlwIjoI5ldUIiwiaWxnIjoI5FMzODQifQ.eyJjcVhdG9yTG9naW41OiJhZG1pb2IiImRpc3BsYXl0YW11IjoIZGVmYXVsdF9hZHNfYXBPX2tleTEgKG9uIGJlaGFsZiBvZiBhZG1pb2I"
- 3

Copy the above installer script and save it as *cloudinsights-ads-monitoring.sh*
- 4

Copy Permissions Command

☐ Reveal Permissions Command

chmod +x cloudinsights-ads-monitoring.sh
- 5

Paste the permissions command in a terminal to enable execute permissions on the installer script.
- 6

Copy Install Command

☐ Reveal Install Command

./cloudinsights-ads-monitoring.sh -i <KUBERNETES_IP> -t <KUBERNETES_TOKEN>
- 7

Paste the install command in a terminal, replace the placeholders with the correct values for your environment, and run the command. It will take several minutes to complete. The script will use the default namespace 'netapp-monitoring'. Additional options are available for customized environments. The default configuration for kubectl should point to the kubernetes cluster to be monitored.
- 8

Complete Setup

8. 指令碼完成後、按一下*完成設定*。

安裝指令碼完成後、Astra Data Store收集器會出現在資料集區清單中。



如果指令碼因為錯誤而結束、您可以在錯誤解決之後再次執行。如果您的環境未使用預設設定、指令碼可支援其他參數、例如監控操作員命名空間和Kubernetes API伺服器連接埠。請使用「./cloudinsights-ads-monitoring.sh -h」中的「-h」選項查看使用量和說明文字。

安裝指令碼會在組態成功時產生類似的輸出：

```
Configuring Cloud Insights monitoring for Astra Data Store . . .
Configuring monitoring namespace
...
Configuring output sink and Fluent Bit plugins
Configuring Telegraf plugins
Configuring Acquisition Unit
...
Acquisition Unit has been installed successfully.
Configuring Astra Data Store data collector
Astra Data Store collector data '<CLUSTER_NAME>' created
Configuration done!
```

代理程式CR範例

以下是執行安裝程式指令碼後、「monitoring NetApp」代理程式CR的外觀範例。

```
spec:
  au:
    isEnabled: true
    storageClassName: auto-sc
  cluster-name: meg-ads-21-22-29-30
  docker-repo: docker.repo.eng.netapp.com/global/astra
  fluent-bit:
  - name: ads-tail
    outputs:
    - sink: ADS_STDOUT
    substitutions:
    - key: TAG
      value: firetapems
    - key: LOG_FILE
      values:
      - /var/log/firetap/*/ems/ems
      - /var/log/firetap/ems/*/ems/ems
    - key: ADS_CLUSTER_NAME
      value: meg-ads-21-22-28-29-30
  - name: agent
  - name: ads-tail-ci
    outputs:
```

```

- sink: CI
substitutions:
- key: TAG
  value: netapp.ads
- key: LOG_FILE
  values:
  - /var/log/firetap/*/ems/ems
  - /var/log/firetap/ems/*/ems/ems
- key: ADS_CLUSTER_NAME
  value: meg-ads-21-22-28-29-30
output-sink:
- api-key: abcd
  domain-name: bz19ngz.gst-adsdemo.ci-dev.netapp.com
  name: CI
serviceAccount: sa-netapp-monitoring
telegraf:
- name: ads-open-metric
  outputs:
  - sink: CI
  run-mode:
  - ReplicaSet
  substitutions:
  - key: URLS
    values:
    - http://astrads-metrics-service.astrads-
system.svc.cluster.local:9341
  - key: METRIC_TYPE
    value: ads-metric
  - key: ADS_CATEGORY
    value: netapp_ads
  - key: ADS_CLUSTER_NAME
    value: meg-ads-21-22-28-29-30
- name: agent
status:
  au-pod-status: UP
  au-uuid: eddeccc6-3aa3-4dd2-a98c-220085fae6a9

```

編輯Cloud Insights 此鏈接

您稍後可以編輯Kubernetes API權杖或Cloud Insights 是使用此功能的循環API存取權杖：

- 如果您想要更新Kubernetes API權杖、您應該從Cloud Insights 這個UI編輯Astra Data Store收集器。
- 如果您想要更新Cloud Insights 遙測和記錄所用的循環API存取權杖、您應該使用kubectl命令來編輯監控操作員CR。

更新Kubernetes API權杖

1. 登入Cloud Insights 到
2. 選取*管理*>*資料收集器*以存取「資料收集器」頁面。
3. 尋找Astra Data Store叢集的項目。
4. 按一下頁面右側的功能表、然後選取*編輯*。
5. 使用新值更新Kubernetes API Token欄位。
6. 選取*儲存Collector *

更新Cloud Insights 程式：更新程式碼

1. 登入Cloud Insights 到
2. 選取「管理>* API存取*」、然後按一下「+ API存取權杖」、即可建立新Cloud Insights 的「循環API存取權杖」。
3. 編輯Agent CR:

```
kubectl --namespace netapp-monitoring edit agent agent-monitoring-netapp
```

