使用Astra資料儲存區

Astra Data Store

NetApp
June 01, 2022

目錄

使用Astra資料儲存區

使用kvecll命令管理Astra Data Store資源

您可以使用kubectl命令及Kubernetes API擴充、來管理Astra Data Store資源。

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"。

如需叢集維護資訊、請參閱 "管理叢集"。

您需要的是 **#8217** ；需要的是什麼

- 您安裝的Astra Data Store KECVECL外掛程式 "安裝Astra Data Store"

列出Astra Data Store的Kubernetes自訂API資源

您可以使用Kubernetes內部的kubectl命令、與Astra Data Store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會在內部用來管理叢集。

此清單特別有助於取得每個Astra Data Store物件的簡短名稱、以減少您的輸入、如稍後所示。

1. 顯示Astra Data Store的Kubernetes自訂API資源清單：

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

   回應：

```
NAME                    SHORTNAMES  APIGROUP            NAMESPACED  KIND
astradsautosupports     adsas   astrads.netapp.io/v1alpha1 true
AstraDSAutoSupport
astradscloudsnapshots   adscs   astrads.netapp.io/v1alpha1 true
AstraDSCloudSnapshot
astradsclusters         adscl   astrads.netapp.io/v1alpha1 true
AstraDSCluster
astradsexportpolicies   adsep   astrads.netapp.io/v1alpha1 true
AstraDSExportPolicy
astradsfaileddrives     adsfd   astrads.netapp.io/v1alpha1 true
AstraDSFailedDrive
astradslicenses         adsli   astrads.netapp.io/v1alpha1 true
AstraDSLicense
astradsnfsoptions       adsnf   astrads.netapp.io/v1alpha1 true
AstraDSNfsOption
astradsnodeinfoes       adsni   astrads.netapp.io/v1alpha1 true
AstraDSNodeInfo
astradsnodemanagements adsnm   astrads.netapp.io/v1alpha1 true
AstraDSNodeManagement
astradsqospolicies      adsqp   astrads.netapp.io/v1alpha1 true
AstraDSQosPolicy
astradsversions         adsve   astrads.netapp.io/v1alpha1 true
AstraDSVersion
astradsvolumefiles      adsvf   astrads.netapp.io/v1alpha1 true
AstraDSVolumeFiles
astradsvolumes          adsvo   astrads.netapp.io/v1alpha1 true
AstraDSVolume
astradsvolumesnapshots adsvs   astrads.netapp.io/v1alpha1 true
AstraDSVolumeSnapshot
```

2. 若要在Kubernetes叢集中取得所有目前的Astra資料儲存區物件、請使用「kubectl Get ads -A」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
CREATED
astrads-system   test234    21Gi   172.25.138.109   astrads-cluster-
9f1c99f   true
astrads-system   test2345   21Gi   172.25.138.111   astrads-cluster-
9f1c99f   true
```

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. 針對Astra Data Store KECVECtl副檔名中的所有命令顯示說明：

```
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
                                         repeated to specify multiple
groups.
      --cache-dir string               Default HTTP cache directory
                                         (default "/u/arda/.kube/http-
cache")
      --certificate-authority string   Path to a cert file for the
certificate authority
      --client-certificate string      Path to a client certificate file
for TLS
      --client-key string              Path to a client key file for TLS
      --cluster string                 The name of the kubeconfig
cluster to use
      --context string                 The name of the kubeconfig
context to use
  -h, --help                           help for astrads
        --insecure-skip-tls-verify       If true, the server's
certificate will not be checked
                                         for validity. This will make
your HTTPS connections insecure
        --kubeconfig string              Path to the kubeconfig file to
use for CLI requests.
    -n, --namespace string               If present, the namespace scope
for this CLI request
        --request-timeout string         The length of time to wait
before giving up on a single
                                         server request. Non-zero values
should contain a
                                         corresponding time unit (e.g.
1s, 2m, 3h).
                                         A value of zero means don't
timeout requests.
                                         (default "0")
    -s, --server string                  The address and port of the
Kubernetes API server
        --token string                   Bearer token for authentication
to the API server
        --user string                    The name of the kubeconfig user
to use
```

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:
      -c, --component string     Specify the component to collect:
[storage , controlplane , vasaprovider, all]
      -d, --duration int         Duration is the duration in hours from
the startTime for collection
                                   of AutoSupport.
                                   This should be a positive integer
      -e, --event string         Specify the callhome event to trigger.
(default "manual")
      -f, --forceUpload          Configure an AutoSupport to upload if
it is in the compressed state
                                   and not
                                   uploading because it was created with
the 'local' option or if
                                   automatic uploads of AutoSupports is
```

```
disabled
                                    at the cluster level.
      -h, --help                   help for collect
      -l, --local                  Only collect and compress the
autosupport bundle. Do not upload
                                    to support.
                                    Use 'download' to copy the collected
bundle after it is in
                                    the 'compressed' state
           --nodes string           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kubecl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kubecl命令「desscribe 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Astra Data Store叢集

您可以使用搭配Astra Data Store的kubecl命令來管理叢集。

- [Add a node]
- [Remove a node]
- [Place a node in maintenance mode]
- [Add drives to a node]
- [Replace a drive]

您需要的是 **#8217** ；需要的是什麼

- 安裝了kubectl和kubecl-astrads外掛程式的系統。請參閱 "安裝Astra Data Store"。

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。

若要使用Astra Control Center新增節點、請參閱 "將節點新增至儲存後端叢集"。

步驟

1. 如果新節點的dataIP尚未納入Astra Data Store叢集CR、請執行下列步驟：

   a. 編輯叢集CR、並在「adsDataNetworks」*「Addresses」（位址）*欄位中新增額外的dataIP。以適合您環境的適當值取代以大寫字母顯示的資訊：

   ```
   kubectl edit astradscluster CLUSTER_NAME -n astrads-system
   ```

   回應：

   ```
   adsDataNetworks:
       -addresses:  dataIP1,dataIP2,dataIP3,dataIP4,NEW_DATA_IP
   ```

   a. 儲存CR。

   b. 將節點新增至Astra Data Store叢集。以適合您環境的適當值取代以大寫字母顯示的資訊：

   ```
   kubectl astrads nodes add --cluster CLUSTER_NAME
   ```

2. 否則、只要新增節點即可。以適合您環境的適當值取代以大寫字母顯示的資訊：

   ```
   kubectl astrads nodes add --cluster CLUSTER_NAME
   ```

3. 確認已新增節點：

   ```
   kubectl astrads nodes list
   ```

移除節點

搭配Astra Data Store使用kubectl命令來移除叢集中的節點。

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. 標記要移除的節點。以適合您環境的適當值取代以大寫字母顯示的資訊：

