



Astra データストアを使用

Astra Data Store

NetApp
January 24, 2022

目次

Astra データストアを使用	1
kubectl コマンドを使用して Astra データストアのプレビューを管理	1
テストアプリケーションを展開します	7
クラスタを管理	11
Cloud Insights で指標を監視	24
イベントログを設定して監視する	36
Astra Control Center with Astra のデータストアプレビューを使用	37
Astra データストアプレビューをアンインストールします	38

Astra データストアを使用

kubectl コマンドを使用して Astra データストアのプレビューを管理

kubectl コマンドを使用し、Kubernetes API 拡張機能を使用して、Astra データストアのプレビューアセットを管理できます。

サンプルアプリの展開方法については、を参照してください ["テストアプリケーションを展開します"](#)。

次のクラスタのメンテナンス情報については、を参照してください ["クラスタを管理"](#)：

- ノードをメンテナンスモードにします
- ドライブを交換します
- ノードを追加します
- ノードを交換

必要なもの

- にインストールした Astra データストアプレビュー用 kubectl プラグイン ["Astra データストアプレビューをインストールします"](#)

Astra データストアプレビュー用の Kubernetes カスタム API リソースを列挙

Kubernetes の内部で kubectl コマンドを使用して、Astra データストアプレビュークラスタとやり取りし、状態を確認できます。

「api-resources」コマンドにリストされている各項目は、Kubernetes カスタムリソース定義（CRD）を表しています。CRD は、アストラデータストアプレビューがクラスタを内部的に管理するために使用します。

このリストは、後で示すように、各 Astra Data Store プレビューオブジェクトの短い名前を取得して入力を減らすのに特に役立ちます。

1. Kubernetes のカスタム API リソースである Astra データストアプレビュー版のリストを表示します。

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

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. Kubernetes クラスタ内の現在の Astra データストアプレビューオブジェクトをすべて取得するには、「`kubectl get ads-a`」コマンドを使用します。

```
bash-4.2$ kubectl get ads -A
```

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/	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. 短縮名の 1 つを使用して、クラスタ内のボリュームの現在の状態を表示します。

```
bash-4.2$ kubectl get adsvo -A
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
```

kubectl 拡張子の help オプションを使用します

'kubectl astras コマンド'には '-h' スイッチが含まれており ' 便利な使用法とフラグ・ドキュメントが提供されています

1. Astra Data Store preview kubectl 拡張機能のすべてのコマンドのヘルプを表示します。

```
bash-4.2$ kubectl astrads -h
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
(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. コマンドの詳細については 'astrads [command]--help' を使用してください

```
# Show help for a specific astrads command

bash-4.2$ kubectl astrads asup collect --help
WARNING: YOU ARE USING A DEV RELEASE
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 (default "manual")	Specify the callhome event to trigger.
-f, --forceUpload	Configure an AutoSupport to upload if it is in the compressed state and not uploading because it was created with 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 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.


```
time format.

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")
```

テストアプリケーションを展開します

Astra データストアプレビューで利用できるテストアプリケーションを導入する手順は次のとおりです。

この例では、Helm リポジトリを使用して Bitnami から MongoDB チャートを導入します。

必要なもの

- Astra データストアプレビュークラスタの導入と構成
- Trident のインストールが完了しました

手順

1. Bitnami から Helm repo を追加します。

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

2. MongoDB の導入

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

3. MongoDB ポッドのステータスを確認します。

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

4. MongoDB で使用した Persistent Volume Claim (PVC ; 永続ボリューム要求)を確認します。

```
~% 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. kubectl コマンド 'get astradsvolume' を使用して 'ボリュームを一覧表示します

```
~% 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. kubectl コマンド 'describe astradsvolume' を使用して 'ボリュームを説明します

```
~% 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

```

クラスタを管理

クラスタを管理するには、`kubectl` コマンドを Astra データストアプレビューで使います。

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

必要なもの

- `kubectl` および `kubectl-astras` プラグインがインストールされたシステム。を参照してください "[Astra データストアプレビューをインストールします](#)"。

