



# **Astra Data Store文件**

## **Astra Data Store**

NetApp  
June 08, 2022

# 目錄

Astra Data Store文件	1
版本資訊	2
本版Astra Data Store的新功能	2
已知問題	3
已知限制	3
概念	5
深入瞭解Astra Data Store	5
Astra Data Store部署模式	6
叢集擴充	7
Astra Data Store的儲存效率	7
Astra資料儲存區的安全性	8
開始使用	10
Astra資料儲存區需求	10
Astra Data Store快速入門	14
安裝Astra Data Store	15
設定Astra Data Store元件	33
Astra Data Store Early Access Program (AP) 版本限制	44
Astra Data Store的常見問題集	45
使用Astra資料儲存區	48
使用kveccli命令管理Astra Data Store資源	48
部署測試應用程式	54
管理Astra Data Store叢集	58
監控Astra資料儲存區	68
安全的Astra資料儲存區	82
更新Astra Data Store授權	87
升級Astra Data Store	88
使用自動指令碼解除安裝Astra Data Store	90
使用Astra Data Store搭配VMware	105
瞭解Astra Data Store與VMware的相關資訊	105
Astra Data Store搭配VMware需求	105
使用VMware設定Astra Data Store	106
監控VMware安裝的元件	113
管理VMware安裝的Astra Data Store元件	115
從VMware整合式環境中解除安裝Astra Data Store	118
知識與支援	119
疑難排解	119
取得協助	119
自動支援監控	119
舊版Astra Data Store	124

法律聲明 ..... 125

    版權 ..... 125

    商標 ..... 125

    專利 ..... 125

隱私權政策 ..... 125

開放原始碼 ..... 125

# Astra Data Store文件

# 版本資訊

我們很高興宣布推出2022.05年Astra Data Store早期存取方案（EAP）版本。

- ["本版Astra Data Store的新功能"](#)
- ["已知問題"](#)
- ["已知限制"](#)

歡迎前往Twitter @NetAppDoc追蹤我們的動態。請透過成為來傳送有關文件的意見反應 ["GitHub貢獻者"](#) 或傳送電子郵件至[doccomments@netapp.com](mailto:doccomments@netapp.com)。

## 本版Astra Data Store的新功能

我們很高興宣布推出2022.05年Astra Data Store早期存取方案（EAP）版本。

### 2022年5月31日（2022.05年市場評估版本）

Astra Data Store 2022.05版本包含下列功能：

- VMware環境整合：
  - NFS vVol資料存放區工作流程和儲存原則型管理儲存資源配置。
  - 適用於VMware vSphere的Astra外掛程式、可從vCenter進行原生儲存管理與監控。
- 增強的叢集管理功能、包括 ["Astra Control Center的GUI型叢集部署"](#)。
- 支援 ["更大規模的叢集擴充"](#)（節點數量較多、CPU和容量增加）。
- 安全性增強功能（金鑰與憑證、以及支援外部金鑰管理（KMIP））。

### 2022年2月8日（2021.12.1）

Astra Data Store預覽（21：12）的修補程式版本（2021：12）。

- 此版本的Astra Data Store預覽支援採用Calico CNI的VXLAN組態。
- Astra Data Store預覽現在支援啟用Calico WireGuard的網路組態。
- 更新的《適用網路介面》一節包含改良的說明性註解。
- 全新的Astra Data Store解除安裝指令碼、現在已成為KECBECKLET命令解除安裝程序的更簡單替代方法。

### 2021年12月21日（21日12）

Astra Data Store預覽的初始版本。

- ["它是什麼"](#)
- ["部署模式與元件"](#)
- ["開始使用所需的一切"](#)
- ["安裝"](#) 和 ["設定"](#)

- ["管理" 和 "監控" 效能](#)
- ["利用功能進行監控Cloud Insights"](#)
- ["取得協助" 和 "使用自動支援監控"](#)

如需詳細資訊、請參閱

- ["已知問題"](#)
- ["已知限制"](#)
- ["NetApp知識庫文章"](#)

## 已知問題

已知問題可識別可能導致您無法成功使用本產品版本的問題。

### Astra Data Store 2202.05早期存取方案（EAP）版本

此版本包含這些已知問題。請仔細檢閱這些已知問題。

#### vSphere中未顯示的儲存後端工作

提交使用現有儲存後端或移除儲存後端的動作時、該工作不會出現在vSphere用戶端的「近期工作」面板中。

#### 建立VM時會建立額外的資料夾

在vSphere中建立VM時、有時會建立額外的資料夾、而非在資料存放區下的每個VM建立單一資料夾。

#### 適用於VMware vSphere的Astra外掛程式中的篩選檢視會間歇性地不顯示資料

虛擬機器的表格欄篩選器偶爾會省略篩選過的資料。篩選後的檢視會改為顯示載入圖示。若要解決此問題、您可以前往其他檢視並重設篩選條件、以再次顯示資料。

#### Kubernetes工作節點的IP位址限制

Astra Data Store叢集建立在下列案例中失敗：

1. Kubernetes工作節點的IP位址裝載在未指派給Astra Data Store的介面上。
2. IP位址與指派給Astra Data Store叢集的叢集/資料IP位址位於相同的子網路中。

如需詳細資訊、請參閱

- ["已知限制"](#)

## 已知限制

已知限制指出產品早期存取方案（AP）版本不支援的平台、裝置或功能、或是無法與產品正確互通的功能。

## Astra Data Store 2022.05早期存取方案（AP）版本

此版本包含這些已知限制。請仔細檢閱這些已知限制。

### VMware整合限制

不支援下列功能：

- VM工作流程：
  - 複製VM、包括從範本複製
  - 在移除VM磁碟之後附加該磁碟
- VVOL複寫
- 刪除VVol資料存放區
- 一流磁碟（FCD）
- VMware VMotion
- 連結模式vCenter伺服器
- 法規遵循警示與通知
- 多vCenter與多VASA Provider組態
  - 支援每個vCenter組態的單一VASA Provider
- 儲存原則型管理僅適用於VM層級、在建立VM後無法修改。

無法同時調整多個**VM**磁碟的大小

在此版本中、您一次只能調整一個磁碟的大小。如果您嘗試一次調整多個磁碟的大小、則調整大小作業僅適用於第一個磁碟。

如需詳細資訊、請參閱

- ["已知問題"](#)

# 概念

## 深入瞭解Astra Data Store

Astra Data Store是Kubernetes原生的共享檔案軟體定義儲存設備（SDS）解決方案、適用於內部部署資料中心、可協助客戶管理其雲端原生應用程式。Astra Data Store為Container和VM工作負載提供原生共享檔案服務、並提供NetApp企業資料管理功能。

有了Astra Data Store、您可以執行下列作業：

- 支援**Kubernetes**容器化工作負載：搭配您常用的企業資料管理服務與工具。
- \*使用Kubernetes的「應用程式即服務」平台進行DevOps\*：建立彈性、軟體定義的自助式平台、提供自動化、可重複執行的服務、免除開發人員的複雜性

Astra Data Store是Astra產品系列的一部分。深入瞭解 ["Astra系列"](#)。

### Astra Data Store功能

Astra Data Store提供端點對端點Kubernetes原生儲存設備與資料管理功能、適用於雲端原生應用程式。這些功能包括：

- \* Kubernetes原生共享檔案服務\*：提供Kubernetes原生的共享檔案服務、使用標準NFS用戶端做為容器和VM的統一化資料儲存區
- 雲端擴充：在相同資源池上提供Kubernetes原生的多重平行檔案系統、以達到類似雲端的擴充與使用率、免除管理儲存設備與叢集的需求。
- \* API第一方法\*：以程式碼形式提供基礎架構、以供自動化叢集與工作負載管理之用。
- 企業級資料管理：提供自動化的應用程式感知資料保護與災難恢復：
  - \* NetApp技術\*：運用NetApp資料管理技術進行快照、備份、複寫及複製、讓使用者能夠在Kubernetes中建置及部署企業應用程式。
  - 恢復能力：為Kubernetes原生工作負載使用複寫與銷毀編碼技術、以提升恢復能力。
  - 資料效率：透過即時重複資料刪除與壓縮功能、在擴充時控制成本。
- 適合您現有的環境：支援以微服務為基礎的傳統工作負載、提供Kubernetes主要發佈產品、提供檔案儲存功能、並可在您選擇的硬體上執行。
- 整合**NetApp Cloud Insights Setc**：提供觀察、分析及監控功能、以持續最佳化。

### Astra Data Store入門

首先、["深入瞭解Astra Data Store需求"](#)。

然後、["開始使用"](#)。

### 以取得更多資訊

- ["Astra系列簡介"](#)
- ["Astra Control Service文件"](#)



- ["Astra Control Center文件"](#)
- ["Astra Trident文件"](#)
- ["使用Astra Control API"](#)
- ["本文檔 Cloud Insights"](#)
- ["本文檔 ONTAP"](#)

## Astra Data Store部署模式

Astra Data Store使用與Kubernetes部署及協調的應用程式、直接在主機上管理儲存磁碟機。

您可以使用下列其中一個選項、在裸機或虛擬伺服器上安裝Astra Data Store：

- 部署在獨立的Kubernetes專屬叢集上、為Kubernetes應用程式提供持續磁碟區服務、並在獨立的叢集（獨立叢集）中執行。
- 在Kubernetes叢集上部署、也會在同一個節點集區（融合式叢集）上裝載其他工作負載應用程式。
- 在Kubernetes叢集上部署、也會在不同的節點集區（分離式叢集）上裝載其他工作負載應用程式。

["深入瞭解Astra Data Store硬體需求"](#)。

Astra Data Store是Astra產品系列的一部分。如需Astra全系列產品的觀點、請參閱 ["Astra系列簡介"](#)。

## Astra Data Store生態系統

Astra資料儲存區可與下列項目搭配運作：

- **\* Astra Control Center\***：使用Astra Control Center軟體、在內部部署環境中、對Kubernetes叢集進行應用程式感知資料管理。輕鬆備份Kubernetes應用程式、將資料移轉至不同的叢集、並即時建立可運作的應用程式複本。

Astra Control Center支援Kubernetes叢集、搭配Astra Trident儲存後端ONTAP 搭配使用支援功能的支援、或搭配使用Astra Data Store儲存後端。

- **\* Astra Trident \***：Astra Trident \*是完全受支援的開放原始碼儲存資源配置程式、由NetApp維護、可讓您為Docker和Kubernetes所管理的容器化應用程式建立儲存磁碟區。

您可以使用Astra Trident在Astra資料儲存區上建立磁碟區。

- **\* 《NetApp雲端基礎架構監控工具》《支援》**、可讓您監控由Astra Control管理的Kubernetes叢集的效能與使用率。Cloud Insights Cloud Insights可將儲存使用量與工作負載建立關聯。Cloud Insights

當您在Cloud Insights Astra Control中啟用「支援」功能時、遙測資訊會顯示在Astra Control UI頁面中。顯示Astra Data Store中所管理資源的相關資訊。Cloud Insights

## Astra Data Store介面

您可以使用不同的介面來完成工作：

- **網路使用者介面（UI）**：Astra Control Service和Astra Control Center使用相同的網路型UI、您可以在其中

管理、移轉及保護應用程式。此UI也會顯示Astra Data Store Volume的相關資訊。

- **\* API\***：Astra Control Service和Astra Control Center使用相同的Astra Control API。使用API、您可以執行與使用UI相同的工作。您也可以使用Astra Control API擷取Astra Data Store的相關資訊。
- **\* kubectl命令\***：若要使用Astra Data Store、您可以直接使用kubectl命令。
- **\* Kubernetes擴充\***：此外、您也可以使用Astra Data Store Kubernetes API擴充。

自訂資源定義（CRD）是Kubernetes REST API的延伸、是在部署Astra Data Store營運者時所建立的API。外部實體透過呼叫Kubernetes API伺服器來與客戶需求日互動。Astra Data Store會監看特定客戶需求日的更新、然後呼叫內部REST API。

## 以取得更多資訊

- ["Astra系列簡介"](#)
- ["Astra Control Service文件"](#)
- ["Astra Control Center文件"](#)
- ["Astra Trident文件"](#)
- ["使用Astra Control API"](#)
- ["本文檔 Cloud Insights"](#)
- ["本文檔 ONTAP"](#)

## 叢集擴充

Astra Data Store支援叢集中不同類型和功能的節點。如果您要擴充叢集、Astra Data Store可支援新增具有任何效能功能的節點、只要這些節點不低於叢集中效能最低的節點即可。新節點的儲存容量必須與現有節點相同。所有節點、包括擴充期間的新節點、都必須至少符合中的最低需求 ["Astra資料儲存區需求"](#)。

## 以取得更多資訊

- ["Astra系列簡介"](#)
- ["Astra Control Service文件"](#)
- ["Astra Control Center文件"](#)
- ["Astra Trident文件"](#)
- ["使用Astra API"](#)
- ["本文檔 Cloud Insights"](#)
- ["本文檔 ONTAP"](#)

## Astra Data Store的儲存效率

Astra Data Store採用以NetApp ONTAP 功能和SolidFire 技術為基礎的儲存效率技術、包括：

- **精簡配置**：精簡配置的磁碟區是指未預先保留儲存設備的磁碟區。而是根據需要動態分配儲存設備。刪除磁碟區或LUN中的資料時、可用空間會釋出回儲存系統。

- 零區塊偵測與消除：Astra Data Store儲存系統具備精簡配置功能、可偵測磁碟區中已清零的區域、以便回收該空間並在其他地方使用。
- 壓縮：壓縮可將資料區塊合併到壓縮群組中、以減少磁碟區所需的實體儲存容量、每個區塊都儲存為單一區塊。壓縮資料的讀取速度比傳統的壓縮方法快、因為Astra Data Store只解壓縮包含所要求資料的壓縮群組、而非整個檔案。
- 重複資料刪除：重複資料刪除技術AFF 會捨棄重複區塊、並以單一共用區塊的參考資料來取代、藉此減少磁碟區（或是整個集合體中的所有磁碟區）所需的儲存容量。讀取已刪除重複資料通常不會收取效能費用。除了在過載節點上、寫入作業所產生的費用可忽略不計。