```
kubectl astrads nodes remove NODE_NAME
```

回應：

```
Removal label set on node sti-rx2540-534d.lab.org
Successfully updated ADS cluster cluster-multinodes-21209 desired node
count from 4 to 3
```

標記要移除的節點後、節點狀態應從「active」（作用中）變更為「present」（目前）。

3. 驗證移除節點的「Present（目前）」狀態：

```
kubectl get nodes --show-labels
```

回應：

```
NAME                                                STATUS   ROLES
AGE     VERSION   LABELS
sti-astramaster-050.lab.org   Ready    control-plane,master   3h39m
v1.20.0
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/
arch=amd64,kubernetes.io/hostname=sti-astramaster-
050.lab.org,kubernetes.io/os=linux,node-role.kubernetes.io/control-
plane=,node-role.kubernetes.io/master=
sti-rx2540-556a.lab.org       Ready    worker                 3h38m
v1.20.0   astrads.netapp.io/cluster=astrads-cluster-
890c32c,astrads.netapp.io/storage-cluster-
status=active,beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,
kubernetes.io/arch=amd64,kubernetes.io/hostname=sti-rx2540-
556a.lab.org,kubernetes.io/os=linux,node-role.kubernetes.io/worker=true
sti-rx2540-556b.lab.org       Ready    worker                 3h38m
v1.20.0   astrads.netapp.io/cluster=astrads-cluster-
890c32c,astrads.netapp.io/storage-cluster-
status=active,beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,
kubernetes.io/arch=amd64,kubernetes.io/hostname=sti-rx2540-
556b.lab.org,kubernetes.io/os=linux,node-role.kubernetes.io/worker=true
sti-rx2540-534d.lab.org       Ready    worker                 3h38m
v1.20.0   astrads.netapp.io/storage-cluster-
status=present,astrads.netapp.io/storage-node-
removal=,beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kuber
netes.io/arch=amd64,kubernetes.io/hostname=sti-rx2540-
557a.lab.org,kubernetes.io/os=linux,node-role.kubernetes.io/worker=true
sti-rx2540-557b.lab.org       Ready    worker                 3h38m
v1.20.0   astrads.netapp.io/cluster=astrads-cluster-
890c32c,astrads.netapp.io/storage-cluster-
status=active,beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,
kubernetes.io/arch=amd64,kubernetes.io/hostname=sti-rx2540-
557b.lab.org,kubernetes.io/os=linux,node-role.kubernetes.io/worker=true
```

4. 從節點解除安裝Astra Data Store。以適合您環境的適當值取代以大寫字母顯示的資訊：

```
kubectl astrads nodes uninstall NODE_NAME
```

5. 驗證節點是否已從叢集移除：

```
kubectl astrads nodes list
```

節點會從Astra資料儲存區移除。

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。以適合您環境的適當值取代以大寫字母顯示的資訊：

```
kubectl astrads maintenance create CR_NAME --node-name=NODE_NAME
--variant=Node
```

例如：

```
kubectl astrads maintenance create maint1 --node-name="nhcitjj1525"
--variant=Node
```

回應：