ノードをメンテナンスモードにします

ホストのメンテナンスやパッケージのアップグレードが必要な場合は、ノードをメンテナンスモードにする必要があります。



ノードは、Astra データストアプレビュークラスタにすでに含まれている必要があります。

ノードが保守モードのときは、クラスタにノードを追加できません。この例では、`nhcitj1525` をメンテナンスモードにします。

手順

1. ノードの詳細を表示します。

```
~% kubectl get nodes
>> Show the output of 'kubectl get nodes'.

[root@scs000039783-1 ~]# kubectl get nodes
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. ノードがまだメンテナンスモードになっていないことを確認します。

```
~% kubectl astrads maintenance list
NAME      NODE NAME  IN MAINTENANCE  MAINTENANCE STATE
MAINTENANCE VARIANT
kubectl astrads      maintenance      create node4 --node
-name="nhcitjj1525" --variant=Node

Maintenance mode astrads-system/node4 created
```

3. メンテナンスモードを有効にします。

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

4. ノードをリストします。

```
~% kubectl astrads nodes list
NODE NAME      NODE STATUS    CLUSTER NAME
nhcitjj1525    Added          ftap-astra-012
nhcitjj1527    Added          ftap-astra-012
nhcitjj1526    Added          ftap-astra-012
nhcitjj1528    Added          ftap-astra-012
...
```

5. メンテナンスモードのステータスを確認します。

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

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

保守モードの場合は 'false' として起動し 'true' に変更します「保守状態」が「準備期間の保守」から「ReadyforMaintenance」に変更されます。

6. ノードのメンテナンスが完了したら、メンテナンスモードを無効にします。

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

7. ノードが保守モードでなくなったことを確認します。

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

ノードを追加します

追加するノードは Kubernetes クラスタに含まれ、クラスタ内の他のノードと同様の設定である必要があります。

手順

1. 新しいノードのデータ IP がまだ astradscluster CR に含まれていない場合は、次の手順を実行します。
 - a. astadscluster CR を編集し、ADS Data Networks Addresses フィールドに追加の dataIP を追加します。

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

```
ADS Data Networks:
  Addresses:  dataIP1, dataIP2, dataIP3, dataIP4, *newdataIP*
```

- b. CR ファイルを保存します。
- c. Astra データストアプレビュークラスタにノードを追加します。

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

2. それ以外の場合は、ノードを追加するだけで済みます。

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

3. ノードが追加されたことを確認します。

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

ノードを交換

クラスタ内の障害が発生したノードを交換するには、kubectl コマンドを Astra Data Store preview とともに使用します。

手順

1. すべてのノードを一覧表示します。

```
~% kubectl astrads nodes list
NODE NAME          NODE STATUS   CLUSTER NAME
sti-rx2540-534d... Added        cluster-multinodes-21209
sti-rx2540-535d... Added        cluster-multinodes-21209
...
```

2. クラスタについて説明します。

```
~% kubectl astrads clusters list
CLUSTER NAME          CLUSTER STATUS  NODE COUNT
cluster-multinodes-21209  created         4
```

3. 障害が発生したノードで 'Node HA' が 'false' としてマークされていることを確認します


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

```
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. クラスター CR を変更して障害ノードを削除します。ノード数が3つ減ります。

```
# cat manifests/astradscluster.yaml
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

[root@scspr2409016001 42733317_42952507_1x5Node_Astra_DAS-002]# cat
manifests/astradscluster.yaml
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] Specify the network interface names for either all or
  # none
  adsNetworkInterfaces:

```

```

managementInterface: "mgmt"
clusterInterface: "data"
storageInterface: "data"

# [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
kubect1 apply -f manifests/astradscluster.yaml
astradscluster.astrads.netapp.io/cluster-multinodes-21209 configured