所有這些功能預設都會啟用。

請參閱 ["儲存效率詳細資料"](#)。

## 以取得更多資訊

- ["Astra系列簡介"](#)
- ["Astra Control Service文件"](#)
- ["Astra Control Center文件"](#)
- ["Astra Trident文件"](#)
- ["使用Astra Control API"](#)
- ["本文檔 ONTAP"](#)

## Astra資料儲存區的安全性

Astra Data Store使用多種方法來保護用戶端和系統管理員對儲存設備的存取、保護通訊和資料、以及防範病毒。

Astra Data Store採用下列安全方法：

- 使用相互傳輸層安全性（MTLS）進行通訊加密
- 角色型存取控制、可控制功能的存取
- 部署安全性
- 憑證管理
- 閒置的軟體加密、包括內部和外部金鑰管理

## 以取得更多資訊

- ["Astra系列簡介"](#)
- ["Astra Control Service文件"](#)
- ["Astra Control Center文件"](#)
- ["Astra Trident文件"](#)
- ["使用Astra Control API"](#)
- ["本文檔 Cloud Insights"](#)

- "本文檔 ONTAP"

# 開始使用

## Astra資料儲存區需求

開始驗證您的環境是否符合Astra Data Store要求。

Astra Data Store支援裸機和VM型部署。Astra Data Store叢集可在具有四個以上工作者節點的Kubernetes叢集上執行。Astra Data Store軟體可與在相同Kubernetes叢集上執行的其他應用程式共存。

- [\[Kubernetes worker node resources\]](#)
- [\[Hardware and software\]](#)
- [\[Networking\]](#)
- [Astra Trident](#)
- [\[Container Network Interfaces\]](#)
- [\[Licensing\]](#)



如果您打算從Astra Control Center管理Astra資料儲存區叢集、請確定您的Astra資料儲存區叢集符合 "[將由Astra Control Center管理的叢集需求](#)" 除了此處概述的要求之外、

## Kubernetes工作節點資源

以下是在Kubernetes叢集中的每個工作節點上、指派給Astra Data Store軟體所需的資源：

資源	最低	最大值
Astra Data Store叢集中的節點數	4.	16
資料磁碟機數量	<ul style="list-style-type: none"><li>• 3（存在獨立的快取裝置）</li><li>• 4（如果沒有快取裝置）</li></ul>	14
資料磁碟機大小	100GiB	4TiB
選用快取裝置的數量	1（8GiB尺寸）	不適用

Astra Data Store為每個Kubernetes工作節點支援下列vCPU和RAM組合：

- 9個vCPU、RAM為38GiB
- 23個vCPU、搭配94 GiB的RAM



若要獲得最佳寫入效能、您應該設定專屬的高耐用度、低延遲、低容量快取裝置。

每個工作節點都有下列額外需求：

- 用於儲存Astra Data Store記錄檔的主機磁碟（開機）可用空間為100GiB或更大。
- 至少一個10GbE或更快的網路介面、適用於叢集、資料和管理流量。此外、也可以使用額外的1GbE或更快介面來分隔管理流量。

## 硬體與軟體

Astra Data Store軟體已在下列硬體平台、軟體和儲存組態上通過驗證。請造訪 ["NetApp社群支援"](#) 如果Kubernetes叢集組態不同、

### 硬體平台

- Dell R640
- Dell R740
- HPE 360
- HPE DL380
- Lenovo SR630
- Lenovo SR650



VM型部署也可以使用列示為相容的伺服器 ["VMware相容性指南"](#)。

### 儲存設備

Astra Data Store已通過SATA、SAS及NVMe TLC SSD的驗證。

對於VM型部署、您可以使用呈現為虛擬磁碟或直通磁碟機的磁碟機。



- 如果您的主機在硬體RAID控制器後方使用SSD、請將硬體RAID控制器設定為使用「PassthThrough」模式。
- 每個磁碟機都應該有唯一的序號。在虛擬機器建立虛擬機器期間、將屬性「disk.enableuid=true」新增至虛擬機器的進階設定。

### 軟體

- Hypervisor：Astra Data Store已通過VMware VM部署與vSphere 7.0的驗證。Astra Data Store不支援KVM型部署。
- Astra Data Store已在下列主機作業系統上通過驗證：
  - Red Hat Enterprise Linux 8.4
  - Red Hat Enterprise Linux 8.2
  - Red Hat Enterprise Linux 7.9
  - Red Hat Enterprise Linux CoreOS (RMCOS)
  - CentOS 8.
  - Ubuntu 20.04
- Astra Data Store已通過下列Kubernetes發佈版本的驗證：
  - Red Hat OpenShift 4.6到4.9
  - Google Anthos 1.8到1.10
  - Kubernetes 1.20到1.23
  - Rancher RKE 1.3.3



Astra Data Store需要Astra Trident來進行儲存資源配置和協調。支援Astra Trident版本210.1至22.04。請參閱 "[Astra Trident安裝說明](#)"。

## 網路

Astra Data Store每個叢集需要一個IP位址、才能用於MVIP。它必須是未使用或未設定的IP位址、與MIP位於同一子網路中。Astra Data Store管理介面應與Kubernetes節點的管理介面相同。

此外、每個節點也可依照下表所述進行設定：



下表使用下列縮寫：MIP：管理IP位址CIP：叢集IP位址MVIP：管理虛擬IP位址

組態	所需的IP位址
每個節點一個網路介面	<ul style="list-style-type: none"><li>• 每個節點兩（2）個：<ul style="list-style-type: none"><li>◦ MI/CIP：每個節點的管理介面上有一（1）個預先設定的IP位址</li><li>◦ 資料IP：每個節點的一（1）個未使用或未設定的IP位址、與MIP位於同一子網路中</li></ul></li></ul>
每個節點有兩個網路介面	<ul style="list-style-type: none"><li>• 每個節點三個：<ul style="list-style-type: none"><li>◦ MIP：每個節點的管理介面上有一（1）個預先設定的IP位址</li><li>◦ CIP：每個節點的資料介面上有一（1）個預先設定的IP位址、與MIP位於不同的子網路中</li><li>◦ 資料IP：在CIP所在的同一子網路中、每個節點有一（1）個未使用或未設定的IP位址</li></ul></li></ul>



這些組態不使用VLAN標記。

## 防火牆考量

在強制執行網路防火牆流量篩選的環境中、防火牆必須設定為允許傳入流量進入下表所列的連接埠和傳輸協定。「IP位址」欄使用下列縮寫：

- MIP：每個節點的管理介面上的主要IP位址
- CIP：每個節點叢集介面上的主要IP位址
- Dip：在節點上設定一或多個資料IP位址
- MVIP：在一個叢集節點上設定的管理虛擬IP位址

連接埠/傳輸協定	IP 位址	目的	附註
111/TCP	突降	NFS	資料IP會在執行時間在叢集節點之間移動。每個節點都應該允許所有資料IP（或整個子網路）的此流量。
442/TCP	MIP	API	
635/TCP	突降	NFS	資料IP會在執行時間在叢集節點之間移動。每個節點都應該允許所有資料IP（或整個子網路）的此流量。
2010/udp	CIP	叢集/節點探索	包括傳入和傳出udp連接埠2010的單點傳播和廣播流量、包括回覆。
2049/ TCP	突降	NFS	資料IP會在執行時間在叢集節點之間移動。每個節點都應該允許所有資料IP（或整個子網路）的此流量。
2181-2183/TCP	CIP	分散式通訊	
2443/TCP	MIP	API	
2443/TCP	MVIP	API	MVIP位址可由任何叢集節點代管、並在需要時於執行時間重新定位。
4000至4006/TCP	CIP	叢集內RPC	
4045-4046/TCP	突降	NFS	資料IP會在執行時間在叢集節點之間移動。每個節點都應該允許所有資料IP（或整個子網路）的此流量。
7700/TCP	CIP	工作階段管理程式	
9919/TCP	MIP	DMS API	
9920/TCP	突降	DMS REST伺服器	
ICMP	CIP + DIP	節點內通訊、健全狀況檢查	資料IP會在執行時間在叢集節點之間移動。每個節點都應該允許所有資料IP（或整個子網路）的此流量。

## Astra Trident

Astra Data Store要求應用程式Kubernetes叢集執行Astra Trident以進行儲存資源配置和協調。支援Astra Trident 版本210.1至22.04。Astra資料儲存區可設定為 **"儲存後端"** 使用Astra Trident來配置持續磁碟區。



## Container網路介面

Astra Data Store已通過下列Container Network Interfaces（Container Network Interfaces、簡稱CNI）的驗證。

- 適用於RKE叢集的Calico
- Calico和Weave Net CNI適用於香草Kubernetes叢集
- 適用於Red Hat OpenShift Container Platform（OCP）的OpenShift SDN
- Google Anthos的Cilium



- 隨Cilium CNI部署的Astra Data Store需要Portmap外掛程式、才能支援hostPort。您可以將「CNI鏈結模式：portmap」新增至cilium組態地圖、然後重新啟動Cilium Pod、以啟用CNI鏈結模式。
- 只有Calico和OpenShift SDN CNI才支援啟用防火牆的組態。

## 授權

Astra Data Store需要有效授權才能啟用完整功能。

["請在此註冊"](#) 以取得Astra Data Store授權。下載授權的指示將會在您註冊後寄送給您。

## 下一步

檢視 ["快速入門"](#) 總覽：


## 以取得更多資訊

["Astra資料儲存區的限制"](#)

# Astra Data Store快速入門

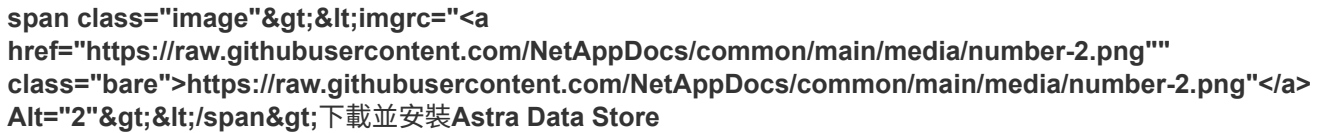
本頁提供Astra Data Store入門所需步驟的高階概觀。每個步驟中的連結都會帶您前往提供更多詳細資料的頁面。

歡迎試用！如果您想要試用Astra Data Store、可以使用90天試用版授權。 ["請在此註冊"](#) 取得Astra Data Store授權。

 <https://raw.githubusercontent.com/NetAppDocs/common/main/media/number-1.png>  
<https://raw.githubusercontent.com/NetAppDocs/common/main/media/number-1.png>  
alt="1"></a> 檢閱Kubernetes叢集需求

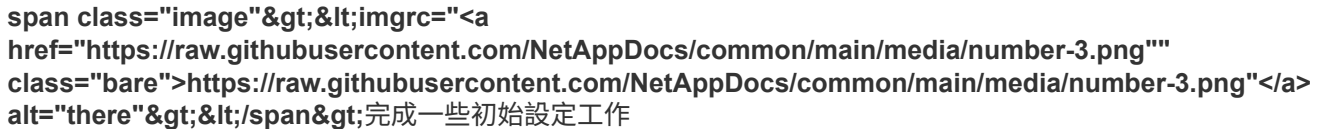
- 叢集必須以正常狀態執行、且至少有四個或更多個工作節點。
- Astra Data Store部署中的每個Kubernetes工作節點、都應該有相同介面類型（SATA、SAS或NVMe）的SSD、以及指派給Astra Data Store叢集的相同磁碟機數量。
- 每個SSD都應有一個唯一的序號。

深入瞭解 ["Astra資料儲存區需求"](#)。

 <https://raw.githubusercontent.com/NetAppDocs/common/main/media/number-2.png>  
Alt="2"></span>下載並安裝Astra Data Store

- 從下載Astra Data Store ["NetApp 支援網站"](#)。
- 在您的本機環境中安裝Astra Data Store。
- 套用Astra Data Store授權。
- 安裝Astra Data Store叢集。

深入瞭解 ["安裝Astra Data Store"](#)。

 <https://raw.githubusercontent.com/NetAppDocs/common/main/media/number-3.png>  
alt="there"></span>完成一些初始設定工作

- 安裝Astra Trident。
- 安裝Kubernetes快照自訂資源定義（CRD）和控制器。
- 將Astra Data Store設定為儲存後端。
- 建立預設的Astra Data Store儲存類別。
- 設定Astra資料儲存區以使用遙測服務。

深入瞭解 ["初始設定程序"](#)。

完成Astra Data Store設定之後、接下來您可以：

- 使用kubectl命令和kubectl astrads擴充功能來管理叢集、包括將節點置於維護模式、更換磁碟機或更換節點等工作。深入瞭解 ["如何搭配Astra Data Store使用kubectl命令"](#)。
- 設定監控端點。深入瞭解 ["設定監控端點"](#)。
- ["使用Astra Data Store搭配VMware"](#)。

["安裝Astra Data Store"](#)。

## 安裝Astra Data Store

您可以使用Kubernetes原生命令或使用Astra Control Center中的UI來安裝Astra Data Store。

### 安裝選項

- 使用**Kubernetes**原生命令：若要使用Kubernetes原生命令安裝Astra Data Store、請完成中所述的安裝步驟[此程序](#)。
- 使用**Astra Control Center**：若要使用Astra Control Center安裝Astra Data Store、請完成中所述的安裝步驟[此程序](#)。

### 您的需求

- ["開始安裝之前、請先準備好環境以進行Astra Data Store部署"](#)。

- 存取 ["NetApp 支援網站"](#)。"註冊" 如果您尚未擁有完整存取權限的NetApp支援網站帳戶、請下載試用版。
- 答 ["NetApp授權檔案 \(NLF\)"](#) 適用於Astra資料儲存區。下載授權的指示將會在您完成後寄送給您 ["註冊"](#)。
- 具有作用中內容叢集管理權限的作用中Kbeconfig。
- 瞭解 ["角色與權限"](#) 由Astra Data Store使用。

## 安裝Astra Data Store

標準Kubernetes叢集的Astra Data Store安裝程序會引導您完成下列高層級步驟。此外、也會說明Red Hat OpenShift Container Platform (OCP) 環境的其他安裝步驟。

- [\[Download the Astra Data Store bundle and extract the images\]](#)
- [\[Copy the binary and push images to your local registry\]](#)
- [\[OpenShift procedure\]](#)
- [\[Install the Astra Data Store operator\]](#)
- [\[Deploy the Astra Data Store version YAML\]](#)
- [\[Apply the Astra Data Store license\]](#)
- [\[Install the Astra Data Store cluster\]](#)
- [\[Understand deployment-related events\]](#)