```
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
```

新增磁碟機至節點

搭配Astra Data Store使用kubectl命令、將實體或虛擬磁碟機新增至Astra Data Store叢集中的節點。

您需要的是 **#8217** ；需要的是什麼

- 符合下列條件的一或多個磁碟機：
  - 已安裝在節點（實體磁碟機）或新增至節點VM（虛擬磁碟機）
  - 磁碟機上沒有分割區
  - 叢集目前未使用磁碟機
  - 磁碟機原始容量不超過叢集中的授權原始容量（例如、授權每個CPU核心提供2TB的儲存容量、10個節點的叢集最大原始磁碟機容量為20TB）
  - 磁碟機至少為節點中其他作用中磁碟機的大小

> Astra Data Store每個節點不需要超過16個磁碟機。如果您嘗試新增第17個磁碟機、磁碟機新增要求將遭拒。

步驟

1. 描述叢集：

```
kubectl astrads clusters list
```

回應：

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

2. 記下叢集名稱。
3. 顯示可新增至叢集中所有節點的磁碟機。以叢集名稱取代叢集名稱：

```
kubectl astrads drives adddrive show-available --cluster=CLUSTER_NAME
```

回應：

```
Node: node1.name
Add drive maximum size: 100.0 GiB
Add drive minimum size: 100.0 GiB
NAME IDPATH SERIAL PARTITIONCOUNT SIZE ALREADYINCLUSTER
sdg /dev/disk/by-id/scsi-3c290e16d52479a9af5eac c290e16d52479a9af5eac 0
100 GiB false
sdh /dev/disk/by-id/scsi-3c2935798df68355dee0be c2935798df68355dee0be 0
100 GiB false

Node: node2.name
Add drive maximum size: 66.7 GiB
Add drive minimum size: 100.0 GiB
No suitable drives to add exist.

Node: node3.name
Add drive maximum size: 100.0 GiB
Add drive minimum size: 100.0 GiB
NAME IDPATH SERIAL PARTITIONCOUNT SIZE ALREADYINCLUSTER
sdg /dev/disk/by-id/scsi-3c29ee82992ed7a36fc942 c29ee82992ed7a36fc942 0
100 GiB false
sdh /dev/disk/by-id/scsi-3c29312aa362469fb3da9c c29312aa362469fb3da9c 0
100 GiB false