```

5. ノードがクラスタから削除されたことを確認します。

```

~% kubect1 get nodes --show-labels
NAME                                STATUS  ROLES                                AGE    VERSION
LABELS
sti-astmaster-237                   Ready   control-plane,master                24h    v1.20.0
sti-rx2540-532d                    Ready   <none>                              24h    v1.20.0
sti-rx2540-533d                    Ready   <none>                              24h

~% kubect1 astrads nodes list
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

~% kubect1 get nodes --show-labels
NAME                                STATUS  ROLES                                AGE    VERSION
LABELS
sti-astmaster-237                   Ready   control-plane,master                24h
sti-rx2540-532d                    Ready   <none>                              24h

~% kubect1 describe astradscluster -n astrads-system
Name:                  cluster-multinodes-21209
Namespace:             astrads-system
Labels:                <none>
Kind:                  AstraDSCluster
Metadata:
...

```

6. クラスタ CR を変更して、交換用のノードをクラスタに追加します。ノード数は 4 に増えます。新しいノードが追加対象として選択されていることを確認します。

```

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

~% kubectl get pods -n astrads-system
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   0           24h

~% kubectl astrads nodes list
NODE NAME                NODE STATUS    CLUSTER NAME
sti-rx2540-534d...       Added          cluster-multinodes-21209
sti-rx2540-535d...       Added          cluster-multinodes-21209

~% kubectl astrads clusters list
CLUSTER NAME                CLUSTER STATUS  NODE COUNT
cluster-multinodes-21209    created         4

~% kubectl astrads drives list
DRIVE NAME    DRIVE ID    DRIVE STATUS    NODE NAME    CLUSTER NAME
scsi-36000..  c3e197f2... Active        sti-rx2540... cluster-
multinodes-21209

```

ドライブを交換します

クラスタ内のドライブで障害が発生した場合は、データの整合性を確保するために、できるだけ早くドライブを交換する必要があります。ドライブで障害が発生すると、クラスタの CR ノードステータス、クラスタの健全性状態情報、および指標エンドポイントに、障害が発生したドライブの情報が表示されます。

nodeStatus.driveStatuses で障害が発生したドライブを示すクラスタの例

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

障害が発生したドライブ CR は、障害が発生したドライブの UUID に対応する名前でクラスタ内に自動的に作成されます。

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

...
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
```


手順

1. 交換可能なドライブを 'kubectl astrads show-replacements' コマンドで一覧表示しますこのコマンドは '交換の制限に適合するドライブをフィルタリングします (クラスタ内では未使用 'マウントなし' パーティションなし' または障害が発生したドライブ以上)

可能な交換用ドライブをフィルタリングせずにすべてのドライブを一覧表示するには '--all' を 'show-replacements' コマンドに追加します

```
~% kubectl astrads faileddrive list --cluster arda-6e4b4af
NAME          NODE                                CLUSTER      STATE
AGE
6000c290      sti-rx2540-300b.lab.netapp.com     ard-6e4b4af
ReadyToReplace 13m

~% kubectl astrads faileddrive show-replacements --cluster ard-6e4b4af
--name 6000c290
NAME  IDPATH          SERIAL  PARTITIONCOUNT  MOUNTED  SIZE
sdh   /scsi-36000c29417 45000c  0                false    100GB
```

2. パスしたシリアル番号でドライブを交換するには 'replace' コマンドを使用しますコマンドは置換を完了するか '--wait' 時間が経過すると失敗します

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

3. kubectl の astrads faileddrive replace' が不適切なシリアル番号を使用して実行された場合 '次のようなエラーが表示されます