4. 找到"output-sink（輸出接收器）"區段、找到名稱為"CI"的項目。
5. 如需標籤「API-金鑰」、請將目前值改為全新Cloud Insights 的REAPI存取權杖。

此區段如下所示：

```
output-sink:
  - api-key: <api key value>
    domain-name: <tenant url>
    name: CI
```

6. 儲存並結束編輯器視窗。

監控營運者將更新Telegraf和Fluent位元、以使用新Cloud Insights 的版的更新版的BIOS API存取權杖。

中斷Cloud Insights 與該功能的連線

若要中斷Cloud Insights 與功能表的連線、您必須Cloud Insights 先從功能表上刪除Astra Data Store收集器。完成後、您可以從監控操作員移除擷取單元、Telegraf和Fluent位元組態。

移除Astra Data Store預覽收集器

1. 登入Cloud Insights 到
2. 選取*管理*>*資料收集器*以存取「資料收集器」頁面。
3. 尋找Astra Data Store叢集的項目。

4. 選取畫面右側的功能表、然後選取*刪除*。
5. 按一下確認頁面上的*刪除*。

移除擷取單元、Telegraf和Fluent位元

1. 編輯Agent CR:

```
kubectl --namespace netapp-monitoring edit agent agent-monitoring-netapp
```

2. 找到「au」區段、並將「isEnabled」設為「假」
3. 找到「Fluent位元」區段、然後移除名為「ads tail-CI」的外掛程式。如果沒有其他外掛程式、您可以移除「Fluent位元」區段。
4. 找到「Telegraf」區段、移除名為「ads開放式指標」的外掛程式。如果沒有其他外掛程式、您可以移除「Telegraf」區段。
5. 找到「output-sink（輸出接收器）」區段、然後移除名為「CI」的接收器。
6. 儲存並結束編輯器視窗。

監控操作員將更新Telegraf和Fluent位元組態、並刪除擷取單元Pod。

7. 如果您將本機目錄用於擷取單元PV、而非儲存資源配置程式、請刪除PV：

```
kubectl delete pv au-lib au-log au-pv
```

然後、刪除正在執行擷取單元的節點上的實際目錄。

8. 在擷取單元Pod刪除之後、您可以從Cloud Insights 功能表中刪除擷取單元。
 - a. 在「支援資料」功能表中、選取*管理*>*資料收集器*。Cloud Insights
 - b. 按一下「擷取單位」標籤。
 - c. 按一下擷取設備Pod旁的功能表。
 - d. 選擇*刪除*。

監控操作員會更新Telegraf和Fluent位元組態、並移除擷取單元。

Open Metrics API說明

以下是可用來從Astra Data Store預覽收集度量的API清單。

- 「說明」行說明指標。
- 「類型」行指出度量是量表還是計數器。

```
# HELP astrads_cluster_capacity_logical_percent Percentage cluster logical capacity that is used (0-100)
```

```

# TYPE astrads_cluster_capacity_logical_percent gauge
# HELP astrads_cluster_capacity_max_logical Max Logical capacity of the
cluster in bytes
# TYPE astrads_cluster_capacity_max_logical gauge
# HELP astrads_cluster_capacity_max_physical The sum of the space in the
cluster in bytes for storing data after provisioning efficiencies, data
reduction algorithms and replication schemes are applied
# TYPE astrads_cluster_capacity_max_physical gauge
# HELP astrads_cluster_capacity_ops The IO operations capacity of the
cluster
# TYPE astrads_cluster_capacity_ops gauge
# HELP astrads_cluster_capacity_physical_percent The percentage of cluster
physical capacity that is used (0-100)
# TYPE astrads_cluster_capacity_physical_percent gauge
# HELP astrads_cluster_capacity_used_logical The sum of the bytes of data
in all volumes in the cluster before provisioning efficiencies, data
reduction algorithms and replication schemes are applied
# TYPE astrads_cluster_capacity_used_logical gauge
# HELP astrads_cluster_capacity_used_physical Used Physical capacity of a
cluster in bytes
# TYPE astrads_cluster_capacity_used_physical gauge
# HELP astrads_cluster_other_latency The sum of the accumulated latency in
seconds for other IO operations of all the volumes in a cluster. Divide by
astrads_cluster_other_ops to get the average latency per other operation
# TYPE astrads_cluster_other_latency counter
# HELP astrads_cluster_other_ops The sum of the other IO operations of all
the volumes in a cluster
# TYPE astrads_cluster_other_ops counter
# HELP astrads_cluster_read_latency The sum of the accumulated latency in
seconds of read IO operations of all the volumes in a cluster. Divide by
astrads_cluster_read_ops to get the average latency per read operation
# TYPE astrads_cluster_read_latency counter
# HELP astrads_cluster_read_ops The sum of the read IO operations of all
the volumes in a cluster
# TYPE astrads_cluster_read_ops counter
# HELP astrads_cluster_read_throughput The sum of the read throughput of
all the volumes in a cluster in bytes
# TYPE astrads_cluster_read_throughput counter
# HELP astrads_cluster_storage_efficiency Efficacy of data reduction
technologies. (logical used / physical used)
# TYPE astrads_cluster_storage_efficiency gauge
# HELP astrads_cluster_total_latency The sum of the accumulated latency in
seconds of all IO operations of all the volumes in a cluster. Divide by
astrads_cluster_total_ops to get average latency per operation
# TYPE astrads_cluster_total_latency counter
# HELP astrads_cluster_total_ops The sum of the IO operations of all the