若要搭配Google Anthos使用Astra Data Store、請完成這些安裝步驟、並將Astra Data Store叢集新增至您的Anthos環境。



若要在Rancher RKE環境中安裝Astra Data Store、請完成下列安裝步驟、並以Rancher命令取代KECBEctl命令。

### 下載Astra Data Store產品組合並擷取映像

1. 登入 ["NetApp 支援網站"](#) 並下載Astra Data Store套裝組合（「Astra\_Data\_Store\_2022.05.tar」）。



如果您要尋找舊版套裝組合的說明、請參閱 ["該版本的文件"](#)。

2. （可選）使用以下命令驗證套件的簽名：

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

3. 建立目錄：

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

4. 擷取影像：

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



影像將擷取至工作目錄中建立的「astrads/images/」目錄。

複製二進位檔並將映像推送至本機登錄

1. 從您用來擷取映像的目錄、將Kkbecl-astra二進位檔複製到安裝k8s kubernetes二進位檔的標準路徑（以下範例中使用「usr/bin/」）。Kustbecl-astrads是自訂的Kvecl擴充功能、可安裝及管理Astra Data Store叢集。



使用「which kubectll」命令尋找安裝kubectll二進位檔的路徑。

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

2. 將Astra Data Store映像目錄中的檔案新增至本機登錄。



請參閱以下自動載入影像的範例指令碼。

- a. 登入您的登錄：

```
docker login [your_registry_path]
```

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

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

- c. 執行指令碼、將影像載入Docker、標記影像、然後[Subforte\_image\_local\_register\_pip]將影像推送到本機登錄：

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

## OpenShift程序

下列程序僅適用於Red Hat OpenShift Container Platform（OCP）上的部署。在非OCP Kubernetes叢集上部署時、請跳過此程序。

- 建立命名空間以部署Astra Data Store
- 建立自訂的SCC
- 建立角色與角色繫結

## 範例 1. 詳細資料

建立名稱空間「astrads系統」、以便安裝所有Astra Data Store元件。

只有在Red Hat OpenShift Container Platform (OCP) 上部署時才需要執行下列步驟。

### 1. 建立命名空間：

```
kubectl create -f ads_namespace.yaml
```

範例：ads\_names.yaml

```
apiVersion: v1
kind: Namespace
metadata:
  labels:
    control-plane: operator
  name: astrads-system
```

OpenShift使用安全內容限制 (SCC) 來控制Pod可以執行的動作。根據預設、任何容器的執行都會被授予受限的SCC、而且只會授予該SCC所定義的功能。

受限的SCC不提供Astra Data Store叢集Pod所需的權限。使用此程序可提供Astra Data Store所需的權限（列於範例中）。

將自訂SCC指派給Astra Data Store命名空間的預設服務帳戶。

只有在Red Hat OpenShift Container Platform (OCP) 上部署時、才需要執行下列步驟。

### 1. 建立自訂SCC：

```
kubectl create -f ads_privileged_scc.yaml
```

範例：ads\_特權\_scc.yaml

```
allowHostDirVolumePlugin: true
allowHostIPC: true
allowHostNetwork: true
allowHostPID: true
allowHostPorts: true
allowPrivilegeEscalation: true
allowPrivilegedContainer: true
allowedCapabilities:
- '*'
allowedUnsafeSysctls:
- '*'
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  type: RunAsAny
groups: []
kind: SecurityContextConstraints
metadata:
  annotations:
    kubernetes.io/description: 'ADS privileged. Grant with caution.'
    release.openshift.io/create-only: "true"
  name: ads-privileged
priority: null
readOnlyRootFilesystem: false
requiredDropCapabilities: null
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: RunAsAny
seccompProfiles:
- '*'
supplementalGroups:
  type: RunAsAny
users:
- system:serviceaccount:astrads-system:default
volumes:
- '*'
```

## 2. 使用「occ Get SCC」命令顯示新增的SCC：

```
# oc get scc/ads-privileged
NAME          PRIV  CAPS  SELINUX  RUNASUSER  FSGROUP  SUPGROUP
PRIORITY      READONLYROOTFS  VOLUMES
ads-privileged true  ["*"] RunAsAny RunAsAny RunAsAny RunAsAny
<no value> false          ["*"]
#
```

建立Astra Data Store預設服務帳戶所需的角色和角色繫結。

下列yaml定義會指派「astrads.netapp.io` API」群組中Astra Data Store資源所需的各種角色（透過角色繫結）。

只有在Red Hat OpenShift Container Platform（OCP）上部署時、才需要執行下列步驟。

#### 1. 建立定義的角色和角色繫結：

```
kubectl create -f oc_role_bindings.yaml
```

範例：oc\_role\_bindings.yaml

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: privcrole
rules:
- apiGroups:
  - security.openshift.io
  resourceNames:
  - ads-privileged
  resources:
  - securitycontextconstraints
  verbs:
  - use
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: default-scc-rolebinding
  namespace: astrads-system
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: privcrole
subjects:
```



```

- kind: ServiceAccount
  name: default
  namespace: astrads-system
---

apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: ownerref
  namespace: astrads-system
rules:
- apiGroups:
  - astrads.netapp.io
  resources:
  - '*/finalizers'
  verbs:
  - update
---

apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: or-rb
  namespace: astrads-system
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: ownerref
subjects:
- kind: ServiceAccount
  name: default
  namespace: astrads-system

```

## 設定私有映像登錄

對於某些環境而言、您可以將組態變更為從使用機密的私有登錄中擷取映像、做為選用步驟。

1. 除非您已在上一步中建立「astrads系統」命名空間：

```
kubectl create namespace astrads-system
```

2. 建立秘密：

```
kubectl create secret docker-registry <secret-name> -n astrads-system
--docker-server=<registry name> --docker-username= <registry username>
--docker-password=<registry user password>
```

3. 將機密組態資訊新增至服務帳戶：

```
kubectl patch serviceaccount default -p '{"imagePullSecrets": [{"name":
"<secret-name>"}]}' -n astrads-system
```



這些變更將會在您執行時套用 [安裝Astra Data Store營運者](#)。

## 安裝Astra Data Store營運者

1. 列出Astra資料儲存區清單：

```
ls astrads/manifests/*.yaml
```

回應：

```
astrads/manifests/monitoring_operator.yaml
astrads/manifests/astradscluster.yaml
astrads/manifests/astradsversion.yaml
astrads/manifests/astradsoperator.yaml
astrads/manifests/vasa_asup_certs.yaml
astrads/manifests/manifest.yaml
astrads/manifests/configuration.yaml
```

2. 使用「kubectl apply」部署營運者：

```
kubectl apply -f ./astrads/manifests/astradsoperator.yaml
```

回應：



命名空間回應可能會因執行標準安裝或而有所不同 ["OpenShift Container Platform安裝"](#)。

```
namespace/astrads-system created
customresourcedefinition.apiextensions.k8s.io/astradsaddrives.astrads.n
etapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsautosupports.astrad
s.netapp.io created
```

```
customresourcedefinition.apiextensions.k8s.io/astradscLOUDsnapshots.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsclusters.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsexportpolicies.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsfaileddrives.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradskeyproviders.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradslICENSES.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsnfsOPTIONS.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsnodeinfoes.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsnodemANAGEMENTS.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradssqOSPolicies.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradssearkeyrotaterequests.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsversions.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsvolumefiles.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsvolumes.astrads.netapp.io created
customresourcedefinition.apiextensions.k8s.io/astradsvolumesnapshots.astrads.netapp.io created
role.rbac.authorization.k8s.io/astrads-astrads-system-admin-role created
role.rbac.authorization.k8s.io/astrads-astrads-system-reader-role created
role.rbac.authorization.k8s.io/astrads-astrads-system-writer-role created
role.rbac.authorization.k8s.io/astrads-leader-election-role created
role.rbac.authorization.k8s.io/astrads-manager-role created
clusterrole.rbac.authorization.k8s.io/astrads-astrads-admin-clusterrole created
clusterrole.rbac.authorization.k8s.io/astrads-astrads-reader-clusterrole created
clusterrole.rbac.authorization.k8s.io/astrads-astrads-writer-clusterrole created
clusterrole.rbac.authorization.k8s.io/astrads-astradsautosupport-editor-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsautosupport-viewer-
```

```
role created
clusterrole.rbac.authorization.k8s.io/astrads-astradscloudsnapshot-
editor-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradscloudsnapshot-
viewer-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradscluster-editor-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradscluster-viewer-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradsexportpolicy-
editor-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsexportpolicy-
viewer-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsfaileddrive-editor-
role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsfaileddrive-viewer-
role created
clusterrole.rbac.authorization.k8s.io/astrads-astradslicense-editor-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradslicense-viewer-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradsnfsoption-editor-
role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsnfsoption-viewer-
role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsnodeinfo-editor-
role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsnodeinfo-viewer-
role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsnodemangement-
editor-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsnodemangement-
viewer-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsqospolicy-viewer-
role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsversion-editor-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradsversion-viewer-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradsvolume-editor-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradsvolume-viewer-role
created
clusterrole.rbac.authorization.k8s.io/astrads-astradsvolumeeditor-
role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsvolumeeditor-
viewer-
```

```

role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsvolumesnapshot-
editor-role created
clusterrole.rbac.authorization.k8s.io/astrads-astradsvolumesnapshot-
viewer-role created
clusterrole.rbac.authorization.k8s.io/astrads-manager-role created
rolebinding.rbac.authorization.k8s.io/astrads-astrads-admin-rolebinding
created
rolebinding.rbac.authorization.k8s.io/astrads-astrads-reader-rolebinding
created
rolebinding.rbac.authorization.k8s.io/astrads-astrads-writer-rolebinding
created
rolebinding.rbac.authorization.k8s.io/astrads-leader-election-
rolebinding created
rolebinding.rbac.authorization.k8s.io/astrads-manager-rolebinding
created
clusterrolebinding.rbac.authorization.k8s.io/astrads-astrads-admin-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/astrads-astrads-reader-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/astrads-astrads-writer-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/astrads-manager-rolebinding
created
configmap/astrads-autosupport-cm created
configmap/astrads-firetap-cm created
configmap/astrads-kevents-asup created
configmap/astrads-metrics-cm created
secret/astrads-autosupport-certs created
secret/astrads-webhook-server-cert created
service/astrads-webhook-service created
deployment.apps/astrads-operator created

```

### 3. 確認Astra Data Store營運者Pod已啟動且正在執行：

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

回應：

NAME	READY	STATUS	RESTARTS	AGE
astrads-operator-5ffb94fbf-7ln4h	1/1	Running	0	17m

## 部署Astra Data Store版本Yaml

1. 使用「kubectl apply」部署：

```
kubectl apply -f ./astrads/manifests/astradsversion.yaml
```

2. 確認Pod正在執行：

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

回應：

NAME	READY	STATUS	RESTARTS
AGE			
astrads-cluster-controller-7f6f884645-xxf2n	1/1	Running	0
117s			
astrads-ds-nodeinfo-2jqnk	1/1	Running	0
2m7s			
astrads-ds-nodeinfo-dbk7v	1/1	Running	0
2m7s			
astrads-ds-nodeinfo-rn9tt	1/1	Running	0
2m7s			
astrads-ds-nodeinfo-vsmhv	1/1	Running	0
2m7s			
astrads-license-controller-fb8fd56bc-bxq7j	1/1	Running	0
2m2s			
astrads-operator-5ffb94fbf-7ln4h	1/1	Running	0
2m10s			

## 套用Astra Data Store授權

1. 套用您從NetApp取得的NetApp授權檔案（NLF）。執行命令之前、請輸入您所在的叢集名稱（「<Astra Data-Store-cluster名稱>」） [即將部署](#) 或已部署及授權檔案路徑（「<file\_path/file.txt>」）：

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

2. 確認已新增授權：

```
kubectl astrads license list
```

回應：

NAME	ADSCUSTER	VALID	PRODUCT
EVALUATION ENDDATE	VALIDATED		
e100000006-ads-capacity	astrads-example-cluster	true	Astra Data Store
true	2023-01-23	2022-04-04T14:38:54Z	

## 安裝Astra Data Store叢集

### 1. 開啟Yaml檔案：

```
vim ./astrads/manifests/astradscluster.yaml
```

### 2. 在Yaml檔案中編輯下列值。



以下步驟為Yaml檔案的簡化範例。

- (必填) 中繼資料：在「metadata」中、將「name」字串變更為叢集名稱。這必須與您在使用時使用的叢集名稱相同 [套用授權](#)。
- (必填) 規格：在「show」中變更下列必要值：
  - 視授權和Astra Data Store安裝大小而定、將「adsNodeConfig」值變更為安裝所需的值：
    - 小型：9個CPU和38個記憶體
    - 中：23個CPU和94個記憶體
  - (選用) 移除「adsNodeSelector」區段的註解。如果您想要限制Astra Data Store只安裝在選取的工作節點集區、請設定此選項。
  - (選用) 指定Astra Data Store叢集應使用的特定節點數、範圍介於4-16之間。
  - 將「mVIP」字串變更為可從叢集中任何工作節點路由傳送之浮動管理IP的IP位址。
  - 在「adsDataNetworks」中、新增一個以逗號分隔的浮動IP位址清單（「Addresses」）、這些位址可從您要掛載NetApp Volume的任何主機路由傳送。每個節點使用一個浮動IP位址。資料網路IP位址應至少與Astra Data Store節點一樣多。對於Astra Data Store、這表示如果您打算稍後擴充叢集、至少需要4個位址或最多16個位址。
  - 在「adsDataNetworks」中、指定資料網路使用的網路遮罩。
  - 在「adsNetworks介面」中、將「<mgmt\_interface\_name>」和「<cluster與\_storage介面名稱>」值取代為您要用於管理、叢集與儲存的網路介面名稱。如果未指定名稱、則節點的主要介面將用於管理、叢集和儲存網路。請務必移除「adsNetworks介面」一節的註解。



叢集和儲存網路必須位於相同的介面上。Astra Data Store管理介面應與Kubernetes節點的管理介面相同。

- (選用) 監控組態：如果您要設定 [監控營運者](#)（若您未使用Astra Control Center進行監控、則為選用）、從區段移除註解、新增套用代理程式CR（監控操作員資源）的命名空間（預設為「NetApp-Monitoring」（NetApp監控）、並新增您在先前步驟中使用的登錄（「您的登錄路徑」）的repo路徑。
- (可選) \* autosupSupportConfig\*：保留 "AutoSupport" 除非您需要設定Proxy、否則預設值為：

- 對於「proxyURL」、請使用連接埠來設定Proxy的URL、以便AutoSupport 進行套裝組合傳輸。



為了簡單起見、以下Yaml範例中已移除部分意見。