```
~% 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'}]}
```

4. ドライブ交換を再実行するには ' 前のコマンドで --force' を使用します

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

を参照してください。

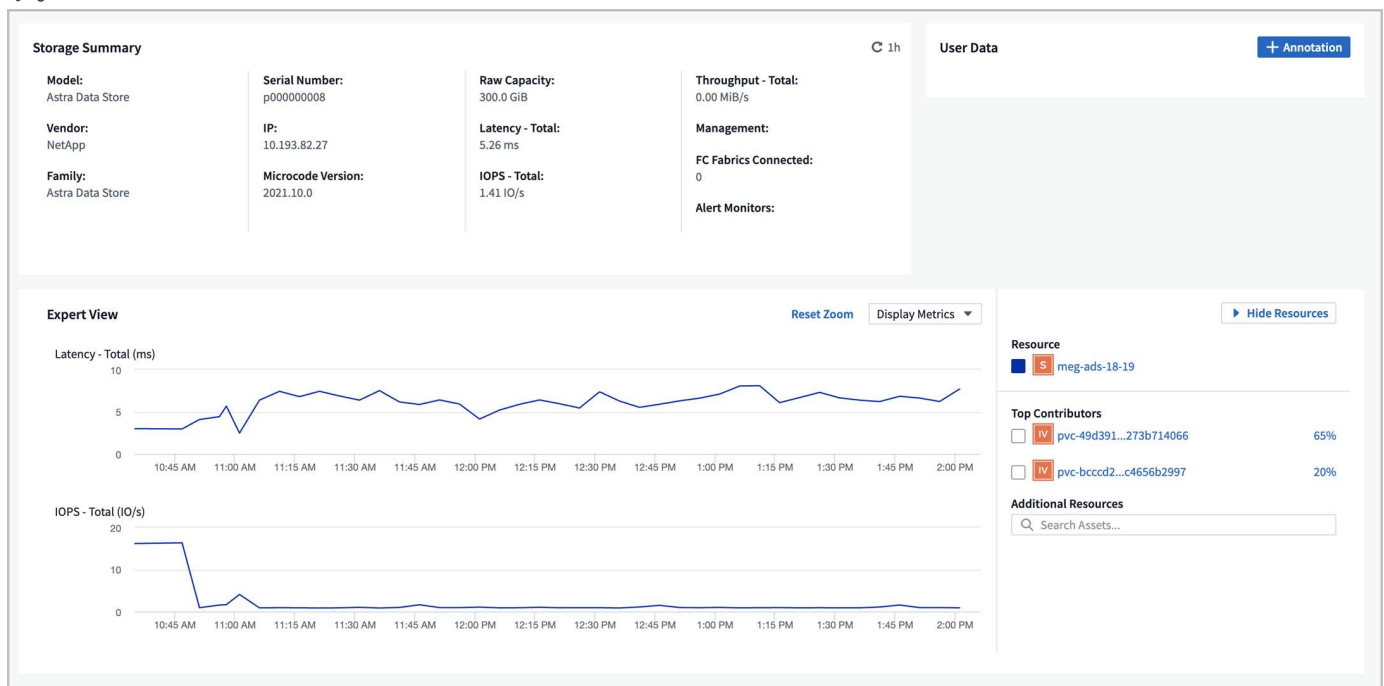
- ["kubectl コマンドを使用して Astra データストアのプレビューを管理"](#)

Cloud Insights で指標を監視

Cloud Insights を使用して、Astra データストアのプレビュー指標を監視できます。

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

Cloud Insights に表示される Astra データストアプレビュー指標の例を次に示します。



を使用して、Astra データストアのプレビューで生成された指標のリストを表示することもできます [\[Open Metrics API help\]](#)。

Cloud Insight の接続の前提条件となるタスクを実行

Cloud Insights を使用して Astra データストアに接続する前に、次の作業が必要です。

- ["Astra Data Store Monitoring Operator をインストールします"](#) これは、Astra Data Store プレビューインストール手順の一部です。
- ["kubectl-astras バイナリをインストールします"](#) これは、Astra Data Store プレビューインストール手順の一部です。
- ["Cloud Insights アカウントを作成します"](#)。
- 'awk'、'curl'、'grep' および 'jq' の各コマンドが使用可能であることを確認します

次の情報を収集します。

- * Cloud Insights API キー *。各カテゴリに対する読み取り / 書き込み権限：Acquisition Unit、Data Collection、Data Ingestion、Log Ingestionこれは、読み取り / 書き込み処理、Acquisition Unit のセットアップ、およびデータの取り込みプロセスのセットアップに使用されます。
- * Kubernetes API サーバの IP アドレスとポート *。Astra データストアプレビュークラスタを監視するために使用します。
- * Kubernetes API トークン *。これは Kubernetes API を呼び出すために使用されます。