```

```

volumes in a cluster
# TYPE astrads_cluster_total_ops counter
# HELP astrads_cluster_total_throughput The sum of the read and write
throughput of all the volumes in a cluster in bytes
# TYPE astrads_cluster_total_throughput counter
# HELP astrads_cluster_utilization_factor The ratio of the current cluster
IO operations based on recent IO sizes to the cluster iops capacity. (0.0
- 1.0)
# TYPE astrads_cluster_utilization_factor gauge
# HELP astrads_cluster_volume_used The sum of used capacity of all the
volumes in a cluster in bytes
# TYPE astrads_cluster_volume_used gauge
# HELP astrads_cluster_write_latency The sum of the accumulated latency in
seconds of write IO operations of all the volumes in a cluster. Divide by
astrads_cluster_write_ops to get the average latency per write operation
# TYPE astrads_cluster_write_latency counter
# HELP astrads_cluster_write_ops The sum of the write IO operations of all
the volumes in a cluster
# TYPE astrads_cluster_write_ops counter
# HELP astrads_cluster_write_throughput The sum of the write throughput of
all the volumes in a cluster in bytes
# TYPE astrads_cluster_write_throughput counter
# HELP astrads_disk_base_seconds Base for busy, pending and queued.
Seconds since collection began
# TYPE astrads_disk_base_seconds counter
# HELP astrads_disk_busy Seconds the disk was busy. 100 *
(astrads_disk_busy / astrads_disk_base_seconds) = percent busy (0-100)
# TYPE astrads_disk_busy counter
# HELP astrads_disk_capacity Raw Capacity of a disk in bytes
# TYPE astrads_disk_capacity gauge
# HELP astrads_disk_io_pending Summation of the count of pending io
operations for a disk times time. Divide by astrads_disk_base_seconds to
get the average pending operation count
# TYPE astrads_disk_io_pending counter
# HELP astrads_disk_io_queued Summation of the count of queued io
operations for a disk times time. Divide by astrads_disk_base_seconds to
get the average queued operations count
# TYPE astrads_disk_io_queued counter
# HELP astrads_disk_read_latency Total accumulated latency in seconds for
disk reads. Divide by astrads_disk_read_ops to get the average latency per
read operation
# TYPE astrads_disk_read_latency counter
# HELP astrads_disk_read_ops Total number of read operations for a disk
# TYPE astrads_disk_read_ops counter
# HELP astrads_disk_read_throughput Total bytes read from a disk
# TYPE astrads_disk_read_throughput counter

```

```

# HELP astrads_disk_write_latency Total accumulated latency in seconds for
disk writes. Divide by astrads_disk_write_ops to get the average latency
per write operation
# TYPE astrads_disk_write_latency counter
# HELP astrads_disk_write_ops Total number of write operations for a disk
# TYPE astrads_disk_write_ops counter
# HELP astrads_disk_write_throughput Total bytes written to a disk
# TYPE astrads_disk_write_throughput counter
# HELP astrads_value_scrape_duration Duration to scrape values
# TYPE astrads_value_scrape_duration gauge
# HELP astrads_volume_capacity_available The minimum of the available
capacity of a volume and the available capacity of the cluster in bytes
# TYPE astrads_volume_capacity_available gauge
# HELP astrads_volume_capacity_available_logical Logical available
capacity of a volume in bytes
# TYPE astrads_volume_capacity_available_logical gauge
# HELP astrads_volume_capacity_percent Percentage of volume capacity
available (0-100). (capacity available / provisioned) * 100
# TYPE astrads_volume_capacity_percent gauge
# HELP astrads_volume_capacity_provisioned Provisioned capacity of a
volume in bytes after setting aside the snapshot reserve. (size - snapshot
reserve = provisioned)
# TYPE astrads_volume_capacity_provisioned gauge
# HELP astrads_volume_capacity_size Total capacity of a volume in bytes
# TYPE astrads_volume_capacity_size gauge
# HELP astrads_volume_capacity_snapshot_reserve_percent Snapshot reserve
percentage of a volume (0-100)
# TYPE astrads_volume_capacity_snapshot_reserve_percent gauge
# HELP astrads_volume_capacity_snapshot_used The amount of volume snapshot
data that is not in the active file system in bytes
# TYPE astrads_volume_capacity_snapshot_used gauge
# HELP astrads_volume_capacity_used Used capacity of a volume in bytes.
This is bytes in the active filesystem unless snapshots are consuming more
than the snapshot reserve. (bytes in the active file system + MAX(0,
snapshot_used-(snapshot_reserve_percent/100*size))
# TYPE astrads_volume_capacity_used gauge
# HELP astrads_volume_other_latency Total accumulated latency in seconds
for operations on a volume that are neither read or write. Divide by
astrads_volume_other_ops to get the average latency per other operation
# TYPE astrads_volume_other_latency counter
# HELP astrads_volume_other_ops Total number of operations for a volume
that are neither read or write
# TYPE astrads_volume_other_ops counter
# HELP astrads_volume_read_latency Total accumulated read latency in
seconds for a volume. Divide by astrads_volume_read_ops to get the average
latency per read operation