Node: node4.name
Add drive maximum size: 66.7 GiB
Add drive minimum size: 100.0 GiB
No suitable drives to add exist.
```

4. 執行下列其中一項：

   - 如果所有可用磁碟機的名稱都相同、您可以將其同時新增至各自的節點。以適合您環境的適當值取代以大寫字母顯示的資訊：

     ```
     kubectl astrads drives adddrive create --cluster=CLUSTER_NAME --name
     REQUEST_NAME --drivesbyname all=DRIVE_NAME
     ```

   - 如果磁碟機的命名方式不同、您可以一次新增一個磁碟機至各自的節點（您需要針對每個需要新增的磁碟機重複此步驟）。以適合您環境的適當值取代以大寫字母顯示的資訊：

     ```
     kubectl astrads drives adddrive create --cluster=CLUSTER_NAME --name
     REQUEST_NAME --drivesbyname NODE_NAME=DRIVE_NAME
     ```

Astra Data Store會建立新增磁碟機的要求、並顯示訊息、顯示要求的結果。

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。如果磁碟機故障、您可以在叢集CR節點狀態、叢集健全狀況資訊和度量端點中查看故障磁碟機的相關資訊。您可以使用下列命令範例來查看故障磁碟機資訊。

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.lab.org
   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 filleddrive show-replacees」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（未在叢集中使用、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

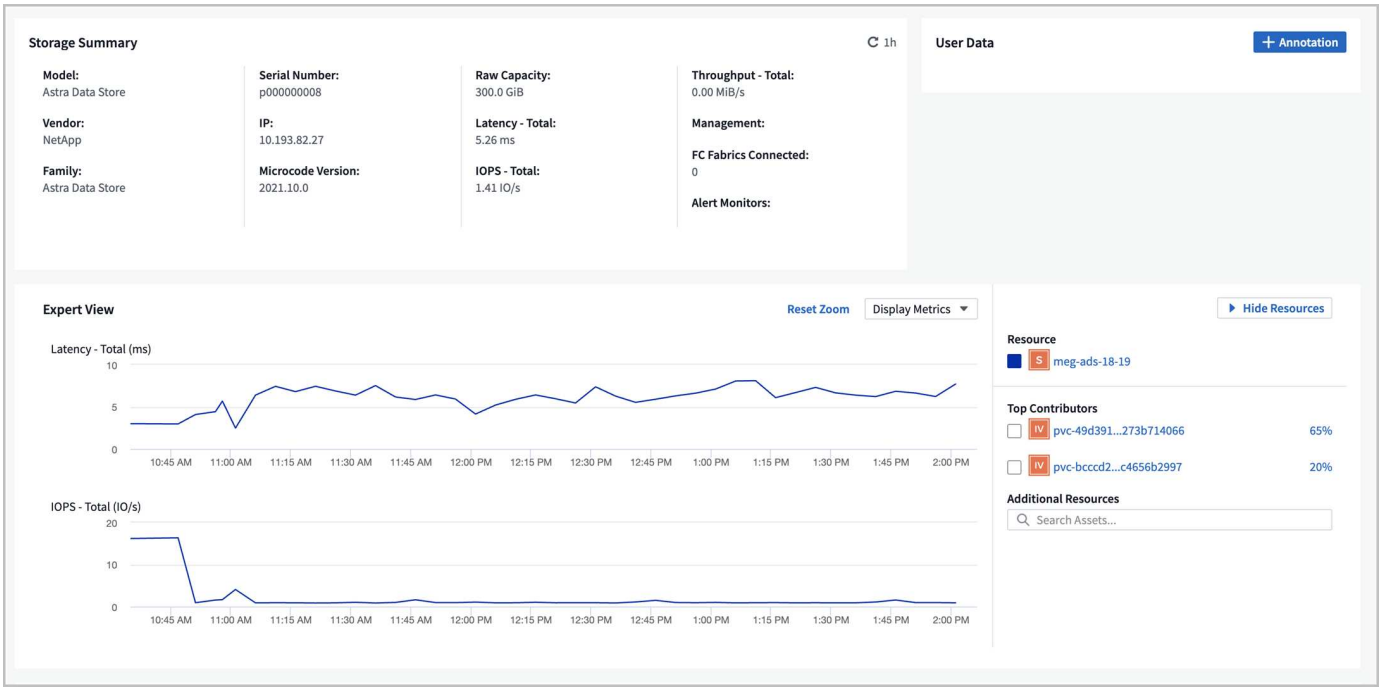
- "使用kvecll命令管理Astra Data Store資源"

監控Astra資料儲存區

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指標。

以下是一些展示Cloud Insights 在《支援》中的Astra Data Store指標範例：



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

您可以完成下列工作：

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

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安裝說明的一部分。
- "安裝kubecl-astrads二進位檔" 這是Astra Data Store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收集器和一個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：

- 這個部門將會將指標傳送到這個數據湖。Cloud Insights Cloud Insights
- 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：

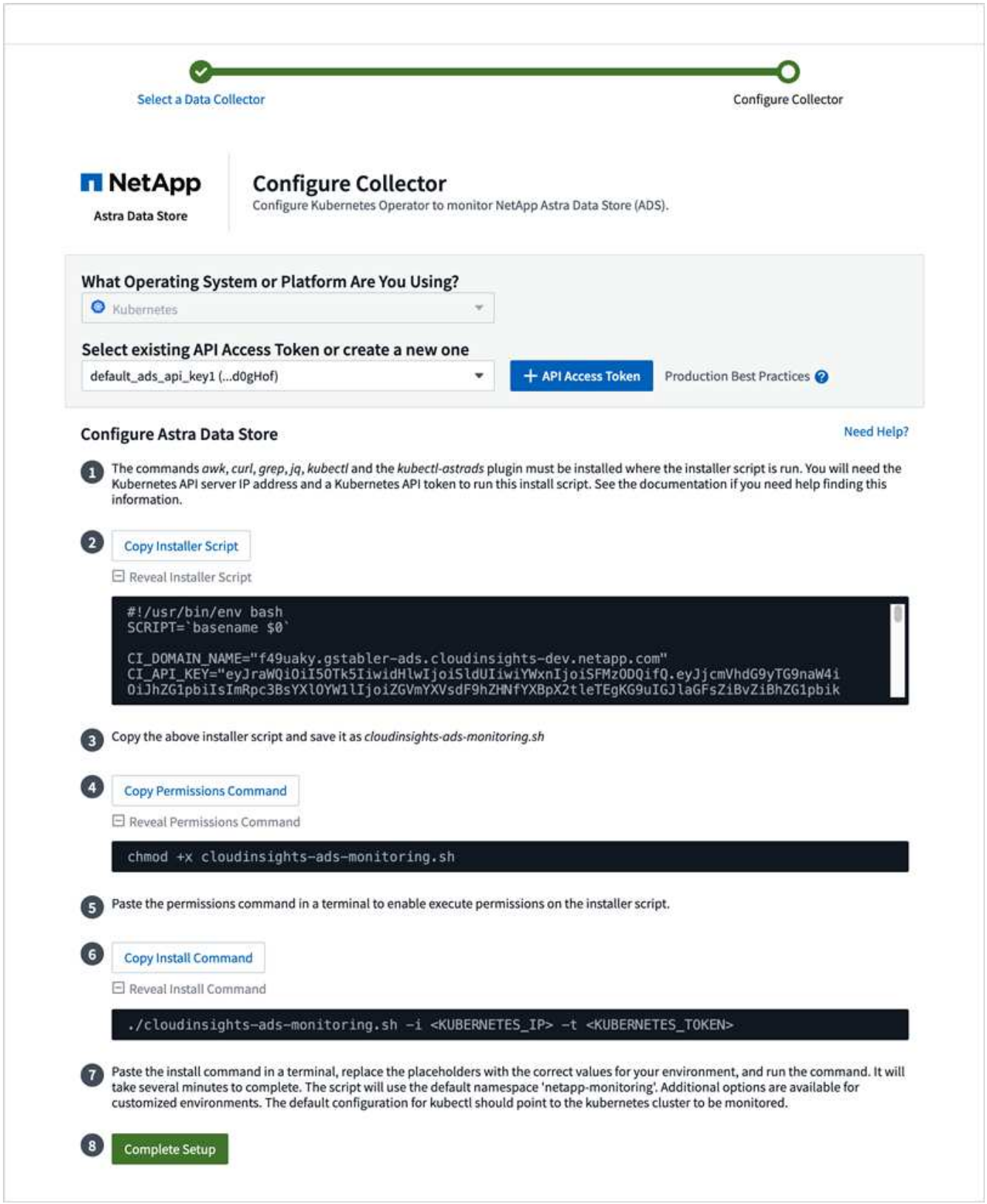
```
USAGE: cloudinsights-ads-monitoring.sh [OPTIONS]
Configure monitoring of Astra Data Store by Cloud Insights.
OPTIONS:
  -h                      Display this help message.
  -d ci_domain_name       Cloud Insights tenant domain name.
  -i kubernetes_ip        Kubernetes API server IP address.
  -k 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。