- * 永続ボリューム構成 *。永続ボリュームのプロビジョニング方法に関する情報。詳細については、以下の「Acquisition Unit」を参照してください。

インストールスクリプトをダウンロードして実行します

Cloud Insights では、Bash スクリプトを使用して Astra データストアのプレビュー監視を監視オペレータから有効にすることができます。インストールスクリプトは、Astra Data Store コレクタ、Telegraf Agent、および Fluent Bit Agent を備えた Acquisition Unit をインストールします。

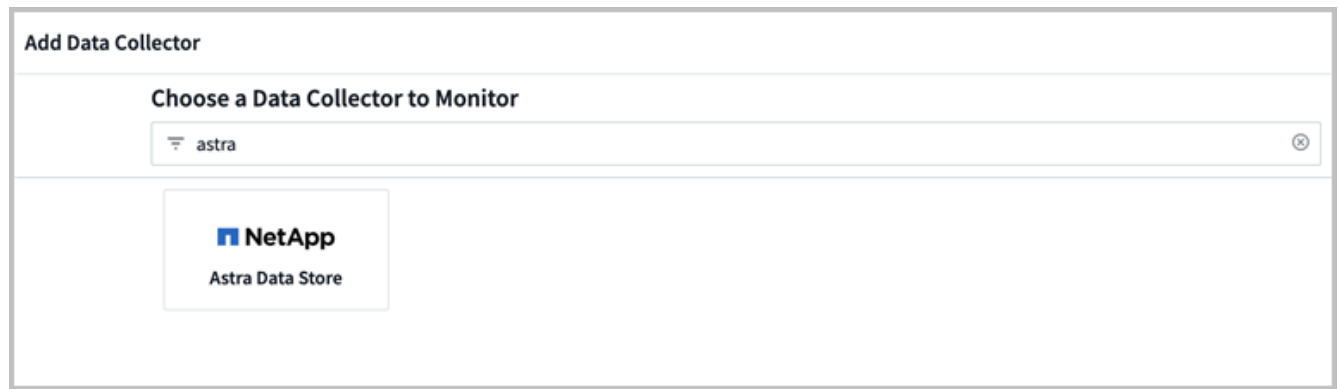
Cloud Insights テナントのドメイン名と選択した API アクセスキーは、ダウンロード時にインストーラスクリプトに組み込まれます。

その後、次のように指標が送信されます。

- Telegraf は指標を Cloud Insights データレイクに送信します。
- FLUENT ビットは、ログ取り込みサービスにログを送信します。

手順

1. Cloud Insights アカウントがない場合は作成します。
2. Cloud Insights にログインします。
3. Cloud Insights メニューから、**Admin>*Data Collector*** をクリックします。
4. 「* + Data Collector *」をクリックして、新しいコレクタを追加します。



5. 「* アストラデータストア *」 タイルをクリックします。
6. 正しい API アクセストークンを選択するか、新しいトークンを作成します。
7. 指示に従って、インストーラスクリプトをダウンロードし、権限を更新し、スクリプトを実行します。

このスクリプトには、Cloud Insights テナントの URL と選択した API アクセストークンが含まれます。



Kubernetes

default_ads_api_key1 (...d0gHof) ▼

+ API Access Token

Production Best Practices ?

[Need Help?](#)

- ## 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="eyJyJWQwOiIiOTk5IiwidHlwIjoiaSdUIiwiaWxnIjoisiFMzODQifQ.eyJjcVhdG9yTG9naW4iOiJhZGipbikImRmcc3B5YXl0YW1lIjoizGVmYXVsZF9hZHNFYXBxMjtleTEgK69uIGJlaGF5ZiBvZiBhZGipbik
```

