



# 使用 **Astra** 数据存储

## Astra Data Store

NetApp  
January 27, 2022

# 目录

- 使用 Astra 数据存储 ..... 1
  - 使用 kubectl 命令管理 Astra Data Store 预览资产 ..... 1
  - 部署测试应用程序 ..... 7
  - 管理集群 ..... 11
  - 使用 Cloud Insights 监控指标 ..... 23
  - 配置和监控事件日志 ..... 35
  - 将 Astra 控制中心与 Astra Data Store 预览结合使用 ..... 36
  - 卸载 Astra Data Store 预览版 ..... 36

# 使用 Astra 数据存储

## 使用 kubectl 命令管理 Astra Data Store 预览资产

您可以使用 kubectl 命令和 Kubernetes API 扩展来管理 Astra Data Store 预览资产。

要了解如何部署示例应用程序，请参见 ["部署测试应用程序"](#)。

有关以下集群维护信息，请参见 ["管理集群"](#)：

- 将节点置于维护模式
- 更换驱动器
- 添加节点
- 更换节点

您需要什么？ [#8217](#) ；将需要什么

- 您安装在中的 Astra Data Store 预览 kubectl 插件 ["安装 Astra Data Store 预览版"](#)

## 列出适用于 Astra Data Store 预览版的 Kubernetes 自定义 API 资源

您可以在 Kubernetes 中使用 kubectl 命令与 Astra Data Store 预览集群进行交互并观察其状态。

从 `api-resources` 命令中列出的每个项目都代表一个 Kubernetes 自定义资源定义（CRD），Astra Data Store 预览版可在内部使用该定义定义（CRD）来管理集群。

此列表对于获取每个 Astra Data Store 预览对象的短名称以减少键入效果特别有用，如后面所示。

1. 显示用于 Astra Data Store 预览的 Kubernetes 自定义 API 资源列表：

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

响应：

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. 要获取 Kubernetes 集群中所有当前的 Astra Data Store 预览对象，请使用 `kubectl get ADS -a` 命令：

```
kubectl get ads -A
```

响应：

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. 使用以下短名称之一显示集群中卷的当前状态：

```
kubectl get adsvo -A
```

响应：

NAMESPACE	NAME	SIZE	IP	CLUSTER
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 **kubectl** 扩展上的 **help** 选项

`kubectl astrad` 命令包含一个 `-h` 交换机，可为您提供使用情况和标志文档。

### 1. 显示有关 Astra Data Store preview `kubectl` 扩展中所有命令的帮助：

```
kubectl astrads -h
```

响应：

```
A kubectl plugin for inspecting your AstraDS deployment

Usage:
  astrads [command]

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

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

operation, this flag can be

groups.

<code>--cache-dir string</code>	Default HTTP cache directory (default <code>"/u/arda/.kube/http-cache"</code> )
<code>--certificate-authority string</code>	Path to a cert file for the certificate authority
<code>--client-certificate string</code>	Path to a client certificate file for TLS
<code>--client-key string</code>	Path to a client key file for TLS
<code>--cluster string</code>	The name of the kubeconfig cluster to use
<code>--context string</code>	The name of the kubeconfig context to use
<code>-h, --help</code>	help for astrads
<code>--insecure-skip-tls-verify</code>	If true, the server's certificate will not be checked for validity. This will make your HTTPS connections insecure
<code>--kubeconfig string</code>	Path to the kubeconfig file to use for CLI requests.
<code>-n, --namespace string</code>	If present, the namespace scope for this CLI request
<code>--request-timeout string</code>	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).
<code>-s, --server string</code>	A value of zero means don't timeout requests. (default <code>"0"</code> )
<code>--token string</code>	The address and port of the Kubernetes API server
<code>--user string</code>	Bearer token for authentication
	The name of the kubeconfig user to use

2. 有关命令的详细信息, 请使用 `astrad [command] -help`。

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

响应:

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

Usage:

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

Examples:

```
# Control plane collection
```

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

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

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

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

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

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

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

NOTE:

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

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

Flags:

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



```

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

```

## 部署测试应用程序

以下是部署可与 Astra Data Store 预览版结合使用的测试应用程序的步骤。

在此示例中，我们使用 Helm 存储库从 BitNami 部署 MongoDB 图表。

您需要什么？ **#8217** ；将需要什么

- 部署和配置了 Astra Data Store 预览集群
- Trident 安装已完成

步骤

1. 从 BitNami 添加 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. 检查 MongoDB POD 的状态:

```

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

```

#### 4. 验证 MongoDB 使用的永久性卷声明（PVC）：

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

#### 5. 使用 kubectl 命令 get astradsvolume 列出卷：

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

#### 6. 使用 kubectl 命令 describe astradsvolume 描述卷：