```
apiVersion: astrads.netapp.io/v1beta1
kind: AstraDSCluster
metadata:
  name: astrads-cluster-name
  namespace: astrads-system
spec:
  adsNodeConfig:
    cpu: 9
    memory: 38
    # [Optional] Specify node selector labels to select the nodes for
    # creating ADS cluster
    # adsNodeSelector:
    #   matchLabels:
    #     customLabelKey: customLabelValue
  adsNodeCount: 4
  mvip: ""
  adsDataNetworks:
    - addresses: ""
      netmask:
        # Specify the network interface names to use for management, cluster
        # and storage networks.
        # If none are specified, the node's primary interface will be used for
        # management, cluster and storage networking.
        # To move the cluster and storage networks to a different interface
        # than management, specify all three interfaces to use here.
        # NOTE: The cluster and storage networks need to be on the same
        # interface.
  adsNetworkInterfaces:
    managementInterface: "<mgmt_interface_name>"
    clusterInterface: "<cluster_and_storage_interface_name>"
    storageInterface: "<cluster_and_storage_interface_name>"
    # [Optional] Provide a monitoring config to be used to setup/configure
    # a monitoring agent.
  # monitoringConfig:
  #   # namespace: "netapp-monitoring"
  #   # repo: "[YOUR REGISTRY]"
  autoSupportConfig:
    autoUpload: true
    enabled: true
    coredumpUpload: false
    historyRetentionCount: 25
    destinationURL: "https://support.netapp.com/put/AsupPut"
```



```
# ProxyURL defines the URL of the proxy with port to be used for
AutoSupport bundle transfer
# proxyURL:
periodic:
  - schedule: "0 0 * * *"
    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
```

3. 使用「kubectl apply」部署叢集：

```
kubectl apply -f ./astrads/manifests/astradscluster.yaml
```

4. 等待幾分鐘、讓叢集建立作業完成、然後確認Pod正在執行：

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

回應範例：

NAME	READY	STATUS	
RESTARTS      AGE			
astrads-cluster-controller-7c67cc7f7b-2jww2 7h31m	1/1	Running	0
astrads-deployment-support-788b859c65-2qjkn 12d	3/3	Running	19
astrads-ds-astrads-cluster-1ab0dbc-j9jzc 5d2h	1/1	Running	0
astrads-ds-astrads-cluster-1ab0dbc-k9wp8 5d1h	1/1	Running	0
astrads-ds-astrads-cluster-1ab0dbc-pwk42 5d2h	1/1	Running	0
astrads-ds-astrads-cluster-1ab0dbc-qhvc6 8h	1/1	Running	0
astrads-ds-nodeinfo-gcmj8 12d	1/1	Running	1
astrads-ds-nodeinfo-j826x 12d	1/1	Running	3
astrads-ds-nodeinfo-vdthh 12d	1/1	Running	3
astrads-ds-nodeinfo-xwgsf 12d	1/1	Running	0
astrads-ds-support-828vw 5d2h	2/2	Running	2
astrads-ds-support-astrads-example-cluster-cfzts 8h	2/2	Running	0
astrads-ds-support-astrads-example-cluster-nzkkr 7h49m	2/2	Running	15
astrads-ds-support-astrads-example-cluster-xxbnp 5d2h	2/2	Running	1
astrads-license-controller-86c69f76bb-s6fb7 8h	1/1	Running	0
astrads-operator-79ff8fbb6d-vpz9m 8h	1/1	Running	0

## 5. 驗證叢集部署進度：

```
kubectl get astradscluster -n astrads-system
```

回應範例：

NAME	STATUS	VERSION	SERIAL NUMBER	MVIP
AGE				
astrads-example-cluster	created	2022.05.0-X	e100000006	
10.x.x.x	13m			

## 瞭解與部署相關的事件

在叢集部署期間、作業狀態應從「空白」變更為「進行中」、改為「已建立」。叢集部署將持續約8至10分鐘。若要在部署期間監控叢集事件、您可以執行下列任一命令：

```
kubectl get events --field-selector involvedObject.kind=AstraDSCluster -n astrads-system
```

```
kubectl describe astradscluster <cluster name> -n astrads-system
```

以下是部署期間的重要事件：

活動	訊息與重要性
控制面板無選項	成功選取[number]個控制面板節點以加入ADS叢集。Astra Data Store營運者利用CPU、記憶體、儲存設備和網路來識別足夠的節點、以建立Astra Data Store叢集。
ADSClusterCreateInProgress	Astra Data Store叢集控制器已啟動叢集建立作業。
ADSClusterCreeSuccess	已成功建立叢集。

如果叢集的狀態並未變更為「進行中」、請查看操作員記錄、以取得節點選擇的詳細資料：

```
kubectl logs -n astrads-system <astrads operator pod name>
```

如果叢集的狀態卡在「In Progress（進行中）」、請檢查叢集控制器的記錄：

```
kubectl logs -n astrads-system <astrads cluster controller pod name>
```

## 使用Astra Control Center安裝Astra資料儲存區

若要搭配Astra Control Center部署及使用Astra Data Store、請執行下列步驟。

您需要的是 **#8217** ；需要的是什麼

- 您已檢閱過 [一般Astra資料儲存區先決條件](#)。

- 您已安裝Astra Control Center。

#### 步驟

1. ["使用Astra Control Center部署Astra資料儲存區"](#)。

#### 下一步

- \* Kubernetes原生部署與協力廠商發佈\*：執行其他作業來完成Astra Data Store部署 ["設定工作"](#)。
- \* Astra控制中心\*：如果您已使用Astra控制中心來部署Astra資料儲存區、則不需要遵循這些步驟 ["設定工作"](#) 除非您想設定任何其他監控選項。部署Astra Data Store之後、您可以使用Astra Control Center UI來完成下列工作：
  - ["監控Astra Data Store資產的健全狀況"](#)。
  - ["管理Astra Data Store後端儲存設備"](#)。
  - ["監控節點、磁碟和持續磁碟區宣告 \(PVCS\) "](#)。

## 設定Astra Data Store元件

您就可以了 ["安裝獨立式Astra Data Store"](#) 並解決一些問題 ["環境先決條件"](#) 您將安裝Astra Trident、設定Kubernetes快照功能、設定儲存後端、以及建立預設儲存類別：



如果您使用Astra Control Center來部署Astra Data Store、除非您打算設定、否則不需要執行這些步驟 [其他監控選項](#)。

- [\[Install Astra Trident\]](#)
- [\[Install Kubernetes snapshot CRDs and Controller\]](#)
- [\[Set up Astra Data Store as storage backend\]](#)
- [\[Create a default Astra Data Store storage class\]](#)
- [\[Configure Astra Data Store monitoring\]](#)

## 安裝Astra Trident

對於Astra Data Store、您需要安裝Astra Trident 21.10.1或更新版本。您可以使用下列其中一個選項來安裝Astra Trident：

- ["使用tridentctl安裝Astra Trident"](#)。
- ["使用Trident操作員安裝Astra Trident"](#)。



您可以手動或使用Helm來部署Trident運算子。

## 安裝Kubernetes Snapshot客戶需求日和控制器

需要Kubernetes快照CRD和控制器才能建立持續磁碟區宣告 (PVR) 快照。如果您的環境尚未安裝CRD和控制器、請執行下列命令來安裝。



以下命令範例假設為目錄「/trident」、但您使用的目錄可以是任何您用來下載Yaml檔案的目錄。

您需要的是 **#8217** ；需要的是什麼

- "開始安裝之前、請先準備好環境以進行Astra Data Store部署"。
- 下載 "Kubernetes快照控制器Yaml檔案"：
  - 設定快照控制器.yaml
  - RBAC快照控制器.yaml
- 下載 "Y反洗錢客戶需求日"：
  - Snapshot.storage ◦ k8s.io\_volumesnapshotclasses.yaml
  - Snapshot.storage ◦ k8s.io\_volumesnapshotcontents ◦ yaml
  - Snapshot.storage : k8s.io\_volumesnapshots.yaml

#### 步驟

1. 套用snapshot.storage ◦ k8s.io\_volumesnapshotclasses.yaml：

```
kubectl apply -f
trident/snapshot.storage.k8s.io_volumesnapshotclasses.yaml
```

回應：

```
customresourcedefinition.apiextensions.k8s.io/volumesnapshotclasses.snapshot.storage.k8s.io configured
```

2. 套用snapshot.storage ◦ k8s.io\_volumesnapshotcontents ◦ yaml：

```
kubectl apply -f
trident/snapshot.storage.k8s.io_volumesnapshotcontents.yaml
```

回應：

```
customresourcedefinition.apiextensions.k8s.io/volumesnapshotcontents.snapshot.storage.k8s.io configured
```

3. 套用snapshot.storage ◦ k8s.io\_volumesnapshots.yaml：

```
kubectl apply -f trident/snapshot.storage.k8s.io_volumesnapshots.yaml
```

回應：

```
customresourcedefinition.apiextensions.k8s.io/volumesnapshots.snapshot.s  
torage.k8s.io configured
```

4. 套用setup-snapshot控制器.yaml：

```
kubectl apply -f trident/setup-snapshot-controller.yaml
```

回應：

```
deployment.apps/snapshot-controller configured
```

5. 套用RBAC快照控制器.yaml：

```
kubectl apply -f trident/rbac-snapshot-controller.yaml
```

回應：

```
serviceaccount/snapshot-controller configured  
clusterrole.rbac.authorization.k8s.io/snapshot-controller-runner  
configured  
clusterrolebinding.rbac.authorization.k8s.io/snapshot-controller-role  
configured  
role.rbac.authorization.k8s.io/snapshot-controller-leaderelection  
configured  
rolebinding.rbac.authorization.k8s.io/snapshot-controller-leaderelection  
configured
```

6. 確認已套用CRD Y反 洗錢檔案：

```
kubectl get crd | grep volumesnapshot
```

回應範例：

```
astradsvolumesnapshots.astrads.netapp.io      2021-08-04T17:48:21Z
volumesnapshotclasses.snapshot.storage.k8s.io 2021-08-04T22:05:49Z
volumesnapshotcontents.snapshot.storage.k8s.io 2021-08-04T22:05:59Z
volumesnapshots.snapshot.storage.k8s.io       2021-08-04T22:06:17Z
```

## 7. 確認已套用快照控制器檔案：

```
kubectl get pods -n kube-system | grep snapshot
```

回應範例：

```
snapshot-controller-7f58886ff4-cdh78
1/1      Running    0          13s
snapshot-controller-7f58886ff4-tmrd9
1/1      Running    0          32s
```

## 將Astra Data Store設定為儲存後端

在ads\_backend.json檔案中設定儲存後端參數、然後建立Astra Data Store儲存後端。

### 步驟

#### 1. 使用安全終端建立「ads\_backend.json」：

```
vi ads_backend.json
```

#### 2. 設定Json檔案：



Json範例遵循下列步驟。

- 將「叢集」值變更為Astra Data Store叢集的叢集名稱。
- 將「命名空間」值變更為您要用於建立磁碟區的命名空間。
- 除非您針對此後端設定匯出原則CR,否則請將「AutoExportPolicy」值變更為「true」。
- 將您要授予存取權的IP位址填入「AutoExportCIDR」清單。使用「0.00.0.0/0」允許所有人。
- 如需「kubeconfig」值、請執行下列動作：
  - 將.kube /組態Yaml檔案轉換為Json格式、並將其最小化、不含空格：

轉換範例：

```
python3 -c 'import sys, yaml, json;
json.dump(yaml.load(sys.stdin), sys.stdout, indent=None)' <
~/.kube/config > kubeconf.json
```

ii. 將編碼為基64、並使用基64輸出作為「kubeconfig」值：

編碼範例：

```
cat kubeconf.json | base64 | tr -d '\n'
```



```

{
  "version": 1,
  "storageDriverName": "astrads-nas",
  "storagePrefix": "",
  "cluster": "example-1234584",
  "namespace": "astrads-system",
  "autoExportPolicy": true,
  "autoExportCIDRs": ["0.0.0.0/0"],
  "kubeconfig": "<base64_output_of_kubeconf_json>",
  "debugTraceFlags": {"method": true, "api": true},
  "labels": {"cloud": "on-prem", "creator": "trident-dev"},
  "defaults": {
    "qosPolicy": "silver"
  },
  "storage": [
    {
      "labels": {
        "performance": "extreme"
      },
      "defaults": {
        "qosPolicy": "gold"
      }
    },
    {
      "labels": {
        "performance": "premium"
      },
      "defaults": {
        "qosPolicy": "silver"
      }
    },
    {
      "labels": {
        "performance": "standard"
      },
      "defaults": {
        "qosPolicy": "bronze"
      }
    }
  ]
}

```

3. 切換到您下載Trident安裝程式的目錄：

```
cd <trident-installer or path to folder containing tridentctl>
```

#### 4. 建立儲存後端：

```
./tridentctl create backend -f ads_backend.json -n trident
```

回應範例：

```
+-----+-----+
+-----+-----+-----+
|      NAME      | STORAGE DRIVER |                      UUID
| STATE  | VOLUMES |
+-----+-----+
+-----+-----+-----+
| example-1234584 | astrads-nas    | 2125fa7a-730e-43c8-873b-
6012fcc3b527 | online |          0 |
+-----+-----+
+-----+-----+-----+
```

## 建立預設的Astra Data Store儲存類別

建立Astra Trident預設儲存類別、並將其套用至儲存後端。

### 步驟

#### 1. 建立Trident - csi儲存類別：

##### a. 建立ads\_sc\_example.yaml：

```
vi ads_sc_example.yaml
```

範例：

```

allowVolumeExpansion: true
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  creationTimestamp: "2022-05-09T18:05:21Z"
  name: ads-silver
  resourceVersion: "3361772"
  uid: 1o023456-da4b-51e3-b430-3aa1e3bg111a
mountOptions:
- vers=4
parameters:
  backendType: astrads-nas
  selector: performance=premium
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate

```

b. 建立Trident - csi :