```

```
# TYPE astrads_volume_read_latency counter
# HELP astrads_volume_read_ops Total number of read operations for a
volume
# TYPE astrads_volume_read_ops counter
# HELP astrads_volume_read_throughput Total read throughput for a volume
in bytes
# TYPE astrads_volume_read_throughput counter
# HELP astrads_volume_total_latency Total accumulated latency in seconds
for all operations on a volume. Divide by astrads_volume_total_ops to get
the average latency per operation
# TYPE astrads_volume_total_latency counter
# HELP astrads_volume_total_ops Total number of operations for a volume
# TYPE astrads_volume_total_ops counter
# HELP astrads_volume_total_throughput Total throughput for a volume in
bytes
# TYPE astrads_volume_total_throughput counter
# HELP astrads_volume_write_latency Total accumulated write latency in
seconds for volume. Divide by astrads_volume_write_ops to get the average
latency per write operation
# TYPE astrads_volume_write_latency counter
# HELP astrads_volume_write_ops Total number of write operations for a
volume
# TYPE astrads_volume_write_ops counter
# HELP astrads_volume_write_throughput Total write throughput for a volume
in bytes
# TYPE astrads_volume_write_throughput counter
```

使用Prometheus和Grafana監控指標

您可以使用Prometheus和Grafana監控Astra Data Store預覽指標。您可以設定Prometheus、從Astra Data Store預覽Kubernetes叢集度量端點收集度量、也可以使用Grafana來視覺化度量資料。

您需要的是 **#8217** ；需要的是什麼

- 請確定您已在Astra Data Store預覽叢集或其他可與Astra Data Store預覽叢集通訊的叢集上下載並安裝Prometheus和Grafana套件。請依照正式文件中的指示安裝每個工具：
 - ["安裝Prometheus"](#)
 - ["安裝Grafana"](#)
- Prometheus和Grafana需要能夠與Astra Data Store預覽Kubernetes叢集通訊。如果未在Astra Data Store預覽叢集上安裝Prometheus和Grafana、您必須確保它們能與Astra Data Store預覽叢集上執行的度量服務通訊。

設定Prometheus

Astra Data Store預覽會在Kubernetes叢集中的TCP連接埠9341上公開度量服務。您必須設定Prometheus、才能從此服務收集指標。

步驟

1. 編輯Prometheus安裝的「Prometheus.yml」組態檔案。
2. 新增指向Astra Data Store預覽服務名稱及其連接埠的服務目標。例如：

```
scrape_configs:
static_configs:
- targets: ['astrads-metrics-service.astrads-system:9341']
```

3. 啟動Prometheus服務。

設定Grafana

您可以設定Grafana以顯示Prometheus收集的指標。

步驟

1. 編輯Grafana安裝的「datasources.yml」組態檔。
2. 將Prometheus新增為資料來源。例如：

```
apiVersion: 1

datasources:
- name: astradatastore-prometheus
  type: prometheus
  access: proxy
  url: http://localhost:9090
  jsonData:
    manageAlerts: false
```

3. 啟動Grafana服務。
4. 請依照Grafana文件中的指示進行 ["開始使用"](#)。

匯入Grafana儀表板範本

您下載以安裝Astra Data Store預覽的套裝組合檔案包含Grafana儀表板範本檔案、可從Grafana匯入。這些儀表板範本可協助您查看Astra Data Store預覽中可用的度量類型、以及如何檢視這些資料。

步驟

1. 開啟Astra Data Store preview ".tar.gz"套裝組合。
2. 開啟「manifest」目錄。
3. 擷取「grafana_cluster.json」和「grafana_volume.json」檔案。
4. 使用Grafana網路UI、["將儀表板範本檔案匯入至Grafana"](#)。

設定及監控事件記錄

若要監控事件管理系統（EMS）記錄、您可以執行下列高層級工作：

- [\[Configure monitoring in the Astra Data Store preview cluster custom resource \(CR\)\]](#)
- [\[Set up Cloud Insights\]](#)
- [\[Stream event logs to Elastic\]](#)。