```
~% 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

```

## 管理集群

您可以使用带有 Astra Data Store 预览版的 `kubectl` 命令来管理集群。

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

您需要什么？ **#8217** ；将需要什么

- 安装了 `kubectl` 和 `kubectl-astrads` 插件的系统。请参见 ["安装 Astra Data Store 预览版"](#)。

## 添加节点

要添加的节点应属于 Kubernetes 集群，并且其配置应与集群中的其他节点类似。

### 步骤

1. 如果新节点的 `dataIP` 尚未加入 `ADSCluster` CR，请执行以下操作：
  - a. 编辑 `astradscluster` CR 并在 `ADS Data Networks Addresses` 字段中添加额外的 `dataIP`：

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

响应:

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

- a. 保存 CR 文件。
- b. 将节点添加到 Astra Data Store 预览集群:

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

2. 否则, 只需添加节点:

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

3. 验证是否已添加此节点:

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

## 将节点置于维护模式

需要执行主机维护或软件包升级时, 应将节点置于维护模式。



此节点必须已属于 Astra Data Store 预览集群。

当节点处于维护模式时, 您无法向集群添加节点。在此示例中, 我们会将节点 `nhcitj1525` 置于维护模式。

### 步骤

1. 显示节点详细信息:

```
~% kubectl get nodes
```

响应:

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. 确保节点尚未处于维护模式：

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

响应（没有节点处于维护模式）：

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

## 3. 启用维护模式。

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

示例：

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

## 4. 列出节点。

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

响应：

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

## 5. 检查维护模式的状态：

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

响应：

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

维护`模式下的 将以 `false 开头, 并更改为 true。M状态 从 PreparingForMaintenance 更改为 ReadyforMaintenance。

#### 6. 完成节点维护后, 禁用维护模式:

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

#### 7. 确保节点不再处于维护模式:

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

## 更换节点

使用 kubectl 命令和 Astra Data Store 预览版替换集群中的故障节点。

### 步骤

#### 1. 列出所有节点:

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

响应:

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

#### 2. 描述集群:

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

响应:



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

### 3. 验证故障节点上的 Node HA 是否标记为 false：

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

响应：

```
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. 通过将 'AddsNode Count' 的值减至 3，修改 astradscluster CR 以删除故障节点：

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

响应：

```
apiVersion: astrads.netapp.io/v1alpha1
kind: AstraDSCluster
metadata:
  name: cluster-multinodes-21209
```

```

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

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

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

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

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

```

```

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

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

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

```

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

##### 5. 验证是否已从集群中删除此节点:

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

响应:

NAME	STATUS	ROLES	AGE	VERSION
LABELS				
sti-astramaster-237	Ready	control-plane,master	24h	v1.20.0
sti-rx2540-532d	Ready	<none>	24h	v1.20.0
sti-rx2540-533d	Ready	<none>	24h	

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

响应:

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

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

响应:

NAME	STATUS	ROLES	AGE	VERSION
LABELS				
sti-astramaster-237	Ready	control-plane,master	24h	v1.20.0
beta.kubernetes.io/arch=amd64,				
sti-rx2540-532d	Ready	<none>	24h	v1.20.0
astrads.netapp.io/node-removal				

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

响应:

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

6. 通过修改集群 CR 将节点添加到集群以进行更换。节点数将递增至 4。验证是否已选取新节点进行添加。

```
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
```

响应:

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

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

响应:

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

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

响应:

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

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

响应:

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

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

响应:

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

## 更换驱动器

当集群中的驱动器发生故障时，必须尽快更换驱动器以确保数据完整性。驱动器发生故障时，您将在集群 CR 节点状态，集群运行状况信息和指标端点中看到故障驱动器信息。

在 **nodeStatus.drivestatuses** 中显示故障驱动器的集群示例

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

响应:

```

...
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
  ...

```

故障驱动器 CR 会在集群中自动创建，其名称与故障驱动器的 UUID 相对应。

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

响应：

```

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

```

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

响应:

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

步骤

1. 使用 `kubectl astrad show-replacements` 命令列出可能的替代驱动器, 该命令可筛选符合更换限制 (未在集群中使用, 未挂载, 无分区以及等于或大于故障驱动器) 的驱动器。

要在不筛选可能的替代驱动器的情况下列出所有驱动器, 请在 `show-replacements` 命令中添加 `-all`。

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

响应:

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

2. 使用 `replace` 命令将驱动器替换为已传递的序列号。如果 `-wait` 时间已过, 则命令将完成替换或失败。

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



如果使用不适当的 `-replaceWith` 序列号执行 `kubectl astrad faileddrive replace`, 则会显示类似以下内容的错误:



```
~% 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. 要重新运行驱动器更换，请使用 `-force` 和上一个命令：

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

有关详细信息 ...

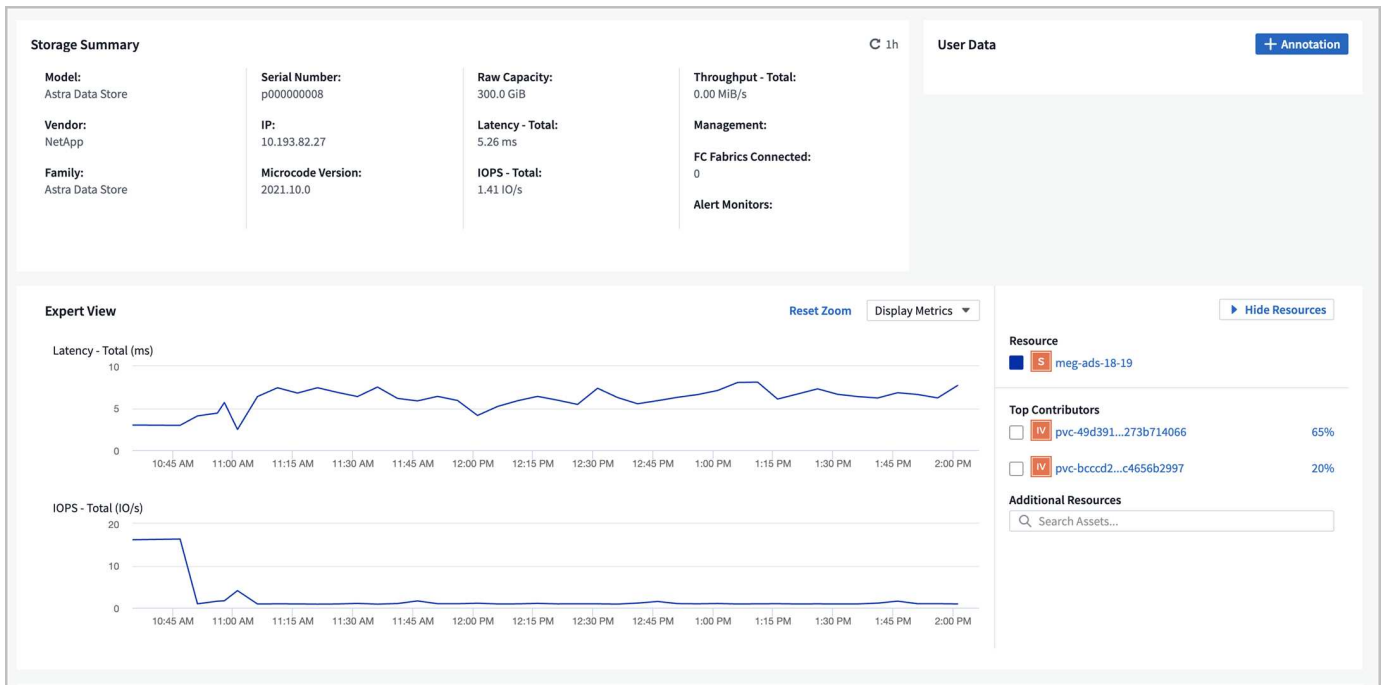
- ["使用 kubectl 命令管理 Astra Data Store 预览资产"](#)

## 使用 Cloud Insights 监控指标

您可以使用 Cloud Insights 监控 Astra 数据存储预览指标。

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

以下是 Cloud Insights 中显示的一些示例 Astra 数据存储预览指标：



您还可以使用显示在 Astra Data Store 预览版中生成的指标列表 [\[Open Metrics API help\]](#)。

## 完成 Cloud Insights 连接前提条件任务

在将 Astra 数据存储与 Cloud Insights 连接之前，您需要：

- "安装 Astra 数据存储监控操作员" 这是 Astra Data Store 预览安装说明的一部分。
- "安装 kubectl-astrad 二进制文件" 这是 Astra Data Store 预览安装说明的一部分。
- "创建 Cloud Insights 帐户"。
- 确保以下命令可用： `jk` ， `curl` ， `grep` 和 `JQ`

收集以下信息：

- 对以下类别具有读 / 写权限的 \* Cloud Insights API 密钥 \*：采集单元，数据收集，数据载入和日志载入。此参数将用于读 / 写操作，设置采集单元和设置数据载入过程。
- \* Kubernetes API 服务器 IP 地址和端口 \*。用于监控 Astra Data Store 预览集群。
- \* Kubernetes API 令牌 \*。此选项用于调用 Kubernetes API。
- \* 永久性卷配置 \*。有关如何配置永久性卷的信息。有关详细信息，请参见下面的 "采集单元"。