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 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: bzl9ngz.gst-adsdemo.ci-dev.netapp.com
    name: CI
  serviceAccount: sa-netapp-monitoring
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。

監控操作員會更新Fluent位元、以使用新Cloud Insights 的更新版的解決方案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（若已設定）和Fluent位元組態。

移除**Astra Data Store**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**（若已設定）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. 如果已設定Telegraf、請找出「Telegraf」區段、然後移除名為「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「Telewraf」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。

   監控操作員會更新Telegraf（若已設定）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（若已設定）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 thoughput 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 thoughput 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指標。您可以設定Prometheus從Astra Data Store 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叢集或其他可與Astra Data Store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 Kubernetes叢集通訊。如果未在Astra Data Store叢集上安裝Prometheus和Grafana、您必須確保它們能與Astra Data Store叢集上執行的度量服務通訊。

設定Prometheus

Astra Data Store在Kubernetes叢集中的TCP連接埠9341上提供度量服務。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「datasourses.yaml」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提供的度量類型、以及如何檢視這些資料。

步驟

1. 開啟Astra Data Store「.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 cluster custom resource (CR)]
- [Set up Cloud Insights]
- [Stream event logs to Elastic]。

在**Astra Data Store**叢集自訂資源（**CR**）中設定監控

如果尚未在Astra Data Store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" 適用於Astra Data Store。

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。

安全的**Astra**資料儲存區

管理安全性憑證

Astra Data Store在叢集的軟體元件之間使用相互傳輸層安全性（MTLS）加密。每個Astra Data Store叢集都有自我簽署的根CA憑證（「astrads-cert-root」）和中介CA憑證（「astrads-cert」（「叢集名稱」）。這些憑證

由Astra Data Store營運者管理；營運者會在每個憑證到期日前7天自動續訂。您也可以手動撤銷憑證。

撤銷憑證

如果Astra Data Store控制器、節點或CA憑證遭入侵、您可以刪除其MTLS機密來撤銷它。當您這麼做時、Astra Data Store營運者會自動發出新的憑證。您可以隨時撤銷Astra Data Store憑證。

如果您撤銷CA憑證、這會撤銷該CA所簽署的任何憑證。

步驟

1. 登入Astra Data Store叢集中的控制器節點。
2. 列出系統上現有的憑證。例如：

```
kubectl get secrets -n astrads-system | grep astrads-cert
```

輸出應類似於下列內容：

```
astrads-cert-astrads-cluster-controller
kubernetes.io/tls     4      6d6h
astrads-cert-astrads-cluster-f23d158
kubernetes.io/tls     4      6d6h
astrads-cert-astrads-ds-dms-astrads-cluster-f23d158
kubernetes.io/tls     4      6d6h
astrads-cert-astrads-ds-support-astrads-cluster-f23d158
kubernetes.io/tls     4      6d6h
astrads-cert-astrads-support-astrads-cluster-f23d158
kubernetes.io/tls     4      6d6h
astrads-cert-root
kubernetes.io/tls     4      6d6h
astrads-cert-sti-net-com
kubernetes.io/tls     5      6d6h
```

3. 在輸出中、記下您需要撤銷的憑證名稱。
4. 使用「kubecll」公用程式來撤銷憑證、並以憑證名稱取代「Certificate _name」（憑證名稱）。例如：

```
kubectl delete secret CERTIFICATE_NAME -n astrads-system
```

現有的憑證會被撤銷、並自動產生新的憑證。

管理外部金鑰

您可以使用一或多個外部金鑰管理伺服器來保護叢集用來存取加密資料的金鑰。外部金鑰管理伺服器是儲存環境中的第三方系統、使用金鑰管理互通性傳輸協定（KMIP）為節點提供金鑰。