在Astra Data Store預覽叢集自訂資源（CR）中設定監控

如果尚未在Astra Data Store預覽叢集CR上設定監控選項、您可以使用「astrads」擴充功能來設定。

輸入：

```
~% kubectl astrads monitoring setup -n <NAMESPACE OF AGENT INSTALLED>
-r <DOCKER REPO TO FIND FLUENT/TELEGRAF ETC IMAGES>
```

其中：

- 安裝代理程式的命名空間：輸入監控代理程式的命名空間、這是監控操作員監控NetApp CR的預設名稱。
- 您可以選擇在Docker登錄中設定Fluent或Telegraf影像所在的位置。根據預設、路徑會設為「docker.repo.eng.netapp.com/global/astra`」、您可以變更此路徑。

設定Cloud Insights 功能

若要檢視記錄、Cloud Insights 可選用設定功能不需使用；不過、使用Cloud Insights 畫面來檢視資料是很有幫助的。請參閱 ["如何設定NetApp Cloud Insights 解決方案"](#) 用於Astra Data Store預覽。

串流事件記錄至Elastic

若要將EMS事件和其他Pod記錄串流至第三方端點（例如Elastic）、請使用「astrads」延伸功能。

輸入：

```
~% kubectl astrads monitoring --host <ELASTIC HOST NAME> --port <ELASTIC
HOST PORT> es
```



彈性主機名稱可以是IP位址。

使用Astra Control Center搭配Astra Data Store預覽

您可以使用Astra Control Center使用者介面（UI）來執行Astra Data Store預覽工作。

設定Astra Control Center進行Astra Data Store預覽

若要使用Astra Control Center UI進行Astra Data Store預覽、您必須完成下列工作：

- "將執行Astra Data Store的基礎Kubernetes叢集新增至Astra Control Center"。
- "將Astra Data Store預覽新增為Astra Control Center的儲存後端"。



如果您新增儲存後端、但沒有具備Astra Data Store預覽功能的Kubernetes叢集、則必須先新增叢集。

您可以在Astra控制中心做什麼

設定Astra Control Center for Astra Data Store預覽之後、您就可以使用Astra Control Center UI來完成下列工作：

- "使用Astra Control Center監控Astra Data Store預覽資產的健全狀況"。
- "管理Astra Data Store預覽後端儲存設備"。
- "監控節點、磁碟和持續磁碟區宣告 (PVCS) "。

以取得更多資訊

- "Astra系列簡介"
- "Astra Control Center文件"
- "Astra Control API"

使用自動指令碼解除安裝Astra Data Store預覽

若要解除安裝Astra Data Store預覽和控制面板、您必須移除工作負載、繫結、磁碟區、匯出原則、Astra Data Store叢集、授權、部署環境、以及Astra Data Store預覽命名空間。

或者、您也可以 "無需指令碼即可解除安裝Astra Data Store預覽"。

您需要的是 **#8217** ；需要的是什麼

- root系統管理權限

Astra Data Store預覽解除安裝程序會引導您完成下列高層級步驟：

- [Remove existing workloads and bindings]
- [Uninstall Astra Data Store cluster]
- [Validate the removal of the astrads-system namespace]
- [Ensure containers are not running on worker nodes]
- [Delete OpenShift Container Platform resources]
- [Troubleshoot the Astra Data Store preview uninstall process]

移除現有的工作負載和繫結

在解除安裝Astra Data Store預覽之前、您必須先移除下列項目

- 所有使用Astra Data Store預覽做為儲存後端的應用程式工作負載
- Trident繫結使用Astra Data Store預覽做為後端

如此可確保Kubernetes環境保持乾淨狀態、這在重新安裝時非常重要。

解除安裝Astra Data Store叢集

若要解除安裝Astra Data Store預覽、您可以使用從NetApp支援網站下載的Astra Data Store tar檔案中的「uninstall.sh」指令碼。

1. 在「manifest」目錄中找到「uninstall.sh」。
2. 執行下列「shed (已執行)」命令：

```
sed -i -e 's~netappdsoperator.yaml~astradsoperator.yaml~' uninstall.sh
```

3. 執行下列指令碼、指出您要解除安裝的項目：

```
./uninstall.sh

You must run this script with an argument specifying what should be
uninstalled
To uninstall the ADS cluster run ./uninstall.sh cluster
To uninstall everything run ./uninstall all
```