- #### 4 Copy Permissions Command

[-] Reveal Permissions Command

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

- 6 Copy Install Command

[-] Reveal Install Command

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

- ## 8 Complete Setup

インストールスクリプトが完了すると、「データソース」リストに「Astra Data Store」コネクタが表示されます。



エラーが原因でスクリプトが終了した場合は、エラーが解決したら再実行できます。デフォルトの設定を使用しない環境では、Monitoring Operator 名前空間や Kubernetes API サーバポートなどの追加のパラメータがサポートされます。「-h」オプションを使用して、使用法とヘルプテキストを表示します。

インストールスクリプトを実行すると、次のような出力が生成されます。

```
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!
```

Acquisition Unit のストレージ

Acquisition Unit には、インストールファイル、設定データ、およびログを格納するために 3 つの Persistent Volume が必要です。監視オペレータは、デフォルトのストレージクラスを使用して、永続ボリュームに関する請求を作成します。インストーラ・スクリプトの実行時に '-s' オプションを使用して '別のストレージ・クラス名を指定できます

Kubernetes クラスタにストレージプロビジョニングツール（NetApp Trident など）がない場合は、インストーラスクリプトの実行時に '-r' オプションを使用してローカルファイルシステムパスを提供できます。'-r' オプションが設定されている場合 'インストーラ・スクリプトは '指定されたディレクトリ内に 3 つの永続ボリュームを作成しますこのディレクトリには、150GB 以上の空きスペースが必要です。

エージェント CR の例

以下に、インストーラスクリプトの実行後の「Monitoring - NetApp」エージェントの CR の例を示します。

```
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

```

インストーラのスクリプトヘルプ

インストーラスクリプトの完全なヘルプテキストを次に示します。

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

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 Key.
  -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.
```

Cloud Insights 接続を編集します

Kubernetes API キーまたは Cloud Insights API キーは、あとから編集できます。

- Kubernetes API キーを更新する場合は、Cloud Insights UI から Astra データストアコレクタを編集する必要があります。
- テレメトリとログに使用される Cloud Insights API キーを更新する場合は、`kubectl` コマンドを使用して Monitoring Operator CR を編集する必要があります。

Kubernetes API トークンを更新します

1. Cloud Insights にログインします。
2. **[Admin>]** > **[* Data Collectors]** を選択して、**[Data Collectors]** ページにアクセスします。
3. Astra データストアクラスターのエントリを探します。
4. ページの右側にあるメニューをクリックし、「* 編集 *」を選択します。

Cloud Insights API アクセストークンを更新します

1. Cloud Insights にログインします。
2. **[Admin>*API Access*]** を選択し、**[*+API アクセストークン*]** をクリックして、新しい Cloud Insights API アクセストークンを作成します。
3. エージェント CR を編集します。

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

4. 「output-sink」セクションを探し、「ci」という名前のエントリを見つけます。
5. ラベル「api-key」の場合は、現在の値を新しいAPI キーに置き換えます。

セクションは次のようになります。

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

6. エディタウィンドウを保存して終了します。

モニタリングオペレータは、Telegraf ビットと Fluent ビットを更新して、新しいAPI キーを使用します。

Cloud Insights から切断します

Cloud Insights から切断するには、最初に Cloud Insights UI から Astra データストアコレクタを削除する必要があります。これが完了したら、モニタリングオペレータから Acquisition Unit、Telegraf、および Fluent の各ビット設定を削除できます。

Astra Data Store コレクタを削除

1. Cloud Insights にログインします。
2. **[Admin>] > [* Data Collectors]** を選択して、**[Data Collectors]** ページにアクセスします。
3. Astra データストアクラスタのエントリを探します。
4. 画面の右側にある kebab メニューを選択し、*** Delete ***（削除）を選択します。
5. 確認ページで *** Delete *** をクリックします。

Acquisition Unit、Telegraf、および Fluent ビットを削除します

1. エージェント CR を編集します。

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

2. 「au」セクションを探し、「IsEnabled:false」を設定します
3. 「FLUENT ビット」セクションを探し、「ADS テール CI」という名前のプラグインを削除します。プラグインがない場合は、「FLUENT - BIT」セクションを削除できます。
4. 「テレグラム」セクションを探し、「ads-open-metric」という名前のプラグインを削除します。プラグインがない場合は、「テレグラム」セクションを削除できます。
5. 「output-sink」セクションを探し、「ci」という名前のシンクを取り外します。
6. エディタウィンドウを保存して終了します。

モニタリングオペレータが Telegraf および Fluent ビット設定を更新し、Acquisition Unit ポッドを削除します。

7. ストレージプロビジョニング担当者ではなく Acquisition Unit PVS にローカルディレクトリを使用した場合は、PVS を削除します。

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

次に、AU を実行していたノードの実際のディレクトリを削除します。

8. Acquisition Unit ポッドが削除されたら、Cloud Insights から Acquisition Unit を削除できます。
 - a. Cloud Insights メニューで、**Admin**>*Data Collector* を選択します。
 - b. [* Acquisition Units * (Acquisition Unit *)] タブをクリックします。
 - c. Acquisition Unit ポッドの横にあるメニューをクリックします。
 - d. [削除 (Delete)] をクリックします。

モニタリングオペレータが Telegraf および Fluent ビット設定を更新し、Acquisition Unit を削除します。

Open Metrics API のヘルプを参照してください

Astra データストアプレビューから指標を収集するために使用できる API のリストを次に示します。

- 「help」行は指標を表します。
- 「type」行は、メトリックがゲージかカウンタかを示します。