## 下载并运行安装脚本

Cloud Insights 提供了 Bash 脚本，用于通过监控操作员启用 Astra 数据存储预览监控。安装脚本将安装一个采集单元以及 Astra 数据存储收集器，一个 Telegraf 代理和一个 Fluent Bit Agent。

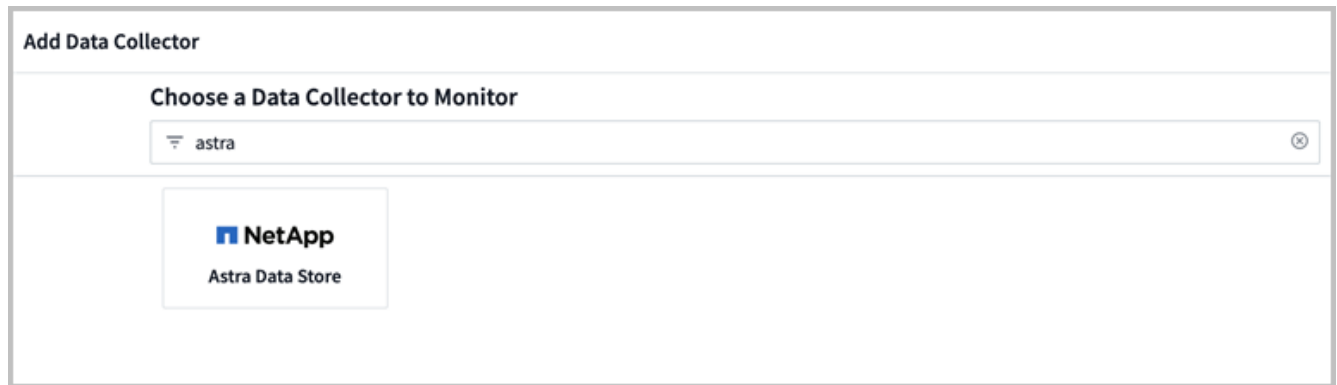
下载 Cloud Insights 租户域名和选定 API 访问密钥后，此域名和 API 访问密钥将嵌入安装程序脚本中。

然后，指标将按如下所示发送：

- Telegraf 将向 Cloud Insights 数据湖发送指标。
- Fluent Bit 会将日志发送到日志载入服务。


#### 步骤

1. 如果您还没有 Cloud Insights 帐户，请创建一个。
2. 登录到 Cloud Insights 。
3. 从 Cloud Insights 菜单中，单击 \* 管理 \* > \* 数据收集器 \* 。
4. 单击 \* + Data Collector\* 以添加新收集器。




5. 单击 \* Astra Data Store\* 图块。
6. 选择正确的 API 访问令牌或创建新的 API 访问令牌。
7. 按照说明下载安装程序脚本，更新权限并运行此脚本。


此脚本包含您的 Cloud Insights 租户 URL 和选定 API 访问令牌。



Select a Data Collector



Configure Collector




**NetApp**  
Astra Data Store

## Configure Collector

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

**What Operating System or Platform Are You Using?**

 Kubernetes

**Select existing API Access Token or create a new one**

default\_ads\_api\_key1 (...d0gHof)

[+ API Access Token](#)

[Production Best Practices ?](#)

**Configure Astra Data Store** [Need Help?](#)

- 1

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

Copy Installer Script

☐ Reveal Installer Script

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

CI_DOMAIN_NAME="f49uaky.gstabler-ads.cloudinsights-dev.netapp.com"
CI_API_KEY="eyJraWQ1OiI5OTk5IiwidHlwIjoIc3R5bWVudF9hZHNfYXBPX2t1eTEgKG9uIGJlaGFsZiBvZiBhZG1pbik"