4. 如果您只想卸載叢集、請輸入「uninstall.sh <cluster >」

否則、如果您想要解除安裝所有項目、請輸入「uninstall.sh」



在大多數情況下、您都會將所有項目解除安裝。如果您想在之後重新部署叢集、可能只想要解除安裝叢集。

5. 出現提示時、請確認您要繼續、然後輸入「eraseDDATA」

回應：

```
./uninstall.sh all

Enter 'erasedata' to confirm you want proceed with the uninstall:
erasedata
+-----+
| Wed Feb  2 10:14:01 EST 2022                |
```

```

| ADS cluster uninstall started |
+-----+
Deleting astradsvolumes
Deleted astradsvolumes
Deleting astradsexportpolicies
Deleted astradsexportpolicies
Deleting astradsvolumesnapshots
Deleted astradsvolumesnapshots
Deleting astradsclusters
Deleting astradsclusters
Deleting astradslicenses
Deleted astradslicenses

+-----+
| Wed Feb  2 10:15:18 EST 2022 |
| ADS cluster uninstall done   |
+-----+

+-----+
| Wed Feb  2 10:15:18 EST 2022 |
| ADS system uninstall started  |
+-----+

Removing astradsversion
astradsversion.astrads.netapp.io "astradsversion" deleted
Removed astradsversion
Removing daemonsets
daemonset.apps "astrads-ds-nodeinfo-astradsversion" deleted
Removed daemonsets
Removing deployments
deployment.apps "astrads-cluster-controller" deleted
deployment.apps "astrads-license-controller" deleted
deployment.apps "astrads-operator" deleted
Removed deployments
Removing all other AstraDS resources
namespace "astrads-system" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsautosupports.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradscloudsnapshots.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsclusters.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsexportpolicies.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsfaileddrives.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io

```

```
"astradslicenses.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsnfsoptions.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsnodeinfoes.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsnodemanagements.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsqospolicies.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsversions.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsvolumefiles.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsvolumes.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsvolumesnapshots.astrads.netapp.io" deleted
role.rbac.authorization.k8s.io "astrads-astrads-system-admin-role"
deleted
role.rbac.authorization.k8s.io "astrads-astrads-system-reader-role"
deleted
role.rbac.authorization.k8s.io "astrads-astrads-system-writer-role"
deleted
role.rbac.authorization.k8s.io "astrads-leader-election-role" deleted
role.rbac.authorization.k8s.io "astrads-manager-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astrads-admin-
clusterrole" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astrads-reader-
clusterrole" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astrads-writer-
clusterrole" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsautosupport-
editor-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsautosupport-
viewer-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscloudsnapshot-
editor-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscloudsnapshot-
viewer-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscluster-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscluster-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsexportpolicy-
editor-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsexportpolicy-
```

```
viewer-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsfaileddrive-
editor-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsfaileddrive-
viewer-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradslicense-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradslicense-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsnfsoption-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsnfsoption-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsnodeinfo-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsnodeinfo-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsnodemanagement-
editor-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsnodemanagement-
viewer-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsqospolicy-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsversion-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsversion-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolume-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolume-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolumeeditor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolumeeditor-
viewer-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolumesnapshot-
editor-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolumesnapshot-
viewer-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-manager-role" deleted
rolebinding.rbac.authorization.k8s.io "astrads-astrads-admin-
rolebinding" deleted
rolebinding.rbac.authorization.k8s.io "astrads-astrads-reader-
rolebinding" deleted
rolebinding.rbac.authorization.k8s.io "astrads-astrads-writer-
rolebinding" deleted
```

```

rolebinding.rbac.authorization.k8s.io "astrads-leader-election-
rolebinding" deleted
rolebinding.rbac.authorization.k8s.io "astrads-manager-rolebinding"
deleted
clusterrolebinding.rbac.authorization.k8s.io "astrads-astrads-admin-
rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "astrads-astrads-reader-
rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "astrads-astrads-writer-
rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "astrads-manager-
rolebinding" deleted
configmap "astrads-autosupport-cm" deleted
configmap "astrads-firetap-cm" deleted
configmap "astrads-kevents-asup" deleted
configmap "astrads-metrics-cm" deleted
secret "astrads-autosupport-certs" deleted
+-----+
| Wed Feb  2 10:16:36 EST 2022                |
| ADS system uninstall done                    |
+-----+

```

驗證刪除astrad-system命名空間

請確定下列命令沒有傳回任何結果：

```
kubectl get ns | grep astrads-system
```

確保容器未在工作節點上執行

驗證「firetap」或「netwd」等容器是否未在工作節點上執行。在每個節點上執行下列項目。

```
ssh <mynode1>
# runc list
```

刪除OpenShift Container Platform資源

如果您在Red Hat OpenShift Container Platform (OCP) 上安裝Astra Data Store預覽、則可以解除安裝OCP安全內容限制 (SCC) 和角色繫結資源。

OpenShift使用安全內容限制 (SCC) 來控制Pod可以執行的動作。

完成標準的解除安裝程序之後、請完成下列步驟。

1. 移除SCC資源：

```
oc delete -f ads_privileged_scc.yaml
```

2. 移除角色繫結資源：

```
oc delete -f oc_role_bindings.yaml
```



請忽略這些步驟中的「找不到資源」錯誤。

3. 從所有Kubernetes節點移除「/var/lib/kubelet/config.yaml」。

Astra Data Store預覽解除安裝程序疑難排解

Kubernetes v1.20中的Astra Data Store預覽解除安裝程序、偶爾會導致Pod處於終止狀態。

如果發生此問題、請執行下列命令、強制刪除「astrads-system」命名空間中的所有Pod：

```
kubect1 delete pods --all -n astrads-system --force --grace-period 0
```