```
kubectl create -f ads_sc_example.yaml
```

回應：

```
storageclass.storage.k8s.io/trident-csi created
```

2. 確認已新增儲存類別：

```
kubectl get storageclass
```

回應：

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE
ads-silver	csi.trident.netapp.io	Delete	Immediate
allowVolumeExpansion	AGE		
true	6h29m		

3. 切換到您下載Trident安裝程式的目錄：

```
cd <trident-installer or path to folder containing tridentctl>
```

#### 4. 確認Astra Trident後端已使用預設的儲存類別參數更新：

```
./tridentctl get backend -n trident -o yaml
```

回應範例：

```
items:
- backendUUID: 2125fa7a-730e-43c8-873b-6012fcc3b527
  config:
    autoExportCIDRs:
    - 0.0.0.0/0
    autoExportPolicy: true
    backendName: ""
    cluster: example-1234584
    credentials: null
    debug: false
    debugTraceFlags:
      api: true
      method: true
    defaults:
      exportPolicy: default
      qosPolicy: bronze
      size: 1G
      snapshotDir: "false"
      snapshotPolicy: none
    disableDelete: false
    kubeconfig: <ID>
    labels:
      cloud: on-prem
      creator: trident-dev
    limitVolumeSize: ""
    namespace: astrads-system
    nfsMountOptions: ""
    region: ""
    serialNumbers: null
    storage:
    - defaults:
        exportPolicy: ""
        qosPolicy: gold
        size: ""
        snapshotDir: ""
        snapshotPolicy: ""
      labels:
        performance: extreme
      region: ""
```

```

supportedTopologies: null
zone: ""
- defaults:
  exportPolicy: ""
  qosPolicy: silver
  size: ""
  snapshotDir: ""
  snapshotPolicy: ""
labels:
  performance: premium
region: ""
supportedTopologies: null
zone: ""
- defaults:
  exportPolicy: ""
  qosPolicy: bronze
  size: ""
  snapshotDir: ""
  snapshotPolicy: ""
labels:
  performance: standard
region: ""
supportedTopologies: null
zone: ""
storageDriverName: astrads-nas
storagePrefix: ""
supportedTopologies: null
version: 1
zone: ""
configRef: ""
name: example-1234584
online: true
protocol: file
state: online
storage:
  example-1234584_pool_0:
    name: example-1234584_pool_0
    storageAttributes:
      backendType:
        offer:
          - astrads-nas
      clones:
        offer: true
      encryption:
        offer: false
    labels:

```

```

    offer:
      cloud: on-prem
      creator: trident-dev
      performance: extreme
  snapshots:
    offer: true
storageClasses:
- trident-csi
supportedTopologies: null
example-1234584_pool_1:
  name: example-1234584_pool_1
  storageAttributes:
    backendType:
      offer:
        - astrads-nas
    clones:
      offer: true
    encryption:
      offer: false
    labels:
      offer:
        cloud: on-prem
        creator: trident-dev
        performance: premium
    snapshots:
      offer: true
storageClasses:
- trident-csi
supportedTopologies: null
example-1234584_pool_2:
  name: example-1234584_pool_2
  storageAttributes:
    backendType:
      offer:
        - astrads-nas
    clones:
      offer: true
    encryption:
      offer: false
    labels:
      offer:
        cloud: on-prem
        creator: trident-dev
        performance: standard
    snapshots:
      offer: true

```

```
storageClasses:
  - ads-silver
supportedTopologies: null
volumes: []
```

## 設定Astra Data Store監控

（選用）您可以設定Astra Data Store、以供其他遙測服務進行監控。如果您未使用Astra Control Center進行Astra Data Store監控、或想要將監控延伸至其他端點、建議您執行此程序。

如果您的Astra Data Store執行個體是獨立部署、使用Cloud Insights 支援程序來監控遙測、或是將記錄串流至第三方端點（例如Elastic）、您可以安裝監控操作員。



針對Astra Control Center部署、系統會自動設定監控操作員。您可以跳過下列程序的前兩個命令。

在設定監控之前、您需要在「astrads系統」命名空間中使用作用中的Astra資料儲存叢集。

### 步驟

1. 執行此安裝命令：

```
kubectl apply -f ./manifests/monitoring_operator.yaml
```

2. 設定Astra資料儲存區以進行監控：

```
kubectl astrads monitoring -n netapp-monitoring -r [YOUR REGISTRY] setup
```

3. 設定Astra Data Store將EMS記錄串流至彈性端點：

```
kubectl astrads monitoring es --port <portname> --host <hostname>
```

## Astra Data Store Early Access Program (AP) 版本限制

Astra Data Store在早期存取方案期間有下列資源限制。

資源	最低	最大值
Astra Data Store叢集中的節點數	4.	16
每個節點的持續磁碟區數目	不適用	50
Volume大小	20億	2TiB
每個Volume的快照	0	1023

資源	最低	最大值
每個Volume的複本	0	9.
每個節點的VM數	0	50

## Astra Data Store的常見問題集

尋找有關Astra Data Store Early Access Program版本安裝、設定、升級及疑難排解的常見問題解答。

### 一般問題

\*我可以使用Astra Data Store Early Access Program正式作業嗎？\*不可以雖然Astra Data Store是專為提供企業級恢復能力而設計與開發、但Astra Data Store的早期存取方案版本並不適用於正式作業工作負載。

\*我可以將Astra Data Store用於虛擬機器工作負載嗎？\*是。Astra Data Store同時支援Kubernetes和VMware vVol工作負載。

請參閱 ["瞭解Astra Data Store與VMware的相關資訊"](#)。

\*我可以使用VMware vSphere來管理Astra Data Store嗎？\*是的、Astra Data Store可以在vCenter內使用適用於VMware vSphere的NetApp Astra外掛程式進行原生管理。請參閱 ["管理VMware安裝的Astra Data Store元件"](#)。

- Astra Data Store是否因其運作而與其他NetApp產品有任何相依關係？\*

是的。Astra Data Store需要NetApp SCSI驅動程式Astra Trident版本210.1及更新版本、才能部署在工作負載Kubernetes叢集上。深入瞭解 ["Astra資料儲存區需求"](#)。

Astra Control Center是啟用VMware工作流程和適用於VMware vSphere的NetApp Astra外掛程式的必要條件。

將Astra Data Store叢集當作儲存後端的應用程式可以使用 ["Astra控制中心"](#) 運用應用程式感知資料管理功能、包括資料保護、災難恢復、以及Kubernetes工作負載移轉。

\*我該如何管理Astra Data Store叢集？\*您可以使用kubectl命令及Kubernetes API副檔名來管理Astra Data Store資源。

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

\*我該如何監控Astra Data Store叢集指標？\*您可以使用Cloud Insights 支援功能來監控Astra Data Store指標（請參閱 ["利用功能表監控指標Cloud Insights"](#)）或Kubernetes原生監控（請參閱 ["使用Prometheus和Grafana進行監控"](#)）。

您也可以監控記錄。請參閱 ["設定及監控事件記錄"](#)。

\*我可以在ONTAP Kubernetes叢集中、搭配使用Astra Data Store和其他的儲存供應商嗎？\*可以。Astra Data Store可與應用程式叢集中的其他儲存供應商一起使用。

\*如果我從Astra Data Store移除Kubernetes叢集、是否會解除安裝Astra Trident？\*如果您解除安裝Astra Data Store、則不會從叢集解除安裝Astra Trident。如果您需要解除安裝Astra Trident、則必須個別執行。



## 授權

- Astra Data Store是否需要授權？\*是的、Astra Data Store需要評估版NetApp授權檔案（NLF）才能執行早期存取方案。

請參閱 ["Astra資料儲存區需求"](#)。

- Astra Data Store試用版授權的有效時間有多長？\* Astra Data Store授權的預設期限為下載日期起算的90天。

## 在Kubernetes叢集上安裝及使用Astra Data Store

\*我可以在裸機或虛擬機器上執行的Kubernetes叢集上安裝Astra Data Store嗎？\*可以。Astra Data Store可安裝在執行裸機或vSphere VM的Kubernetes叢集上。請參閱 ["Astra資料儲存區需求"](#)。

- Astra Data Store支援的Kubernetes版本有哪些？\*

Astra Data Store可搭配與v1.20及更新版本相容的Kubernetes發佈產品。不過、目前並未針對所有Kubernetes配送進行驗證。深入瞭解 ["Astra資料儲存區需求"](#)。

\*我的Kubernetes叢集大於4個工作節點。我可以在其中安裝Astra Data Store嗎？\*是。Astra Data Store叢集一開始必須部署在Kubernetes叢集中的4個工作節點上。部署之後、您可以將叢集擴充至最多16個工作者節點。

- Astra Data Store是否支援從私有登錄進行離線安裝？\*是。Astra Data Store可從本機登錄離線安裝。請參閱 ["安裝Astra Data Store"](#)。

\*我需要網際網路連線才能使用Astra Data Store嗎？\*不需要、Astra Data Store Early Access Program不需要網際網路連線。不過、建議您連線至NetApp AutoSupport 功能後端、以便定期傳送遙測套件。這種連線方式可以是直接連線、也可以透過Proxy連線。

- Astra Data Store使用什麼角色和權限？\*您必須是Kube管理員、才能部署Astra Data Store營運者。

Astra Data Store有一組名為「astrads-ds-nodeinfo」的特殊權限取消程式集、可用來探索用於選取節點的節點資源。

此外、營運者將使用特殊權限Kubernetes工作、在所選的工作節點上安裝儲存叢集的容器、以建置Astra Data Store儲存叢集。

\*我需要更新哪些資訊清單檔案才能安裝Astra Data Store？\*請從下載的Astra Data Store套裝組合下載 ["NetApp支援網站"](#)、您將獲得下列資訊清單：

- astradscluster · yaml
- astradsoper.yaml
- astradsversion.yaml
- 監控\_opuler.yaml

您需要使用部署專屬的組態來更新「astradscluster · yaml」資訊清單。請參閱 ["安裝Astra Data Store"](#)。

## 疑難排解與支援

有了Astra Data Store、您可以使用NetApp Container Slack通路來存取社群支援。此通路由NetApp支援與我們

的技術行銷工程師監控。

## "NetApp Container Slack通路"

請參閱 "[Astra Data Store支援營運](#)"。

\*我要如何提出支援案例、或是詢問有關快速問題的說明？\*提出支援案例、或是取得快速問題的說明、並在上回報您的問題或問題 "[NetApp Container Slack通路](#)"。NetApp支援部門將與您密切合作、以盡力提供協助。

\*如何提出新功能的申請？\*如果您對支援的組態或功能有任何疑問、[請聯絡astra.feedback@netapp.com](mailto:astra.feedback@netapp.com)。

\*如何產生支援記錄套裝組合？\*請參閱 "[產生支援服務組合](#)" 以取得有關為Astra Data Store設定及下載支援記錄套裝組合的說明。

- Astra Data Store找不到Kubernetes節點。如何修正此問題？\*請參閱 "[安裝Astra Data Store](#)"。
- IPv6位址是否可用於管理、資料及叢集網路？\*否、Astra Data Store僅支援IPV4位址。未來Astra Data Store版本將新增IPv6支援。

在**Astra Data Store**上配置磁碟區時、會使用什麼**NFS**版本？Astra Data Store支援NFS v4.1、以供Kubernetes應用程式配置的所有磁碟區使用、以及為VMware工作負載配置的所有磁碟區使用NFSv3。

請參閱 "[Astra資料儲存區需求](#)" 和 "[Astra資料儲存區的限制](#)"。

## 升級**Astra Data Store**

\*我可以從Astra Data Store預覽版本升級嗎？\*是。您可以從Astra Data Store Early Access Program版本升級至未來版本。

# 使用Astra資料儲存區

## 使用kvecll命令管理Astra Data Store資源

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

若要瞭解如何部署範例應用程式、請參閱 ["部署測試應用程式"](#)。

如需叢集維護資訊、請參閱 ["管理叢集"](#)。

您需要的是 **#8217** ；需要的是什麼

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

## 列出Astra Data Store的Kubernetes自訂API資源

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

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

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

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

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

回應：

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

2. 若要在Kubernetes叢集中取得所有目前的Astra資料儲存區物件、請使用「kubectl Get ads -A」命令：

```
kubectl get ads -A
```

回應：

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

NAMESPACE	NAME			
STATUS	VERSION	SERIAL NUMBER	MVIP	AGE

```
astrads-system    astradscluster.astrads.netapp.io/astrads-cluster-9f1
created    arda-9.11.1    e000000009    10.224.8.146    46h
```

```
NAMESPACE    NAME
```

```
AGE
```

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

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

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

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

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

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

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

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

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

### 3. 使用其中一個簡短名稱來顯示叢集中磁碟區的目前狀態：

```
kubectl get adsvo -A
```

回應：

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

## 使用KECBEVL副檔名上的說明選項

NetApp為「kubectl」命令提供「astrads」延伸、可讓您執行Astra Data Store叢集管理工作、例如新增授權、管理節點、疑難排解問題、以及擴充Astra Data Store叢集。「astrads」擴充功能包括「-h」選項、提供如何使用擴充功能及執行哪些工作的相關資訊。

### 1. 顯示Astra Data Store「kubectl」擴充中所有命令的說明：

```
kubectl astrads -h
```

回應：

```
A kubectl plugin for inspecting your AstraDS deployment

Usage:
  astrads [command]

Available Commands:
  asup          Manage AutoSupport
  clusters      Manage clusters
  drives        Manage drives in a cluster
  faileddrive   Manage drive replacement in a cluster
  help          Help about any command
  license       Manage license in the astrads cluster
```

maintenance	Manage maintenance status of a node
monitoring	Manage Monitoring Output
nodes	Manage nodes in a cluster

#### Flags:

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

to use

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

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

回應：

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

Usage:

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

Examples:

```
# Control plane collection
```

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

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

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

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

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

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

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

NOTE:

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

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

Flags:

```
-c, --component string      Specify the component to collect:  
[storage , controlplane , vasaprovider, all]
```

```
-d, --duration int          Duration is the duration in hours from  
the startTime for collection  
of AutoSupport.
```