Astra Data Store在建立Astra Data Store叢集時、預設會使用內部金鑰提供者啟用靜止軟體加密（sear）功能。

管理金鑰包括下列自訂資源定義（客戶需求日）：

- 適用**DSKeyProvider**：設定外部KMIP伺服器、此伺服器可以是伺服器叢集。
- 《**DSSEARKeyRotate**：從金鑰提供者取得新的金鑰加密金鑰、並提供給Astra Data Store。

您可以執行下列與外部金鑰管理相關的工作：

- [Set up external key management]
- [Check the software encryption at rest status]
- [Change external to internal key management]
- [Rotate keys for security]

設定外部金鑰管理

在Astra Data Store中設定外部金鑰管理時、會使用「kubectl astrads」命令。

您需要叢集或KMIP伺服器上的SSL憑證、才能設定外部金鑰、例如使用OpenSSL。

步驟

1. 準備金鑰提供者用戶端的憑證。包括用戶端憑證、用戶端私密金鑰及信任CA套裝組合。

   您將在叢集或KMIP伺服器上準備SSL憑證、以便設定外部金鑰、例如使用OpenSSL。

2. 登入Astra Data Store叢集中的其中一個節點。
3. 輸入下列kubecll副檔名命令、設定Astra Data Store叢集的金鑰提供者：

```
kubectl-astrads key-provider certs --key key.pem
--client-cert client_cert.pem --ca-cert server_ca.pem
--hostnames=<kmip_server_ip> <key_provider_cr_name>
--namespace astrads-system --cluster <ads_cluster_name>
```

下列範例會針對As叢集「astradse-Cluster-f23d158」設定名為「hashicorp」的外部金鑰提供者。

```
kubectl-astrads key-provider certs --key key.pem
--client-cert client_cert.pem --ca-cert server_ca.pem
--hostnames=10.235.nnn.nnn hashicorp
--namespace astrads-system --cluster astrads-cluster-f23d158
```

1. 將Astra Data Store叢集設定為使用外部金鑰管理程式、透過適用的適用項（適用）。顯示說明。

```
kubectl-astrads clusters sears -h
```

回應：

```
Configure SEARS in AstraDS cluster

Usage:
  astrads clusters sears [flags]

Flags:
  -d, --duration string   Duration for key rotation (default "2160h")
  -h, --help              help for sears


Global Flags:
      --ads-cluster-name string        Name of the ADS Cluster
      --ads-cluster-namespace string   Namespace of the ADS Cluster
      ...
```

下列命令可將Astra Data Store叢集設定為使用「適用的」「適用的DSKeyProvider hashicorp」做為sar的金鑰管理程式。命令也會使用按鍵旋轉時間、預設值為90天（2160小時）。

```
kubectl-astrads clusters sears -d 500h hashicorp
--ads-cluster-name=astrads-cluster-f23d158
--ads-cluster-namespace=astrads-system
```

檢查軟體加密的靜止狀態

您可以在閒置時檢查軟體加密的組態。

步驟

1. 檢查適用的電池。

```
Name:         astrads-cluster-f23d158
Namespace:    astrads-system
Labels:       <none>
Annotations:  <none>
API Version:  astrads.netapp.io/v1beta1
Kind:         AstraDSCluster
...
Spec:
...
  Software Encryption At Rest:
    Ads Key Provider:     hashicorp
    Key Rotation Period:  500h0m0s
...
Status:
...
  Software Encryption At Rest Status:
    Key Active Time:       2022-05-16T15:53:47Z
    Key Provider Name:     hashicorp
    Key Provider UUID:     ccfc2b0b-dd98-5ca4-b778-99debef83550
    Key UUID:              nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnnn
```

將外部變更為內部金鑰管理

如果您目前使用外部金鑰管理程式、可以將其變更為內部金鑰管理程式。

步驟

1. 移除SoftwareEncryptionAtRest組態、以變更適用的DSCluster CR。
2. （選用）刪除先前的適用的適用選項。

不會自動移除先前的金鑰提供者和密碼。

旋轉金鑰以確保安全性

金鑰輪替可強化安全性。依預設、Astra Data Store每90天自動旋轉金鑰一次。您可以變更預設設定。此外、您也可以視需要隨時旋轉按鍵。

設定自動金鑰旋轉

1. 更新CRD中的「適用」參數。

```
kubectl patch astradscluster astrads-cluster-f23d158
-n astrads-system
--type=merge -p '{"spec": {"softwareEncryptionAtRest": {
"keyRotationPeriod": "3000h"}}}'
```

設定隨需金鑰旋轉

1. 建立可旋轉金鑰的適用的適用選項：「Request CR」（建立適用的適用選項）。

```
cat << EOF | kubectl apply -f  -
apiVersion: astrads.netapp.io/v1beta1
kind: AstraDSSEARKeyRotateRequest
metadata:
  name: manual
  namespace: astrads-system
spec:
  cluster: astrads-cluster-f23d158
EOF
```