無需指令碼即可解除安裝Astra Data Store預覽

若要在不使用自動指令碼的情況下手動解除安裝Astra Data Store預覽、您將會移除工作負載、繫結、磁碟區、匯出原則、叢集、授權、部署環境及Astra Data Store預覽命名空間。

或者、您也可以 "[使用指令碼解除安裝Astra Data Store預覽](#)"。

您需要的是 **#8217** ；需要的是什麼

- root系統管理權限

Astra Data Store預覽解除安裝程序會引導您完成下列高層級步驟：

- [\[Remove existing workloads and bindings\]](#)
- [\[Uninstall the Astra Data Store preview cluster and control plane\]](#)
- [\[Delete the license\]](#)
- [\[Delete the Astra Data Store preview installation\]](#)
- [\[Validate the removal of the astrads-system namespace\]](#)
- [\[Ensure containers are not running on worker nodes\]](#)
- [\[Delete OpenShift Container Platform resources\]](#)
- [\[Troubleshoot the Astra Data Store preview uninstall process\]](#)

移除現有的工作負載和繫結

在解除安裝Astra Data Store預覽之前、您必須先移除下列項目

- 所有使用Astra Data Store預覽做為儲存後端的應用程式工作負載
- Trident繫結使用Astra Data Store預覽做為後端

如此可確保Kubernetes環境保持乾淨狀態、這在重新安裝時非常重要。

解除安裝Astra Data Store預覽叢集和控制面板

請依照下列步驟手動解除安裝Astra Data Store預覽。

刪除磁碟區並匯出原則

刪除叢集之前、您應該刪除Astra Data Store預覽磁碟區和匯出原則。



如果您未先刪除磁碟區和匯出原則、叢集刪除程序會暫停、直到Astra Data Store預覽磁碟區物件遭到刪除為止。在開始刪除叢集之前移除這些項目會更有效率。

步驟

1. 刪除磁碟區：

```
~% kubectl delete astradsvolumes --all -A
~% kubectl get astradsvolumes -A
```

2. 刪除匯出原則：

```
~% kubectl delete astradsexportpolicies --all -A
~% kubectl get astradsexportpolicies -A
```

刪除Astra Data Store預覽叢集

刪除叢集只會刪除Astra Data Store預覽叢集物件自訂資源（CR）以及叢集範圍的資源。



即使刪除叢集、運算子、nodeinfo Pod和叢集控制器（即Kubernetes範圍內的資源）仍會保留。

刪除叢集也會從節點解除安裝基礎作業系統、這會停止「fifetap」和「netwd」服務。

卸載程式需要大約一分鐘的時間才能完成。然後、Astra Data Store預覽叢集範圍內的資源便會開始移除。

1. 刪除叢集：

```
~% kubectl delete astradsclusters --all -A
~% kubectl get astradsclusters -A
```

刪除授權

1. 對叢集中的每個工作節點執行SSH、並驗證「fifetap」或「netwd」未在工作節點中執行。
2. 刪除Astra Data Store預覽授權：

```
~% kubectl delete astradslicenses --all -A
~% kubectl get astradslicenses -A
```

刪除Astra Data Store預覽安裝

刪除叢集中的控制器、運算子、命名空間和支援Pod。

1. 刪除Astra Data Store預覽安裝物件：

```
~% kubectl delete astradsversion astradsversion -n astrads-system
~% kubectl get astradsversion -n astrads-system
```

2. 刪除資料儲存示範與所有Astra Data Store預覽控制器資源：

```
~% kubectl delete ds --all -n astrads-system
~% kubectl get ds -n astrads-system

~% kubectl delete deployments --all -n astrads-system
~% kubectl get deployments -n astrads-system
```

3. 刪除剩餘成品和運算子yaml檔案：

```
~% kubectl delete -f ./manifests/astradsoperator.yaml
~% kubectl get pods -n astrads-system
```

驗證刪除astrad-system命名空間

請確定下列命令沒有傳回任何結果：

```
~% kubectl get ns | grep astrads-system
```


確保容器未在工作節點上執行

驗證「fifetap」或「netwd」等容器是否未在工作節點上執行。在每個節點上執行下列項目。

```
ssh <mynode1>  
# runc list
```

刪除OpenShift Container Platform資源

如果您在Red Hat OpenShift Container Platform (OCP) 上安裝Astra Data Store預覽、則可以解除安裝OCP安全內容限制 (SCC) 和角色繫結資源。