```
- 3

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

Copy Permissions Command

☐ Reveal Permissions Command

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

```
- 5

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

Copy Install Command

☐ Reveal Install Command

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

```
- 7

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

Complete Setup

8. 完成脚本后，单击 \* 完成设置 \*。

安装脚本完成后，Astra Data Store 收集器将显示在 Datasources 列表中。



如果脚本因错误而退出，您可以在解决错误后重新运行它。如果您的环境不使用默认设置，则该脚本支持其他参数，例如监控操作员命名空间和 Kubernetes API 服务器端口。使用 `-h` 选项查看使用情况和帮助文本。

运行安装脚本会生成如下输出：

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

## 采集单元存储

采集单元需要三个永久性卷来存储安装文件，配置数据和日志。监控操作员使用默认存储类创建永久性卷请求。在运行安装程序脚本时，您可以使用 `-s` 选项指定其他存储类名称。

如果您的 Kubernetes 集群没有存储配置程序（例如 NetApp Trident），您可以在运行安装程序脚本时使用 `-r` 选项提供本地文件系统路径。设置 `-r` 选项后，安装程序脚本会在提供的目录中创建三个永久性卷。此目录至少需要 150 GB 的可用空间。

## 代理 CR 示例

以下是运行安装程序脚本后 `monitoring-NetApp` 代理 CR 的外观示例。

```
spec:
  au:
    isEnabled: true
    storageClassName: auto-sc
  cluster-name: meg-ads-21-22-29-30
  docker-repo: docker.repo.eng.netapp.com/global/astra
  fluent-bit:
  - name: ads-tail
    outputs:
    - sink: ADS_STDOUT
  substitutions:
  - key: TAG
    value: firetapems
  - key: LOG_FILE
```

```

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

```

## 安装程序脚本帮助

安装程序脚本的完整帮助文本如下所示：

```
./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.
```

## 编辑 Cloud Insights 连接

您可以稍后编辑 Kubernetes API 密钥或 Cloud Insights API 密钥：

- 如果要更新 Kubernetes API 密钥，应从 Cloud Insights UI 编辑 Astra 数据存储收集器。
- 如果要更新用于遥测和日志的 Cloud Insights API 密钥，应使用 kubectl 命令编辑监控操作员 CR。

### 更新 Kubernetes API 令牌

1. 登录到 Cloud Insights。
2. 选择 \* 管理 \* > \* 数据收集器 \* 以访问数据收集器页面。
3. 找到 Astra Data Store 集群的条目。
4. 单击页面右侧的菜单，然后选择 \* 编辑 \*。

### 更新 Cloud Insights API 访问令牌

1. 登录到 Cloud Insights。
2. 选择 \* 管理 \* > \* API 访问 \* 并单击 \* + API 访问令牌 \*，创建新的 Cloud Insights API 访问令牌。
3. 编辑代理 CR：

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

4. 找到 output-sink 部分，然后找到名为 ` "Ci" ` 的条目。
5. 对于标签 api-key ，请将当前值替换为新的 API 密钥。

此部分如下所示：

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

6. 保存并退出编辑器窗口。

监控操作员将更新 Telegraf 和 Fluent Bit 以使用新的 API 密钥。

## 断开与 Cloud Insights 的连接

要断开与 Cloud Insights 的连接，您需要先从 Cloud Insights UI 中删除 Astra 数据存储收集器。完成后，您可以从监控操作员中删除采集单元，Telegraf 和 Fluent 位配置。

### 删除 Astra Data Store 预览收集器

1. 登录到 Cloud Insights 。
2. 选择 \* 管理 \* > \* 数据收集器 \* 以访问数据收集器页面。
3. 找到 Astra Data Store 集群的条目。
4. 选择屏幕右侧的串行菜单，然后选择 \* 删除 \* 。
5. 单击确认页面上的 \* 删除 \* 。

### 删除采集单元，Telegraf 和 Fluent 位

1. 编辑代理 CR：

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

2. 找到 au 部分并设置 isenabled : false
3. 找到 flual-bit 部分，然后删除名为 ` ads-tail ci ` 的插件。如果没有其他插件，您可以删除 flual-bit 部分。
4. 找到 telaff 部分，然后删除名为 ` ads-open-metric ` 的插件。如果没有其他插件，您可以删除 电报 部分。
5. 找到 output-sink 部分，然后删除名为 ` "Ci" ` 的接收器。



## 6. 保存并退出编辑器窗口。

监控操作员将更新 Telegraf 和 Fluent 位配置并删除采集单元 POD。

## 7. 如果您使用本地目录作为采集单元 PV，而不是存储配置程序，请删除这些 PV：

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

然后，删除运行 AU 的节点上的实际目录。

## 8. 删除采集单元 POD 后，您可以从 Cloud Insights 中删除采集单元。

- 在 Cloud Insights 菜单中，选择 \* 管理 \* > \* 数据收集器 \*。
- 单击 \* 采集单元 \* 选项卡。
- 单击采集单元 POD 旁边的菜单。
- 单击 \* 删除 \*。