```
# 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_scraper_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

```

イベントログを設定して監視する

Event Management System（EMS；イベント管理システム）ログを監視するには、次の高度なタスクを実行します。

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

Astra データストアプレビュークラスタカスタムリソース（CR）で監視を設定する

Astra Data Store プレビュークラスタ CR でモニタリングオプションが設定されていない場合は、「astras」拡張機能を使用して設定できます。

入力するコマンド

```

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

```

ここで、

- インストールされているエージェントの名前空間：監視エージェントです。監視オペレータの CR です。
- -r はオプションです

Cloud Insights をセットアップする

を参照してください ["NetApp Cloud Insights のセットアップ方法"](#) Astra データストアプレビュー版で使用でき

ます。

イベントログを **Elastic** にストリーミングする

EMS イベントやその他のポッドログをストリーミングして、ログを Elastic などのサードパーティのエンドポイントにストリーミングするには、「astrads」拡張機能を使用します。

入力するコマンド

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



Elastic ホスト名は IP アドレスでもかまいません。

Astra Control Center with Astra のデータストアプレビューを使用

Astra Control Center のユーザーインターフェース（UI）を使用して、Astra データストアのプレビュータスクを実行できます。

Astra Control Center for Astra のデータストアプレビューのセットアップ

Astra Data Store プレビュー用の Astra Control Center UI を使用するには、次のタスクを実行する必要があります。

- クラスタを追加Astra Data Store プレビューがインストールされた、基盤となる Kubernetes クラスタです。を参照してください ["Astra データストアプレビュークラスタを Astra Control Center インストールにインポートします"](#)。
- Astra データストアプレビュー用のストレージバックエンドをクラスタに追加します。



ストレージバックエンドを追加し、Astra データストアプレビューを含む Kubernetes クラスタがない場合は、最初にクラスタを追加する必要があります。

Astra Control Center でできること

Astra Control Center for Astra Data Store プレビューをセットアップしたら、Astra Control Center UI を使用して次のタスクを実行できます。

- ["Astra Control Center を使用して、Astra データストアのプレビュー資産の健全性を監視"](#)。
- ["Astra データストアのプレビューバックエンドストレージを管理"](#)。
- ["ノード、ディスク、および永続ボリューム要求（PVC）とその状態を表示します"](#)。

を参照してください。

- ["Astra ファミリーの紹介"](#)

- ["Astra Control Center のドキュメント"](#)
- ["Astra Control API を使用"](#)

Astra データストアプレビューをアンインストールします

Astra データストアプレビューをアンインストールするには、ワークロード、バインド、ボリューム、エクスポートポリシー、クラスタ、ライセンス、導入環境、Astra データストアプレビューネームスペース。

必要なもの

- root 管理権限

Astra Data Store プレビューアンインストールプロセスでは、次の手順が簡単に実行できます。

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

既存のワークロードとバインドを削除します

Astra データストアプレビューをアンインストールする前に、次の項目を削除する必要があります

- ストレージバックエンドとして Astra データストアプレビューを使用するすべてのアプリケーションワークロード
- バックエンドとして Astra データストアプレビューを使用する Trident バインディング

これにより、Kubernetes 環境をクリーンな状態のまま維持できます。これは、を再インストールする場合に重要です。

Astra データストアプレビュークラスタとコントロールプレーンをアンインストールします

Astra Data Store プレビューを手動でアンインストールするには、次の手順に従います。

ボリュームとエクスポートポリシーを削除します

クラスタを削除する前に、Astra データストアプレビューボリュームとエクスポートポリシーを削除する必要があります。