```
-e, --event string          Specify the callhome event to trigger.  
(default "manual")
```

```
-f, --forceUpload           Configure an AutoSupport to upload if  
it is in the compressed state  
and not
```



```

the 'local' option or if
disabled

    -h, --help            help for collect
    -l, --local            Only collect and compress the
autosupport bundle. Do not upload
                           to support.
                           Use 'download' to copy the collected
bundle after it is in
                           the 'compressed' state
                           Specify nodes to collect for storage
                           --nodes string
                           component. (default "all")
                           -t, --startTime string
                           StartTime is the starting time for
                           collection of AutoSupport.
                           This should be in the ISO 8601 date
                           time format.
                           Example format accepted:
                           2021-01-01T15:20:25Z, 2021-01-
01T15:20:25-05:00
                           -u, --usermessage string
                           UserMessage is the additional message
                           to include in the
                           AutoSupport subject.
                           (default "Manual event trigger from
CLI")

```

## 部署測試應用程式

以下是部署可搭配Astra Data Store使用的測試應用程式的步驟。

在此範例中、我們使用Helm儲存庫部署Bitnami的MongoDB圖表。

您需要的是 **#8217** ；需要的是什麼

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

步驟

1. 從Bitnami新增Helm repo：

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

2. 部署MongoDB：

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

### 3. 檢查MongoDB Pod的狀態：

```
~% kubectl get pods -n ns-mongodb
```

NAME	READY	STATUS	RESTARTS	AGE
mongodb-9846ff8b7-rfr4r	1/1	Running	0	67s

### 4. 驗證MongoDB使用的持續磁碟區宣告 (PVC)：

```
~% kubectl get pvc -n ns-mongodb
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES
STORAGECLASS	AGE			
mongodb	Bound	pvc-1133453a-e2f5-48a5	8Gi	RWO
trident-csi	97s			

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

```
~% kubectl get astradsvolume pvc-1133453a-e2f5-48a5 -n astrads-system
```

NAME	SIZE	IP	CLUSTER	CREATED
pvc-1133453a-e2f5-48a5	8830116Ki	10.192.2.192	jai-ads	true

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

```
~% kubectl describe astradsvolume pvc-1133453a-e2f5-48a5 -n astrads-
system
```

Name: pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07

Namespace: astrads-system

Labels: astrads.netapp.io/cluster=jai-ads  
astrads.netapp.io/mip=10.192.1.39  
astrads.netapp.io/volumeUUID=cf33fd38-a451-596c-b656-61b8270d2b5e  
trident.netapp.io/cloud=on-prem  
trident.netapp.io/creator=trident-dev  
trident.netapp.io/performance=premium

Annotations: provisioning: {"provisioning":{"cloud":"on-prem","creator":"trident-dev","performance":"premium"}}  
trident:  
{"trident":{"version":"21.10.0-test.jenkins-trident-stable-v21.10-

```

2+e03219ce37294d9ba54ec476bbe788c1a7772548","backendUUID":"","platform":
...
API Version:  astrads.netapp.io/v1alpha1
Kind:          AstraDSVolume
Metadata:
  Creation Timestamp:  2021-12-08T19:35:26Z
  Finalizers:
    trident.netapp.io/astradsvolume-finalizer
    astrads.netapp.io/astradsvolume-finalizer
  Generation:  1
  Managed Fields:
    API Version:  astrads.netapp.io/v1alpha1
    Fields Type:  FieldsV1
    fieldsV1:
      f:metadata:
        f:labels:
          f:astrads.netapp.io/cluster:
          f:astrads.netapp.io/mip:
          f:astrads.netapp.io/volumeUUID:
        f:status:
          .:
          f:cluster:
          f:conditions:
          f:created:
          f:displayName:
          f:exportAddress:
          f:internalName:
          f:mip:
          f:permissions:
          f:qosPolicy:
          f:requestedSize:
          f:restoreCacheSize:
          f:size:
          f:snapshotReservePercent:
          f:state:
          f:volumePath:
          f:volumeUUID:
    Manager:      cluster-controller
    Operation:    Update
    Time:         2021-12-08T19:35:32Z
    API Version:  astrads.netapp.io/v1alpha1
    Fields Type:  FieldsV1
    fieldsV1:
      f:status:
        f:exportPolicy:
    Manager:      dms-controller

```

```

Operation:      Update
Subresource:    status
Time:           2021-12-08T19:35:32Z
API Version:    astrads.netapp.io/v1alpha1
Fields Type:    FieldsV1
fieldsV1:
  f:metadata:
    f:annotations:
      .:
    f:provisioning:
    f:trident:
  f:finalizers:
    v:"trident.netapp.io/astradsvolume-finalizer":
  f:labels:
    .:
    f:trident.netapp.io/cloud:
    f:trident.netapp.io/creator:
    f:trident.netapp.io/performance:
  f:spec:
    .:
    f:cluster:
    f:displayName:
    f:exportPolicy:
    f:noSnapDir:
    f:permissions:
    f:qosPolicy:
    f:size:
    f:snapshotReservePercent:
    f:type:
    f:volumePath:
Manager:        trident_orchestrator
Operation:      Update
Time:           2021-12-08T19:35:34Z
Resource Version: 12007115
UID:            d522ae4f-e793-49ed-bbe0-9112d7f9167b
Spec:
  Cluster:      jai-ads
  Display Name:  pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07
  Export Policy: pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07
  No Snap Dir:  true
  Permissions:  0777
  Qos Policy:   silver
  Size:         9042036412
  Snapshot Reserve Percent: 5
  Type:         ReadWrite
  Volume Path:  /pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07

```

```

Status:
Cluster:  jai-ads
Conditions:
  Last Transition Time:  2021-12-08T19:35:32Z
  Message:              Volume is online
  Reason:              VolumeOnline
  Status:              True
  Type:              AstraDSVolumeOnline
  Last Transition Time:  2021-12-08T19:35:32Z
  Message:              Volume creation request was successful
  Reason:              VolumeCreated
  Status:              True
  Type:              AstraDSVolumeCreated
Created:              true
Display Name:         pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07
Export Address:       10.192.2.192
Export Policy:        pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07
Internal Name:        pvc_1133453a_e2f5_48a5_a06c_d14b8aa7be07
Mip:                  10.192.1.192
Permissions:          777
Qos Policy:           silver
Requested Size:        9042036412
Restore Cache Size:    0
Size:                  8830116Ki
Snapshot Reserve Percent: 5
State:                online
Volume Path:          /pvc-1133453a-e2f5-48a5-a06c-d14b8aa7be07
Volume UUID:          cf33fd38-a451-596c-b656-61b8270d2b5e
Events:
  Type      Reason          Age   From                      Message
  ----      -
  Normal    VolumeCreated  3m9s  ADSClusterController    Volume creation
request was successful

```

## 管理Astra Data Store叢集

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

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

您需要的是 **#8217** ；需要的是什麼

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

## 新增節點

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



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

### 步驟

1. 如果新節點的dataIP尚未納入Astra Data Store叢集CR、請執行下列步驟：
  - a. 編輯叢集CR、並在「adsDataNetworks」\*「Addresses」（位址）\*欄位中新增額外的dataIP。以適合您環境的適當值取代以大寫字母顯示的資訊：

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

回應：

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

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

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

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

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

3. 確認已新增節點：

```
kubectl astrads nodes list
```

## 移除節點

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

### 步驟

1. 列出所有節點：

```
kubectl astrads nodes list
```

回應：

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

2. 標記要移除的節點。以適合您環境的適當值取代以大寫字母顯示的資訊：

```
kubectl astrads nodes remove NODE_NAME
```

回應：

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

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

3. 驗證移除節點的「Present（目前）」狀態：

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

回應：

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

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

```
kubectl astrads nodes uninstall NODE_NAME
```

5. 驗證節點是否已從叢集移除：

```
kubectl astrads nodes list
```

節點會從Astra資料儲存區移除。



## 將節點置於維護模式

當您需要執行主機維護或套件升級時、應將節點置於維護模式。



節點必須已是Astra Data Store叢集的一部分。

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

### 步驟

1. 顯示節點詳細資料：

```
kubectl get nodes
```

回應：

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

2. 確保節點尚未處於維護模式：

```
kubectl astrads maintenance list
```

回應（維護模式中沒有節點）：

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

3. 啟用維護模式。以適合您環境的適當值取代以大寫字母顯示的資訊：

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

例如：

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

回應：

```
Maintenance mode astrads-system/maint1 created
```

#### 4. 列出節點：

```
kubectl astrads nodes list
```

回應：

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

#### 5. 檢查維護模式的狀態：

```
kubectl astrads maintenance list
```

回應：

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

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

#### 6. 節點維護完成後、請停用維護模式：

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

#### 7. 確保節點不再處於維護模式：

```
kubectl astrads maintenance list
```

## 新增磁碟機至節點

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

您需要的是 **#8217** ；需要的是什麼

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



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

## 步驟

### 1. 描述叢集：

```
kubectl astrads clusters list
```

回應：

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

### 2. 記下叢集名稱。

### 3. 顯示可新增至叢集中所有節點的磁碟機。以叢集名稱取代叢集名稱：

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

回應：

```

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

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

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

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

```

#### 4. 執行下列其中一項：

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

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

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

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

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

## 更換磁碟機

當叢集中的磁碟機故障時、必須儘快更換磁碟機、以確保資料完整性。如果磁碟機故障、您可以在叢集CR節點狀態、叢集健全狀況資訊和度量端點中查看故障磁碟機的相關資訊。您可以使用下列命令範例來查看故障磁碟機資訊。

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

```
kubectl get adsc1 -A -o yaml
```

回應：

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

故障磁碟機CR會在叢集中自動建立、名稱對應於故障磁碟機的UUID。

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

回應：

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

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

回應：

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

步驟

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

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

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

回應：

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

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

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



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

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

3. 若要重新執行磁碟機更換、請使用之前的命令「-force」：

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

以取得更多資訊

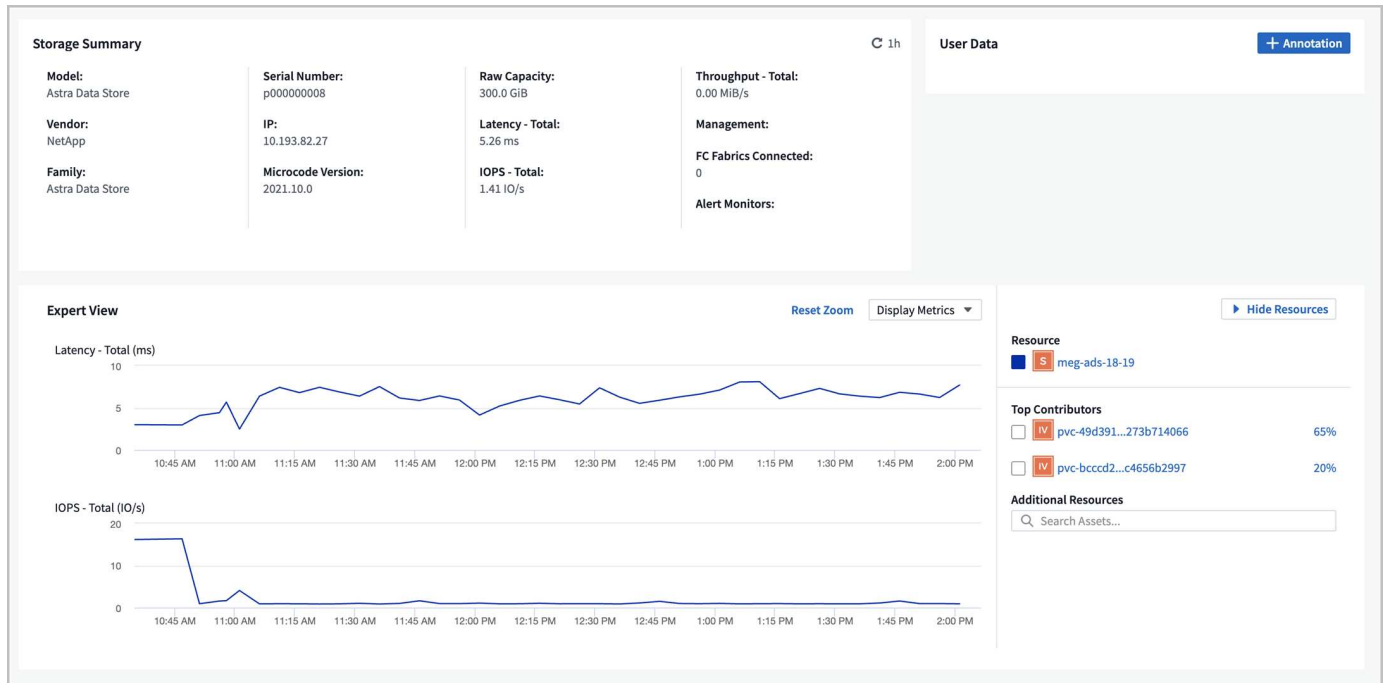
- ["使用kveccli命令管理Astra Data Store資源"](#)

## 監控Astra資料儲存區

### 利用功能表監控指標Cloud Insights

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

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



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

您可以完成下列工作：

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

完成**Cloud Insights** 連線先決條件工作

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

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

收集下列資訊：

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



## 擷取單元儲存

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

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

## 下載並執行安裝指令碼

提供Bash指令碼、可透過監控操作員啟用Astra Data Store監控功能。Cloud Insights安裝指令碼會安裝擷取單元、其中含有Astra Data Store收集器和一個Fluent位元代理程式。

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

然後、會以下列方式傳送指標：

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

## 顯示安裝程式指令碼說明

安裝程式指令碼的完整說明文字如下所示：

顯示安裝程式指令碼說明文字：

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

回應：

```

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

```

#### 執行安裝指令碼

1. 如果Cloud Insights 您還沒有這個帳戶、請建立一個這個帳戶。
2. 登入Cloud Insights 到
3. 在「支援資料」功能表中、按一下「管理」>「資料收集器」Cloud Insights。
4. 按一下「+資料收集器」以新增收集器。



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

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



Select a Data Collector



Configure Collector



**NetApp**  
Astra Data Store

## Configure Collector

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

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


Kubernetes

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

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

[+ API Access Token](#)

[Production Best Practices ?](#)