更新Astra Data Store授權

您可以更新Astra Data Store安裝的評估授權、以延長評估期間。您可以使用下列三種方法之一來更新授權：

- 若要使用Astra Control Center更新Astra Data Store授權、請參閱 "更新儲存後端授權"。
- 若要使用Astra VMware外掛程式更新Astra Data Store授權、請參閱 "使用VMware管理Astra資料儲存區"。
- 若要使用命令列更新Astra Data Store授權、請參閱 [Update the Astra Data Store license using the command line]。

使用命令列更新Astra Data Store授權

您可以使用「kubecll」公用程式來更新Astra Data Store授權。

步驟

1. 請套用您從NetApp取得的替換NetApp授權檔案（NLF）。在執行命令之前、請輸入叢集名稱（「<Astra Data-Store-cluster名稱>」）和授權檔案路徑（「<file_path/file.txt>'）：

```
kubectl astrads license add --license-file-path <file_path/file.txt>
--ads-cluster-name <Astra-Data-Store-cluster-name> -n astrads-system
```

2. 確認已新增授權：

```
kubectl astrads license list
```

您應該會看到類似下列的回應：

```
NAME         ADSCLUSTER                  VALID   PRODUCT
EVALUATION   ENDDATE     VALIDATED
p100000006   astrads-example-cluster     true    Astra Data Store    true
2023-01-23   2022-04-04T14:38:54Z
```

升級Astra Data Store

您可以升級Astra Data Store、以善用最新的功能與修正程式。您可以使用Astra Data Store「kubecll」擴充功能來升級Astra Data Store。

使用KECBECVL升級Astra資料儲存區

您可以使用Astra Data Store「kubecll」擴充功能來升級Astra Data Store。

下載Astra Data Store產品組合並擷取映像

步驟

1. 登入 "NetApp 支援網站" 並下載Astra Data Store套裝組合（「Astra Data_Store_2022.05.tar」）。
2. （可選）使用以下命令驗證套件的簽名：

```
openssl dgst -sha256 -verify Astra_Data_Store_2022.05.pub -signature
Astra_Data_Store_2022.05.sig 2022.12.01_ads.tar
```

3. 建立目錄：

```
mkdir Astra_Data_Store_2022.05
cd Astra_Data_Store_2022.05
```

4. 擷取影像：

```
tar -vxzf <path to tar file>/Astra_Data_Store_2022.05.tar
```

> 影像將擷取至工作目錄中建立的「astrads/images/」目錄。

複製二進位檔並將映像推送至本機登錄

步驟

1. 從您用來擷取映像的目錄、將Kubertl-astrads二進位檔複製到安裝Kubernetes Kubecl二進位檔的標準路徑（以下範例將使用「usr/bin/」作為路徑）。Kustbecl-astrads是自訂的Kvecll擴充功能、可安裝及管理Astra Data Store叢集。

使用「which kubecll」命令尋找安裝kubecl二進位檔的路徑。

```
cp -p .astrads/bin/kubectl-astrads /usr/bin/.
```

2. 將Astra Data Store映像目錄中的檔案新增至本機登錄。

請參閱以下自動載入影像的範例指令碼。

a. 登入您的登錄：

```
docker login [your_registry_path]
```

b. 將環境變數設為您要推送Astra Data Store映像的登錄路徑、例如「REpo.company.com`」。

```
export REGISTRY=repo.company.com/astrads
```

c. 執行下列指令碼、將影像載入Docker、標記影像、然後將影像推送到本機登錄：

```
for astraImageFile in $(ls astrads/images/*.tar) ; do
   astraImage=$(docker load --input ${astraImageFile} | sed 's~Loaded
image(s): ~~')
   astraImageShort=`echo $astraImage | sed 's~.*/~~'`
   docker tag ${astraImage} ${REGISTRY}/${astraImageShort}
   docker push ${REGISTRY}/${astraImageShort}
done
sed -i 's~\[YOUR REGISTRY\]~'${REGISTRY}'~'
./astrads/manifests/*.yaml
```

執行升級

步驟

1. 將「astradsoper.yaml」檔案複製到本機目錄：

```
cp /PATH/TO/FILE/astradsoperator.yaml ./
```

2. 升級營運者。以適合您環境的適當資訊取代大寫字母中的引數：

```
kubectl-astrads upgrade ads-operator --repository-url REPOSITORY_URL
--operator-yaml astradsoperator.yaml
```

3. 開始Astra Data Store升級。以適合您環境的適當資訊取代大寫字母中的引數：

```
kubectl-astrads upgrade ads-version --repository-url REPOSITORY_URL
--ads-version-yaml ./astrads/manifests/astradsversion.yaml
```

此時會出現一則訊息、通知您升級已開始、並需要幾分鐘時間才能完成。

使用自動指令碼解除安裝Astra Data Store

若要解除安裝Astra Data Store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、部署環境及Astra Data Store命名空間。

您可以使用不同的方法來解除安裝：

- [Uninstall Astra Data Store with an automated script]
- [Uninstall Astra Data Store manually without a script]
- [Troubleshoot the Astra Data Store uninstall process]