ボリュームとエクスポートポリシーを最初に削除しないと、Astra Data Store のプレビューボリュームオブジェクトが削除されるまで、クラスタの削除プロセスは一時停止します。クラスタの削除を開始する前に、それらの項目を削除の方が効率的です。

手順

1. ボリュームを削除します。

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

2. エクスポートポリシーを削除します。

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

Astra データストアプレビュークラスタを削除

クラスタを削除すると、Astra Data Store preview cluster object custom resource（CR）とクラスタを対象としたリソースのみが削除される。



オペレータ、nodeinfo ポッド、およびクラスタコントローラ（Kubernetes を対象としたリソース）は、クラスタを削除しても削除されません。

クラスタを削除すると '基盤となるオペレーティング・システムもノードからアンインストールされ 'FIRETAAP' および netwd' サービスが停止します

アンインストーラが完了するまでに約 1 分かかります。次に、Astra データストアのプレビュー用リソースの削除が、クラスタを対象として開始します。

1. クラスタを削除します。

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

ライセンスを削除します

1. クラスタ内の各ワーカー・ノードに SSH 接続し 'ワーカー・ノードで 'FIRETAAP' または 'netwd' が実行されていないことを確認します
2. Astra データストアプレビューライセンスを削除します。

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

Astra データストアプレビューインストールを削除します

クラスタ内のコントローラ、演算子、ネームスペース、およびサポートポッドを削除します。

1. Astra Data Store プレビューインストールオブジェクトを削除します。

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

2. DemonSets データストアと Astra データストアのすべてのプレビューコントローラリソースを削除します。

```
~% 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. 残りのアーティファクトと演算子 YAML ファイルを削除します。

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

astrs-system 名前空間の削除を検証します

次のコマンドで結果が返されないことを確認します。

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

ワーカーノードでコンテナが実行されていないことを確認します

'FIRETAAP' や 'netwd' などのコンテナがワーカー・ノードで実行されていないことを確認します各ノードで次のコマンドを実行します。

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

OpenShift Container Platform リソースを削除します

Red Hat OpenShift Container Platform (OCP) に Astra Data Store preview をインストールした場合は、OCP セキュリティコンテキスト制約 (SCC) と役割バインディングリソースをアンインストールできます。

OpenShift では、セキュリティコンテキスト制約 (SCC) を使用して、ポッドで実行できるアクションを制御します。

標準のアンインストールプロセスが完了したら、次の手順を実行します。

1. SCC リソースを削除します。

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

2. ロールバインドリソースを削除します

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



これらの手順で「resources not found errors」を無視します。

3. すべての Kubernetes ノードから /var/lib/kubelet/config.yaml を削除します

手動削除のサンプル

次に、手動アンインストールスクリプトの実行例を示します。

```
$ 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

```

Copyright Information

Copyright © 2022 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system-without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark Information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.