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

- 1

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

Copy Installer Script

☐ Reveal Installer Script

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

CI_DOMAIN_NAME="f49uaky.gstabler-ads.cloudinsights-dev.netapp.com"
CI_API_KEY="eyJraWQ1OiI5OTk5IiwidHlwIjo1IiwiaWYxIjo1SFZ0DQ1fQ.eyJjcVhdG9yTG9naW41OiJhZG1pb2IiImRpc3BsYXl0YW1lIjo1ZGVmYXVsdF9hZHNfYXhBpX2t1eTEgKG9uIGJlaGFsZiBvZiBhZG1pb2I"

```
- 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 kubect
- 8

Complete Setup

8. 指令碼完成後、按一下\*完成設定\*。

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



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

安裝指令碼會在組態成功時產生類似的輸出：

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

代理程式**CR**範例

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

```

spec:
  au:
    isEnabled: true
    storageClassName: auto-sc
  cluster-name: meg-ads-21-22-29-30
  docker-repo: docker.repo.eng.netapp.com/global/astra
  fluent-bit:
  - name: ads-tail
    outputs:
    - sink: ADS_STDOUT
    substitutions:
    - key: TAG
      value: firetapems
    - key: LOG_FILE
      values:
      - /var/log/firetap/*/ems/ems
      - /var/log/firetap/ems/*/ems/ems
    - key: ADS_CLUSTER_NAME
      value: meg-ads-21-22-28-29-30
  - name: agent
  - name: ads-tail-ci
    outputs:
    - sink: CI
    substitutions:
    - key: TAG
      value: netapp.ads
    - key: LOG_FILE
      values:
      - /var/log/firetap/*/ems/ems
      - /var/log/firetap/ems/*/ems/ems
    - key: ADS_CLUSTER_NAME
      value: meg-ads-21-22-28-29-30
  output-sink:
  - api-key: abcd
    domain-name: bz19ngz.gst-adsdemo.ci-dev.netapp.com
    name: CI
  serviceAccount: sa-netapp-monitoring
  status:
  au-pod-status: UP
  au-uuid: eddeccc6-3aa3-4dd2-a98c-220085fae6a9

```

編輯Cloud Insights 此鏈接

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

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

#### 更新Kubernetes API權杖

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

#### 更新Cloud Insights 程式：更新程式碼

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

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

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

此區段如下所示：

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

6. 儲存並結束編輯器視窗。

監控操作員會更新Fluent位元、以使用新Cloud Insights 的更新版的解決方案API存取權杖。

#### 中斷Cloud Insights 與該功能的連線

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

#### 移除Astra Data Store收集器

1. 登入Cloud Insights 到

2. 選取\*管理\*>\*資料收集器\*以存取「資料收集器」頁面。
3. 尋找Astra Data Store叢集的項目。
4. 選取畫面右側的功能表、然後選取\*刪除\*。
5. 按一下確認頁面上的\*刪除\*。

移除擷取單元、Telegraf（若已設定）和Fluent位元

1. 編輯Agent CR:

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

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

監控操作員會更新Telegraf（若已設定）和Fluent位元組態、並刪除擷取單元Pod。

7. 如果您將本機目錄用於擷取單元PV、而非儲存資源配置程式、請刪除PV：

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

然後、刪除正在執行擷取單元的節點上的實際目錄。

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

監控操作員會更新Telegraf（若已設定）和Fluent位元組態、並移除擷取單元。

## Open Metrics API說明

以下是可用來從Astra Data Store收集度量的API清單。

- 「說明」行說明指標。
- 「類型」行指出度量是量表還是計數器。

```

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

```



```

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

```

```

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

```

```

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

```

## 使用Prometheus和Grafana監控指標

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

您需要的是 **#8217** ；需要的是什麼

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

## 設定Prometheus

Astra Data Store在Kubernetes叢集中的TCP連接埠9341上提供度量服務。您必須設定Prometheus、才能從此服務收集指標。

## 步驟

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

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

3. 啟動Prometheus服務。

## 設定Grafana

您可以設定Grafana以顯示Prometheus收集的指標。

## 步驟

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

```
apiVersion: 1

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

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

## 匯入Grafana儀表板範本

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

## 步驟

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

## 設定及監控事件記錄

若要監控事件管理系統（EMS）記錄、您可以執行下列高層級工作：

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

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

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

輸入：

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

其中：

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

## 設定Cloud Insights 功能

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

## 串流事件記錄至Elastic

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

輸入：

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



彈性主機名稱可以是IP位址。

## 安全的Astra資料儲存區

### 管理安全性憑證

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

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

## 撤銷憑證

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



如果您撤銷CA憑證、這會撤銷該CA所簽署的任何憑證。

## 步驟

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

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

輸出應類似於下列內容：

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

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

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

現有的憑證會被撤銷、並自動產生新的憑證。

## 管理外部金鑰

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



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

管理金鑰包括下列自訂資源定義 (客戶需求日)：

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

您可以執行下列與外部金鑰管理相關的工作：

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

## 設定外部金鑰管理

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

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

### 步驟

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



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

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

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

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

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

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

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

回應：

Configure SEARS in AstraDS cluster

Usage:

```
astrads clusters sears [flags]
```

Flags:

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

Global Flags:

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

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

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

檢查軟體加密的靜止狀態

您可以在閒置時檢查軟體加密的組態。

步驟

1. 檢查適用的電池。



```

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

```

## 將外部變更為內部金鑰管理

如果您目前使用外部金鑰管理程式、可以將其變更為內部金鑰管理程式。

### 步驟

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



不會自動移除先前的金鑰提供者和密碼。

## 旋轉金鑰以確保安全性

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

### 設定自動金鑰旋轉

1. 更新CRD中的「適用」參數。

```

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

```

## 設定隨需金鑰旋轉

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

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

## 更新Astra Data Store授權

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

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

### 使用命令列更新Astra Data Store授權

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

#### 步驟

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

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

2. 確認已新增授權：

```
kubectl astrads license list
```

您應該會看到類似下列的回應：

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

## 升級Astra Data Store

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

### 使用KECBECVL升級Astra資料儲存區

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

#### 下載Astra Data Store產品組合並擷取映像

##### 步驟

1. 登入 "[NetApp 支援網站](#)" 並下載Astra Data Store套裝組合（「Astra\_Data\_Store\_2022.05.tar」）。
2. （可選）使用以下命令驗證套件的簽名：

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

3. 建立目錄：

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

4. 擷取映像：

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



影像將擷取至工作目錄中建立的「astrads/images/」目錄。

#### 複製二進位檔並將映像推送至本機登錄

##### 步驟

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



使用「which kubectll」命令尋找安裝kubectll二進位檔的路徑。

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

2. 將Astra Data Store映像目錄中的檔案新增至本機登錄。



請參閱以下自動載入影像的範例指令碼。

- a. 登入您的登錄：

```
docker login [your_registry_path]
```

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

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

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

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

## 執行升級

### 步驟

1. 將「astradsoper.yaml」檔案複製到本機目錄：

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

2. 升級營運者。以適合您環境的適當資訊取代大寫字母中的引數：

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

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

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

此時會出現一則訊息、通知您升級已開始、並需要幾分鐘時間才能完成。

## 使用自動指令碼解除安裝Astra Data Store

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

您可以使用不同的方法來解除安裝：

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

## 使用自動指令碼解除安裝Astra Data Store

此程序使用自動指令碼來解除安裝Astra Data Store。

您需要的是 **#8217** ；需要的是什麼

- root系統管理權限

Astra Data Store解除安裝程序會引導您完成下列高層級步驟：

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

移除現有的工作負載和繫結

在解除安裝Astra Data Store之前、您必須先移除下列項目

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

如此可確保Kubernetes環境保持乾淨狀態、這在重新安裝時非常重要。

### 解除安裝Astra Data Store叢集

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

1. 在「manifest」目錄中找到「uninstall.sh」。

2. 執行下列「shed（已執行）」命令：

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

3. 執行下列指令碼、指出您要解除安裝的項目：

```
./uninstall.sh
```

```
You must run this script with an argument specifying what should be  
uninstalled
```

```
To uninstall the ADS cluster run ./uninstall.sh cluster
```

```
To uninstall everything run ./uninstall all
```

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

否則、如果您想要解除安裝所有項目、請輸入「uninstall.sh」



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

5. 出現提示時、請確認您要繼續、然後輸入「eraseDDATA」

回應：

```
./uninstall.sh all
```

```
Enter 'erasedata' to confirm you want proceed with the uninstall:  
erasedata
```

```
+-----+
```

```
| Wed Feb  2 10:14:01 EST 2022 |
```

```
| ADS cluster uninstall started |
```

```
+-----+
```

```
Deleting astradsvolumes
```

```
Deleted astradsvolumes
```

```
Deleting astradsexportpolicies
```

```
Deleted astradsexportpolicies
```

```
Deleting astradsvolumesnapshots
```

```
Deleted astradsvolumesnapshots
```

```
Deleting astradsclusters
```

```
Deleted astradsclusters
```

```
Deleting astradslicenses
```

```
Deleted astradslicenses
```

```

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

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

Removing astradsversion
astradsversion.astrads.netapp.io "astradsversion" deleted
Removed astradsversion
Removing daemonsets
daemonset.apps "astrads-ds-nodeinfo-astradsversion" deleted
Removed daemonsets
Removing deployments
deployment.apps "astrads-cluster-controller" deleted
deployment.apps "astrads-license-controller" deleted
deployment.apps "astrads-operator" deleted
Removed deployments
Removing all other AstraDS resources
namespace "astrads-system" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsautosupports.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradscLOUDsnapshots.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsclusters.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsexportpolicies.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsfaileddrives.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradslICENSES.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsnfsOPTIONS.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsnodeinfoes.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsnodemANAGEMENTS.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsgOSPolicies.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsversions.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsvOLUMEFILES.astrads.netapp.io" deleted

```

```
customresourcedefinition.apiextensions.k8s.io
"astradsvolumes.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsvolumesnapshots.astrads.netapp.io" deleted
role.rbac.authorization.k8s.io "astrads-astrads-system-admin-role"
deleted
role.rbac.authorization.k8s.io "astrads-astrads-system-reader-role"
deleted
role.rbac.authorization.k8s.io "astrads-astrads-system-writer-role"
deleted
role.rbac.authorization.k8s.io "astrads-leader-election-role" deleted
role.rbac.authorization.k8s.io "astrads-manager-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astrads-admin-
clusterrole" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astrads-reader-
clusterrole" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astrads-writer-
clusterrole" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsautosupport-
editor-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsautosupport-
viewer-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscloudsnapshot-
editor-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscloudsnapshot-
viewer-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscluster-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscluster-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsexportpolicy-
editor-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsexportpolicy-
viewer-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsfaileddrive-
editor-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsfaileddrive-
viewer-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradslicense-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradslicense-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsnfsoption-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsnfsoption-viewer-
role" deleted
```



```
clusterrole.rbac.authorization.k8s.io "astrads-astradsnodeinfo-editor-  
role" deleted  
clusterrole.rbac.authorization.k8s.io "astrads-astradsnodeinfo-viewer-  
role" deleted  
clusterrole.rbac.authorization.k8s.io "astrads-astradsnodemanagement-  
editor-role" deleted  
clusterrole.rbac.authorization.k8s.io "astrads-astradsnodemanagement-  
viewer-role" deleted  
clusterrole.rbac.authorization.k8s.io "astrads-astradsqospolicy-viewer-  
role" deleted  
clusterrole.rbac.authorization.k8s.io "astrads-astradsversion-editor-  
role" deleted  
clusterrole.rbac.authorization.k8s.io "astrads-astradsversion-viewer-  
role" deleted  
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolume-editor-  
role" deleted  
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolume-viewer-  
role" deleted  
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolumefile-editor-  
role" deleted  
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolumefile-viewer-  
role" deleted  
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolumesnapshot-  
editor-role" deleted  
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolumesnapshot-  
viewer-role" deleted  
clusterrole.rbac.authorization.k8s.io "astrads-manager-role" deleted  
rolebinding.rbac.authorization.k8s.io "astrads-astrads-admin-  
rolebinding" deleted  
rolebinding.rbac.authorization.k8s.io "astrads-astrads-reader-  
rolebinding" deleted  
rolebinding.rbac.authorization.k8s.io "astrads-astrads-writer-  
rolebinding" deleted  
rolebinding.rbac.authorization.k8s.io "astrads-leader-election-  
rolebinding" deleted  
rolebinding.rbac.authorization.k8s.io "astrads-manager-rolebinding"  
deleted  
clusterrolebinding.rbac.authorization.k8s.io "astrads-astrads-admin-  
rolebinding" deleted  
clusterrolebinding.rbac.authorization.k8s.io "astrads-astrads-reader-  
rolebinding" deleted  
clusterrolebinding.rbac.authorization.k8s.io "astrads-astrads-writer-  
rolebinding" deleted  
clusterrolebinding.rbac.authorization.k8s.io "astrads-manager-  
rolebinding" deleted  
configmap "astrads-autosupport-cm" deleted
```

```
configmap "astrads-firetap-cm" deleted
configmap "astrads-kevents-asup" deleted
configmap "astrads-metrics-cm" deleted
secret "astrads-autosupport-certs" deleted
+-----+
| Wed Feb  2 10:16:36 EST 2022 |
| ADS system uninstall done    |
+-----+
```