监控操作员将更新 Telegraf 和 Fluent 位配置并删除采集单元。

## 打开指标 API 帮助

下面列出了可用于从 Astra Data Store 预览版收集指标的 API。

- "help" 行说明了指标。
- "type" 行表示指标是量表还是计数器。

```
# HELP astrads_cluster_capacity_logical_percent Percentage cluster logical
capacity that is used (0-100)
# TYPE astrads_cluster_capacity_logical_percent gauge
# HELP astrads_cluster_capacity_max_logical Max Logical capacity of the
cluster in bytes
# TYPE astrads_cluster_capacity_max_logical gauge
# HELP astrads_cluster_capacity_max_physical The sum of the space in the
cluster in bytes for storing data after provisioning efficiencies, data
reduction algorithms and replication schemes are applied
# TYPE astrads_cluster_capacity_max_physical gauge
# HELP astrads_cluster_capacity_ops The IO operations capacity of the
cluster
# TYPE astrads_cluster_capacity_ops gauge
# HELP astrads_cluster_capacity_physical_percent The percentage of cluster
physical capacity that is used (0-100)
# TYPE astrads_cluster_capacity_physical_percent gauge
# HELP astrads_cluster_capacity_used_logical The sum of the bytes of data
in all volumes in the cluster before provisioning efficiencies, data
reduction algorithms and replication schemes are applied
```

```

# TYPE astrads_cluster_capacity_used_logical gauge
# HELP astrads_cluster_capacity_used_physical Used Physical capacity of a
cluster in bytes
# TYPE astrads_cluster_capacity_used_physical gauge
# HELP astrads_cluster_other_latency The sum of the accumulated latency in
seconds for other IO operations of all the volumes in a cluster. Divide by
astrads_cluster_other_ops to get the average latency per other operation
# TYPE astrads_cluster_other_latency counter
# HELP astrads_cluster_other_ops The sum of the other IO operations of all
the volumes in a cluster
# TYPE astrads_cluster_other_ops counter
# HELP astrads_cluster_read_latency The sum of the accumulated latency in
seconds of read IO operations of all the volumes in a cluster. Divide by
astrads_cluster_read_ops to get the average latency per read operation
# TYPE astrads_cluster_read_latency counter
# HELP astrads_cluster_read_ops The sum of the read IO operations of all
the volumes in a cluster
# TYPE astrads_cluster_read_ops counter
# HELP astrads_cluster_read_throughput The sum of the read throughput of
all the volumes in a cluster in bytes
# TYPE astrads_cluster_read_throughput counter
# HELP astrads_cluster_storage_efficiency Efficacy of data reduction
technologies. (logical used / physical used)
# TYPE astrads_cluster_storage_efficiency gauge
# HELP astrads_cluster_total_latency The sum of the accumulated latency in
seconds of all IO operations of all the volumes in a cluster. Divide by
astrads_cluster_total_ops to get average latency per operation
# TYPE astrads_cluster_total_latency counter
# HELP astrads_cluster_total_ops The sum of the IO operations of all the
volumes in a cluster
# TYPE astrads_cluster_total_ops counter
# HELP astrads_cluster_total_throughput The sum of the read and write
throughput of all the volumes in a cluster in bytes
# TYPE astrads_cluster_total_throughput counter
# HELP astrads_cluster_utilization_factor The ratio of the current cluster
IO operations based on recent IO sizes to the cluster iops capacity. (0.0
- 1.0)
# TYPE astrads_cluster_utilization_factor gauge
# HELP astrads_cluster_volume_used The sum of used capacity of all the
volumes in a cluster in bytes
# TYPE astrads_cluster_volume_used gauge
# HELP astrads_cluster_write_latency The sum of the accumulated latency in
seconds of write IO operations of all the volumes in a cluster. Divide by
astrads_cluster_write_ops to get the average latency per write operation
# TYPE astrads_cluster_write_latency counter
# HELP astrads_cluster_write_ops The sum of the write IO operations of all

```

```

the volumes in a cluster
# TYPE astrads_cluster_write_ops counter
# HELP astrads_cluster_write_throughput The sum of the write throughput of
all the volumes in a cluster in bytes
# TYPE astrads_cluster_write_throughput counter
# HELP astrads_disk_base_seconds Base for busy, pending and queued.
Seconds since collection began
# TYPE astrads_disk_base_seconds counter
# HELP astrads_disk_busy Seconds the disk was busy. 100 *
(astrads_disk_busy / astrads_disk_base_seconds) = percent busy (0-100)
# TYPE astrads_disk_busy counter
# HELP astrads_disk_capacity Raw Capacity of a disk in bytes
# TYPE astrads_disk_capacity gauge
# HELP astrads_disk_io_pending Summation of the count of pending io
operations for a disk times time. Divide by astrads_disk_base_seconds to
get the average pending operation count
# TYPE astrads_disk_io_pending counter
# HELP astrads_disk_io_queued Summation of the count of queued io
operations for a disk times time. Divide by astrads_disk_base_seconds to
get the average queued operations count
# TYPE astrads_disk_io_queued counter
# HELP astrads_disk_read_latency Total accumulated latency in seconds for
disk reads. Divide by astrads_disk_read_ops to get the average latency per
read operation
# TYPE astrads_disk_read_latency counter
# HELP astrads_disk_read_ops Total number of read operations for a disk
# TYPE astrads_disk_read_ops counter
# HELP astrads_disk_read_throughput Total bytes read from a disk
# TYPE astrads_disk_read_throughput counter
# HELP astrads_disk_write_latency Total accumulated latency in seconds for
disk writes. Divide by astrads_disk_write_ops to get the average latency
per write operation
# TYPE astrads_disk_write_latency counter
# HELP astrads_disk_write_ops Total number of write operations for a disk
# TYPE astrads_disk_write_ops counter
# HELP astrads_disk_write_throughput Total bytes written to a disk
# TYPE astrads_disk_write_throughput counter
# HELP astrads_value_scrape_duration Duration to scrape values
# TYPE astrads_value_scrape_duration gauge
# HELP astrads_volume_capacity_available The minimum of the available
capacity of a volume and the available capacity of the cluster in bytes
# TYPE astrads_volume_capacity_available gauge
# HELP astrads_volume_capacity_available_logical Logical available
capacity of a volume in bytes
# TYPE astrads_volume_capacity_available_logical gauge
# HELP astrads_volume_capacity_percent Percentage of volume capacity