OpenShift使用安全內容限制 (SCC) 來控制Pod可以執行的動作。

完成標準的解除安裝程序之後、請完成下列步驟。

1. 移除SCC資源：

```
oc delete -f ads_privileged_scc.yaml
```

2. 移除角色繫結資源：

```
oc delete -f oc_role_bindings.yaml
```



請忽略這些步驟中的「找不到資源錯誤」。

3. 從所有Kubernetes節點移除「/var/lib/kubelet/config.yaml」。

手動刪除範例

以下是執行手動解除安裝指令碼的範例。

```
$ kubectl delete astradsvolumes --all -A  
No resources found  
$ kubectl delete astradsexportpolicies --all -A  
No resources found  
$ kubectl delete astradsclusters --all -A  
astradscluster.astrads.netapp.io "astrads-sti-c6220-09-10-11-12" deleted  
  
$ kubectl delete astradslicenses --all -A  
astradslicense.astrads.netapp.io "e900000005" deleted  
  
$ kubectl delete astradsdeployment astradsdeployment -n astrads-system  
astradsdeployment.astrads.netapp.io "astradsdeployment" deleted
```

```

$ kubectl delete ds --all -n astrads-system
daemonset.apps "astrads-ds-astrads-sti-c6220-09-10-11-12" deleted
daemonset.apps "astrads-ds-nodeinfo-astradsdeployment" deleted
daemonset.apps "astrads-ds-support" deleted

$ kubectl delete deployments --all -n astrads-system
deployment.apps "astrads-cluster-controller" deleted
deployment.apps "astrads-deployment-support" deleted
deployment.apps "astrads-license-controller" deleted
deployment.apps "astrads-operator" deleted

$ kubectl delete -f ../../firetap/sds/manifests/netappsdsoperator.yaml
namespace "astrads-system" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsautosupports.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradscloudsnapshots.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsclusters.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsdeployments.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsexportpolicies.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsfaileddrives.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradslicenses.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsnfsoptions.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsnodeinfoes.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsqospolicies.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsvolumefiles.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsvolumes.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsvolumesnapshots.astrads.netapp.io" deleted
role.rbac.authorization.k8s.io "astrads-leader-election-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscloudsnapshot-
editor-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscloudsnapshot-
viewer-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscluster-editor-role"

```

```
deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscluster-viewer-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradslicense-editor-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradslicense-viewer-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolume-editor-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolume-viewer-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-autosupport-editor-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-autosupport-viewer-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-manager-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-metrics-reader" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappexportpolicy-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappexportpolicy-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappsdsdeployment-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappsdsdeployment-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappsdsnfsoption-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappsdsnfsoption-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappsdsnodeinfo-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappsdsnodeinfo-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-proxy-role" deleted
rolebinding.rbac.authorization.k8s.io "astrads-leader-election-
rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "astrads-manager-rolebinding"
deleted
clusterrolebinding.rbac.authorization.k8s.io "astrads-proxy-rolebinding"
deleted
configmap "astrads-autosupport-cm" deleted
configmap "astrads-firetap-cm" deleted
configmap "astrads-fluent-bit-cm" deleted
configmap "astrads-kevents-asup" deleted
configmap "astrads-metrics-cm" deleted
service "astrads-operator-metrics-service" deleted
```

```
Error from server (NotFound): error when deleting
"/.../export/firetap/sds/manifests/netappsdsoperator.yaml":
deployments.apps "astrads-operator" not found

$ kubectl get ns | grep astrads-system

[root@sti-rx2540-535c ~]# runc list
ID          PID          STATUS      BUNDLE      CREATED      OWNER
```

Astra Data Store預覽解除安裝程序疑難排解

Kubernetes v1.20中的Astra Data Store預覽解除安裝程序、偶爾會導致Pod處於終止狀態。

如果發生此問題、請執行下列命令、強制刪除「astrad-system」命名空間中的所有Pod：

```
kubectl delete pods --all -n astrads-system --force --grace-period 0
```

版權資訊

Copyright©2022 NetApp、Inc.版權所有。美國印製本文件中版權所涵蓋的任何部分、不得以任何形式或任何方式（包括影印、錄製、在未事先取得版權擁有者書面許可的情況下、在電子擷取系統中進行錄音或儲存。

衍生自受版權保護之NetApp資料的軟體必須遵守下列授權與免責聲明：

本軟體係由NetApp「依現狀」提供、不含任何明示或暗示的保證、包括但不限於適售性及特定用途適用性的暗示保證、特此聲明。在任何情況下、NetApp均不對任何直接、間接、偶發、特殊、示範、或衍生性損害（包括但不限於採購替代商品或服務；使用損失、資料或利潤損失；或業務中斷）、無論是在合約、嚴格責任或侵權行為（包括疏忽或其他）中、無論是因使用本軟體而產生的任何責任理論（包括疏忽或其他）、即使已被告知可能造成此類損害。

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。除非NetApp以書面明確同意、否則NetApp不承擔因使用本文所述產品而產生的任何責任或責任。使用或購買本產品並不代表NetApp擁有任何專利權利、商標權利或任何其他智慧財產權。

本手冊所述產品可能受到一或多個美國國家/地區的保護專利、國外專利或申請中。

限制權利圖例：政府使用、複製或揭露受DFARS 252.277-7103（1988年10月）和FAR 52-227-19（1987年6月）技術資料與電腦軟體權利條款（c）（1）（ii）分段所述限制。

商標資訊

NetApp、NetApp標誌及所列的標章 <http://www.netapp.com/TM> 為NetApp、Inc.的商標。其他公司和產品名稱可能為其各自所有者的商標。