使用自動指令碼解除安裝Astra Data Store

此程序使用自動指令碼來解除安裝Astra Data Store。

您需要的是 **#8217** ；需要的是什麼

- root系統管理權限

Astra Data Store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]

移除現有的工作負載和繫結

在解除安裝Astra Data Store之前、您必須先移除下列項目

- 所有使用Astra Data Store做為儲存後端的應用程式工作負載
- Trident繫結使用Astra Data Store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、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~netappsdsoperator.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-astradsvolumefile-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolumefile-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「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、您可以解除安裝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。

無需指令碼即可手動解除安裝Astra Data Store

此程序可在不使用指令碼的情況下手動解除安裝Astra Data Store。

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

您需要的是 **#8217** ；需要的是什麼

- root系統管理權限

Astra Data Store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 cluster and control plane]
- [Delete the license]
- [Delete the Astra Data Store installation]
- [Validate the removal of the astrads-system namespace]
- [Ensure containers are not running on worker nodes]
- [Delete OpenShift Container Platform resources]

移除現有的工作負載和繫結

在解除安裝Astra Data Store之前、您必須先移除下列項目

- 所有使用Astra Data Store做為儲存後端的應用程式工作負載
- Trident繫結使用Astra Data Store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**叢集和控制面板

請依照下列步驟手動解除安裝Astra Data Store。

刪除磁碟區並匯出原則

刪除叢集之前、您應該先刪除Astra Data Store Volume及匯出原則。

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 Volume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**叢集

刪除叢集只會刪除Astra Data Store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。接著、Astra Data Store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授權：

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

刪除**Astra Data Store**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、您可以解除安裝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」。

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解除安裝程序疑難排解

如果您需要疑難排解解除安裝程序、請檢閱下列建議。

Pod處於終止狀態

Astra Data Store解除安裝程序偶爾會導致Pod在Kubernetes中維持終止狀態。

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
```

服務品質原則指向舊叢集

如果您只刪除Astra Data Store叢集並重新部署、可能無法建立持續磁碟區宣告（PVc）或磁碟區、因為服務品質（QoS）原則指向舊叢集、而且找不到。

1. 若要避免這種情況、請在刪除Astra Data Store叢集之後、手動刪除QoS原則：

```
kubectl delete AstraDSQosPolicy --all -A
```

2. 刪除整個Astra Data Store部署（不只是叢集）：

```
uninstall.sh all
```

刪除或解除安裝**Astra Data Store**之後、不會移除金鑰提供者**CRS**

如果外部金鑰提供者已針對正在刪除或解除安裝的Astra Data Store叢集進行設定、您可能需要手動清除任何未移除的金鑰提供者CR。

範例 **1.** 詳細資料

請使用下列因應措施指示：

步驟

1. 確認未移除金鑰提供者CRS：

```
kubectl get astradskeyprovider --selector
astrads.netapp.io/cluster=astrads-cluster-example -n astrads-system
```

回應：

```
NAME                    AGE
externalkeyprovider1    94s
```

2. 移除金鑰提供者CRS：
   a. 移除最終化工具：

```
kubectl edit astradskeyprovider -n astrads-system
```

   b. 移除下方反白顯示的最終化工具行：

```
kubectl edit astradskeyprovider externalkeyprovider1 -n astrads-
system
```

```
apiVersion: astrads.netapp.io/v1beta1
kind: AstraDSKeyProvider
metadata:
  creationTimestamp: "2022-05-24T16:38:27Z"
  finalizers:
  - astrads.netapp.io/astradskeyprovider-finalizer
  generation: 1
  labels:
    astrads.netapp.io/cluster: astrads-cluster-example
    astrads.netapp.io/rsid: "1"
  name: externalkeyprovider1
  namespace: astrads-system
  resourceVersion: "1134699"
  uid: a11111b2-31c0-4575-b7f3-97f9abla1bla
spec:
  cluster: astrads-cluster-example
  kmipServer:
    hostnames:
    - 10.xxx.xxx.xxx
    port: 5696
    secretRef: externalkeyprovider1
status:
  keyProviderUUID: a1b2cd34-4fc6-5bae-9184-2288c673181d
  kmipServerStatus:
    capabilities: '{ KMIP_library_version()=17367809,
KMIP_library_version_str()="KMIP
      1.9.3a  8-Apr-2019", KMIP_library_version_tag()="KMIP part
of KMIP 1.9.3a  8-Apr-2019",
      KMIP_library_is_eval()=false,
KMIP_library_fips_capable()=true(FIPS140),
KMIP_SSL_provider_build_version()=268444095,
      KMIP_SSL_provider_version()=268444095,
KMIP_SSL_provider_version_str()="OpenSSL
      1.0.2zb-fips  23 Sep 2021" }'
    keyServerUUID: 8422bdd0-74ad-579d-81bd-6d544ac4224a
```

c. 移除釋放器之後、請刪除金鑰提供者CR：

```
kubectl delete astradskeyprovider <key-provider-cr-name> -n
astrads-system
```

版權資訊

商標資訊