```

```

available (0-100). (capacity available / provisioned) * 100
# TYPE astrads_volume_capacity_percent gauge
# HELP astrads_volume_capacity_provisioned Provisioned capacity of a
volume in bytes after setting aside the snapshot reserve. (size - snapshot
reserve = provisioned)
# TYPE astrads_volume_capacity_provisioned gauge
# HELP astrads_volume_capacity_size Total capacity of a volume in bytes
# TYPE astrads_volume_capacity_size gauge
# HELP astrads_volume_capacity_snapshot_reserve_percent Snapshot reserve
percentage of a volume (0-100)
# TYPE astrads_volume_capacity_snapshot_reserve_percent gauge
# HELP astrads_volume_capacity_snapshot_used The amount of volume snapshot
data that is not in the active file system in bytes
# TYPE astrads_volume_capacity_snapshot_used gauge
# HELP astrads_volume_capacity_used Used capacity of a volume in bytes.
This is bytes in the active filesystem unless snapshots are consuming more
than the snapshot reserve. (bytes in the active file system + MAX(0,
snapshot_used-(snapshot_reserve_percent/100*size))
# TYPE astrads_volume_capacity_used gauge
# HELP astrads_volume_other_latency Total accumulated latency in seconds
for operations on a volume that are neither read or write. Divide by
astrads_volume_other_ops to get the average latency per other operation
# TYPE astrads_volume_other_latency counter
# HELP astrads_volume_other_ops Total number of operations for a volume
that are neither read or write
# TYPE astrads_volume_other_ops counter
# HELP astrads_volume_read_latency Total accumulated read latency in
seconds for a volume. Divide by astrads_volume_read_ops to get the average
latency per read operation
# TYPE astrads_volume_read_latency counter
# HELP astrads_volume_read_ops Total number of read operations for a
volume
# TYPE astrads_volume_read_ops counter
# HELP astrads_volume_read_throughput Total read throughput for a volume
in bytes
# TYPE astrads_volume_read_throughput counter
# HELP astrads_volume_total_latency Total accumulated latency in seconds
for all operations on a volume. Divide by astrads_volume_total_ops to get
the average latency per operation
# TYPE astrads_volume_total_latency counter
# HELP astrads_volume_total_ops Total number of operations for a volume
# TYPE astrads_volume_total_ops counter
# HELP astrads_volume_total_throughput Total throughput for a volume in
bytes
# TYPE astrads_volume_total_throughput counter
# HELP astrads_volume_write_latency Total accumulated write latency in

```

```
seconds for volume. Divide by astrads_volume_write_ops to get the average
latency per write operation
# TYPE astrads_volume_write_latency counter
# HELP astrads_volume_write_ops Total number of write operations for a
volume
# TYPE astrads_volume_write_ops counter
# HELP astrads_volume_write_throughput Total write throughput for a volume
in bytes
# TYPE astrads_volume_write_throughput counter
```

## 配置和监控事件日志

要监控事件管理系统（EMS）日志，您可以执行以下高级任务：

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

### 在 **Astra Data Store** 预览集群自定义资源（CR）中配置监控

如果尚未在 Astra Data Store 预览集群 CR 上配置监控选项，您可以使用 `astrad extensions` 进行设置。

输入 ...

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

其中：

- 已安装代理的命名空间：这是监控代理，它是监控操作员的 CR。
- `-r`` 是可选的。

### 设置 **Cloud Insights**

请参见 ["如何设置 NetApp Cloud Insights"](#) 用于 Astra Data Store 预览版。

### 将事件日志流式传输到 **Elastic**

要将 EMS 事件和其他 POD 日志流式传输到将日志流式传输到 Elastic 等第三方端点，请使用 `astrad` 扩展。

输入 ...

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



弹性主机名可以是 IP 地址。

## 将 Astra 控制中心与 Astra Data Store 预览结合使用

您可以使用 Astra 控制中心用户界面（UI）执行 Astra Data Store 预览任务。

### 为 Astra 数据存储预览设置 Astra 控制中心

要使用适用于 Astra Data Store 的 Astra 控制中心 UI 预览版，您需要完成以下任务：

- 添加集群。这是安装了 Astra Data Store 预览版的底层 Kubernetes 集群。请参见 ["将 Astra Data Store 预览集群导入到您的 Astra Control Center 安装中"](#)。
- 为该集群添加 Astra Data Store 预览存储后端。



如果您添加了存储后端，并且不存在具有 Astra Data Store 预览版的 Kubernetes 集群，则需要先添加一个集群。

### 您可以在 Astra 控制中心执行的操作

在为 Astra 数据存储预览设置 Astra 控制中心后，您可以使用 Astra 控制中心 UI 完成以下任务：

- ["使用 Astra 控制中心监控 Astra Data Store 预览资产的运行状况"](#)。
- ["管理 Astra Data Store 预览后端存储"](#)。
- ["查看节点，磁盘和永久性卷声明（PVC）及其状态"](#)。

### 有关详细信息 ...

- ["Astra 系列简介"](#)
- ["Astra 控制中心文档"](#)
- ["使用 Astra Control API"](#)

## 卸载 Astra Data Store 预览版

要卸载 Astra Data Store 预览版，您需要删除工作负载，绑定，卷，导出策略，集群，许可证，部署环境和 Astra Data Store 预览命名空间。

您需要什么？ **#8217** ；将需要什么

- root 管理权限

Astra Data Store 预览卸载过程将指导您完成以下高级步骤：

- [\[Remove existing workloads and bindings\]](#)
- [\[Delete the volumes and export policies\]](#)
- [\[Delete the Astra Data Store preview cluster\]](#)

- [\[Delete the license\]](#)
- [\[Delete the Astra Data Store preview installation\]](#)
- [\[Validate the removal of the astrads-system namespace\]](#)
- [\[Ensure containers are not running on worker nodes\]](#)
- [\[Delete OpenShift Container Platform resources\]](#)

## 删除现有工作负载和绑定

卸载 Astra Data Store 预览版之前，必须先删除以下内容

- 使用 Astra Data Store 预览版作为存储后端的所有应用程序工作负载
- 使用 Astra Data Store 预览作为后端的 Trident 绑定

这样可以确保 Kubernetes 环境保持干净，这在重新安装时非常重要。

## 卸载 Astra Data Store 预览集群和控制平面

按照以下步骤手动卸载 Astra Data Store 预览版。

### 删除卷和导出策略

在删除集群之前，您应删除 Astra Data Store 预览卷和导出策略。



如果不先删除卷和导出策略，则集群删除过程将暂停，直到删除 Astra Data Store 预览卷对象为止。在开始删除集群之前删除这些项会更高效。

### 步骤

#### 1. 删除卷：

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

#### 2. 删除导出策略：

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

## 删除 Astra Data Store 预览集群

删除集群时，只会删除 Astra Data Store 预览集群对象自定义资源（CR）以及集群范围的资源。



即使在删除集群后，操作符，nodeinfo Pod 和集群控制器（即 Kubernetes 范围的资源）仍会保持不变。

删除集群还会从节点卸载底层操作系统，从而停止 fireap 和 netwd 服务。

卸载程序大约需要一分钟才能完成。然后，开始删除 Astra Data Store 预览集群范围的资源。

1. 删除集群：

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

## 删除许可证

1. 通过 SSH 连接到集群中的每个工作节点，并验证 `fireap` 或 `netwd` 未在工作节点中运行。
2. 删除 Astra Data Store 预览许可证：

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

## 删除 Astra Data Store 预览安装

删除集群中的控制器，操作员，命名空间和支持 Pod。

1. 删除 Astra Data Store 预览安装对象：

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

2. 删除数据存储 DemonSets 和所有 Astra Data Store 预览控制器资源：

```
~% 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. 删除其余项目和操作符 YAML 文件：

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



验证是否删除了 **astrads-system** 命名空间

确保以下命令不返回任何结果：

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

确保工作节点上未运行容器

验证 **fireap** 或 **netwd** 等容器是否未在工作节点上运行。在每个节点上运行以下命令。

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

## 删除 OpenShift 容器平台资源

如果您在 Red Hat OpenShift 容器平台（OCP）上安装了 Astra Data Store preview，则可以卸载 OCP 安全上下文约束（SCC）和绑定资源。

OpenShift 使用安全上下文约束（SCC）来控制 Pod 可以执行的操作。

完成标准卸载过程后，请完成以下步骤。

### 1. 删除 SCC 资源：

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

### 2. 删除 rolebindings 资源：

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



忽略这些步骤中的 "未找到资源错误"。

### 3. 从所有 Kubernetes 节点中删除 `/var/lib/kubelet/config.yaml`。

## 手动删除示例

下面显示了执行手动卸载脚本的示例。

```
$ 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
```

## 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.