### 驗證刪除astrad-system命名空間

請確定下列命令沒有傳回任何結果：

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

### 確保容器未在工作節點上執行

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

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

### 刪除OpenShift Container Platform資源

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

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

完成標準的解除安裝程序之後、請完成下列步驟。

#### 1. 移除SCC資源：

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

#### 2. 移除角色繫結資源：

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



請忽略這些步驟中的「找不到資源」錯誤。

## 無需指令碼即可手動解除安裝Astra Data Store

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

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

您需要的是 **#8217** ；需要的是什麼

- root系統管理權限

Astra Data Store解除安裝程序會引導您完成下列高層級步驟：

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

移除現有的工作負載和繫結

在解除安裝Astra Data Store之前、您必須先移除下列項目

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

如此可確保Kubernetes環境保持乾淨狀態、這在重新安裝時非常重要。

解除安裝Astra Data Store叢集和控制面板

請依照下列步驟手動解除安裝Astra Data Store。

刪除磁碟區並匯出原則

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



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

步驟

1. 刪除磁碟區：

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

## 2. 刪除匯出原則：

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

### 刪除Astra Data Store叢集

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



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

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

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

## 1. 刪除叢集：

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

### 刪除授權

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

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

### 刪除Astra Data Store安裝

刪除叢集中的控制器、運算子、命名空間和支援Pod。

## 1. 刪除Astra Data Store安裝物件：

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

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

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

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

### 3. 刪除剩餘成品和運算子yaml檔案：

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

### 驗證刪除**astrad-system**命名空間

請確定下列命令沒有傳回任何結果：

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

### 確保容器未在工作節點上執行

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

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

### 刪除**OpenShift Container Platform**資源

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

OpenShift使用安全內容限制（SCC）來控制Pod可以執行的動作。

完成標準的解除安裝程序之後、請完成下列步驟。

#### 1. 移除SCC資源：

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

#### 2. 移除角色繫結資源：

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



請忽略這些步驟中的「找不到資源錯誤」。

## 手動刪除範例

以下是執行手動解除安裝指令碼的範例。

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

$ kubectl delete astradslicenses --all -A
astradslicense.astrads.netapp.io "e900000005" deleted

$ kubectl delete astradsdeployment astradsdeployment -n astrads-system
astradsdeployment.astrads.netapp.io "astradsdeployment" deleted

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

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

$ kubectl delete -f /.../firetap/sds/manifests/netappsdsoperator.yaml
namespace "astrads-system" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsautosupports.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradscloudsnapshots.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsclusters.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsdeployments.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsexportpolicies.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsfaileddrives.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
```

```
"astradslicenses.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsnfsoptions.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsnodeinfoes.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsqospolicies.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsvolumefiles.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsvolumes.astrads.netapp.io" deleted
customresourcedefinition.apiextensions.k8s.io
"astradsvolumesnapshots.astrads.netapp.io" deleted
role.rbac.authorization.k8s.io "astrads-leader-election-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscloudsnapshot-
editor-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscloudsnapshot-
viewer-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscluster-editor-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradscluster-viewer-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradslicense-editor-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradslicense-viewer-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolume-editor-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-astradsvolume-viewer-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-autosupport-editor-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-autosupport-viewer-role"
deleted
clusterrole.rbac.authorization.k8s.io "astrads-manager-role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-metrics-reader" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappexportpolicy-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappexportpolicy-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappsdsdeployment-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappsdsdeployment-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappsdsnfsoption-editor-
role" deleted
```

```

clusterrole.rbac.authorization.k8s.io "astrads-netappsdsnfsoption-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappsdsnodeinfo-editor-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-netappsdsnodeinfo-viewer-
role" deleted
clusterrole.rbac.authorization.k8s.io "astrads-proxy-role" deleted
rolebinding.rbac.authorization.k8s.io "astrads-leader-election-
rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "astrads-manager-rolebinding"
deleted
clusterrolebinding.rbac.authorization.k8s.io "astrads-proxy-rolebinding"
deleted
configmap "astrads-autosupport-cm" deleted
configmap "astrads-firetap-cm" deleted
configmap "astrads-fluent-bit-cm" deleted
configmap "astrads-kevents-asup" deleted
configmap "astrads-metrics-cm" deleted
service "astrads-operator-metrics-service" deleted
Error from server (NotFound): error when deleting
"/.../export/firetap/sds/manifests/netappsdsoperator.yaml":
deployments.apps "astrads-operator" not found

$ kubectl get ns | grep astrads-system

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

```

## Astra Data Store解除安裝程序疑難排解

如果您需要疑難排解解除安裝程序、請檢閱下列建議。

### Pod處於終止狀態

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

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

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

### 服務品質原則指向舊叢集

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

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



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

## 2. 刪除整個Astra Data Store部署（不只是叢集）：

```
uninstall.sh all
```

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

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

## 範例 2. 詳細資料

請使用下列因應措施指示：

### 步驟

#### 1. 確認未移除金鑰提供者CRS：

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

回應：

NAME	AGE
externalkeyprovider1	94s

#### 2. 移除金鑰提供者CRS：

##### a. 移除最終化工具：

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

##### b. 移除下方反白顯示的最終化工具行：

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

```

apiVersion: astrads.netapp.io/v1beta1
kind: AstraDSKeyProvider
metadata:
  creationTimestamp: "2022-05-24T16:38:27Z"
  finalizers:
  - astrads.netapp.io/astradskeyprovider-finalizer
  generation: 1
  labels:
    astrads.netapp.io/cluster: astrads-cluster-example
    astrads.netapp.io/rsid: "1"
  name: externalkeyprovider1
  namespace: astrads-system
  resourceVersion: "1134699"
  uid: a11111b2-31c0-4575-b7f3-97f9ab1a1bla
spec:
  cluster: astrads-cluster-example
  kmipServer:
    hostnames:
      - 10.xxx.xxx.xxx
    port: 5696
    secretRef: externalkeyprovider1
status:
  keyProviderUUID: a1b2cd34-4fc6-5bae-9184-2288c673181d
  kmipServerStatus:
    capabilities: '{ KMIP_library_version()=17367809,
KMIP_library_version_str()="KMIP
  1.9.3a 8-Apr-2019", KMIP_library_version_tag()="KMIP part
of KMIP 1.9.3a 8-Apr-2019",
    KMIP_library_is_eval()=false,
KMIP_library_fips_capable()=true(FIPS140),
KMIP_SSL_provider_build_version()=268444095,
    KMIP_SSL_provider_version()=268444095,
KMIP_SSL_provider_version_str()="OpenSSL
  1.0.2zb-fips 23 Sep 2021" }'
  keyServerUUID: 8422bdd0-74ad-579d-81bd-6d544ac4224a

```

c. 移除釋放器之後、請刪除金鑰提供者CR：

```

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

```

# 使用Astra Data Store搭配VMware

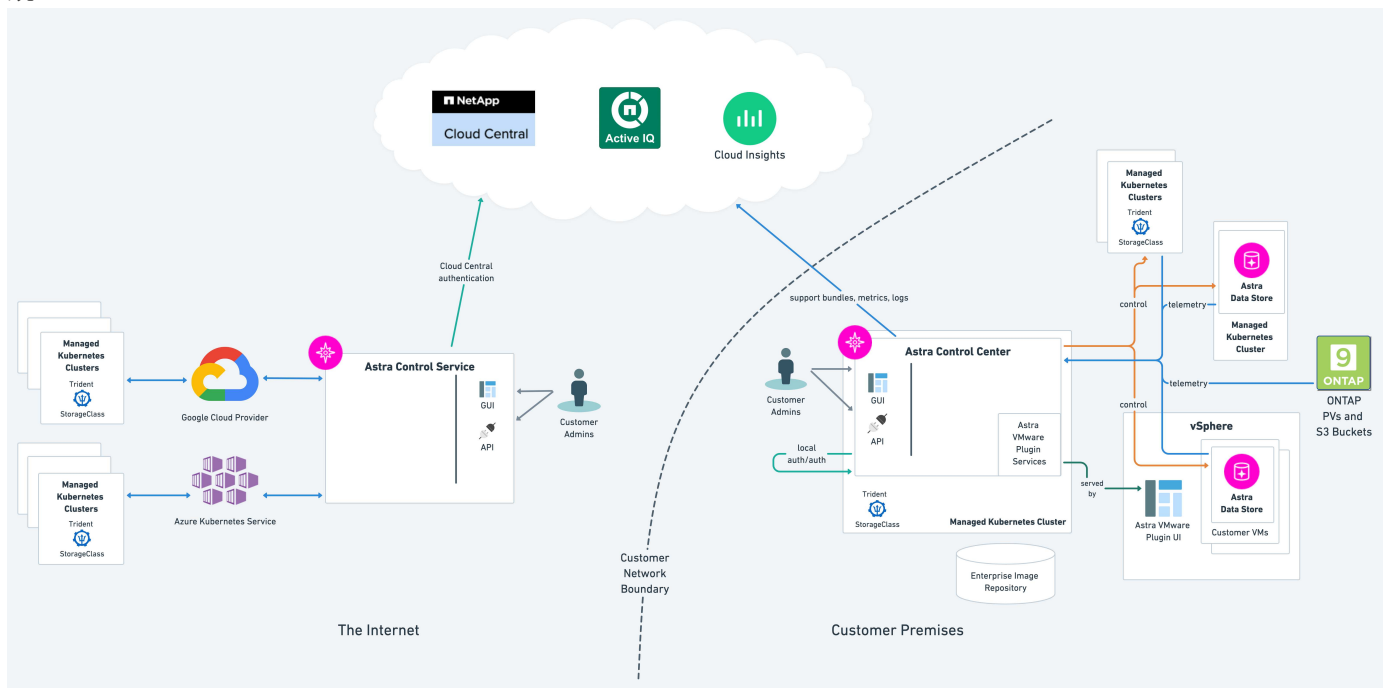
## 瞭解Astra Data Store與VMware的相關資訊

Astra Data Store同時支援容器化和虛擬化的工作負載。整合vVols與儲存原則型管理、讓vSphere管理員能夠套用儲存品質的服務。適用於VMware vSphere的NetApp Astra外掛程式可提供您熟悉的管理與監控體驗、免除繁瑣的儲存工作。

適用於VMware vSphere的Astra外掛程式具有下列優點：

- VM精細的儲存資源配置、具備完整vVols與VASA整合功能
- 儲存原則型管理整合
- vCenter外掛程式適用於vSphere原生管理

下圖顯示Astra系列與VMware的生態系統。



以取得更多資訊

- ["Astra Control Center文件"](#)
- ["Astra系列簡介"](#)

## Astra Data Store搭配VMware需求

驗證環境是否符合一般需求之後 ["Astra資料儲存區需求"](#)、您應該確認您的環境也符合VMware元件的最低需求、例如VMware VASA供應商和NetApp Astra Plugin for VMware vSphere。

## VMware vSphere需求

Astra Data Store使用VMware VASA Provider作為API介面、與儲存設備進行通訊。確保您的環境符合基本需求"[VASA供應商要求](#)" 以及下列額外需求：

- VMware vSphere 7.0（支援Update 1到Update 3）
- 一個未指派的入口流量IP位址



- 適用於VMware vSphere的NetApp Astra外掛程式不支援vSphere 7.0 Update 3c；請改用vSphere 7.0 Update 3D。
- 適用於VMware vSphere的NetApp Astra外掛程式不支援連結模式vCenter伺服器。

## 適用於VMware vSphere的NetApp Astra外掛程式需求

適用於VMware vSphere的NetApp Astra外掛程式具有下列需求：

- 在Kubernetes叢集上執行Astra Control Center執行個體
- 在Kubernetes叢集上執行的授權Astra Data Store執行個體

支援的網頁瀏覽器

適用於VMware vSphere的NetApp Astra外掛程式支援下列最新版本的網頁瀏覽器：

- Mozilla Firefox
- Microsoft Edge（基於鉻）
- Google Chrome

以取得更多資訊

- "[Astra Control Center文件](#)"
- "[Astra系列簡介](#)"
- "[Astra資料儲存區需求](#)"

## 使用VMware設定Astra Data Store

您可以將Astra Data Store設定為儲存後端、並使用適用於VMware vSphere的NetApp Astra外掛程式加以管理。

使用VMware設定Astra Data Store需要執行下列工作：

- [\[Activate VMware vSphere services using Astra Control Center\]](#)。
- [\[Add a vCenter using Astra Control Center\]](#)。
- [\[Create a custom SCC \(if using OpenShift\)\]](#)
- [\[Use an existing storage backend in the Astra Plugin for VMware vSphere\]](#)。
- [\[Create a datastore using the Astra Plugin for VMware vSphere\]](#)。

- [\[Generate VM storage policies\]](#)。

在您使用VMware設定Astra Data Store之前、您必須確保下列事項：

- Astra控制中心是 **"已安裝"** 並設定。



對於Astra Data Store Early Access Program (EAP) 版本、如果您打算使用Astra Control Center來管理Astra Data Store並啟用VMware工作流程、請僅在「pCloud」命名空間上部署Astra Control Center。

- Astra Data Store已獲得授權並已部署。請參閱 ["安裝Astra Data Store"](#)。



對於Astra Data Store Early Access Program (EAP) 版本、Astra Control Center和Astra Data Store必須部署於兩個不同的Kubernetes叢集。

- 用於部署Astra Control Center和Astra Data Store的Kubernetes叢集必須已由Astra Control Center管理。
- 您在新增vCenter之前、已上傳Astra Control Center和VASA供應商套件。 ["請參閱管理軟體套件"](#)。

## 使用Astra Control Center啟動VMware vSphere服務

啟動Astra Control Center上的vSphere服務、開始使用VMware設定Astra Data Store。



Astra Control Center中的VMware vSphere服務預設不會啟用。

1. 登入Astra Control Center。
2. 從左側導覽中選取\*叢集\*。

橫幅會顯示尚未啟用VMware vSphere服務的訊息。

3. 選取\*啟用VMware vSphere服務\*。

此動作可能需要一段時間。服務啟用後、\*新增vCenter\*按鈕即會啟用。

## 使用Astra Control Center新增vCenter

新增第一個vCenter、註冊適用於VMware vSphere的Astra外掛程式。

您必須具有管理權限、才能將vCenter新增至Astra Control Center。



在VMware vSphere註冊外掛程式之後、「VMware捷徑」頁面會出現Astra Plugin for VMware vSphere圖示。

1. 登入Astra Control Center。
2. 從左側導覽中選取\*叢集\*。
3. 選取\*新增vCenter\*。
4. 輸入vCenter伺服器詳細資料、vCenter連接埠、以及管理使用者名稱和密碼、以提供給Astra Control Center。



如此可在VMware vSphere用戶端中部署此vCenter的Astra外掛程式。

#### 5. 選取\*「Add\*」。

vCenter會顯示在「叢集」頁面上、而Astra控制儀表板上的託管vCenter總數會更新。這也會啟動適用於VMware vSphere部署的Astra外掛程式。

### 驗證vCenter新增項目

新增的vCenter會顯示在「叢集」頁面和儀表板上。



vCenter和Kubernetes叢集都會顯示在Astra控制中心儀表板上。

1. 存取Astra控制中心。
2. 從左側導覽中選取\*叢集\*。
3. 確認「叢集」頁面上出現新管理的vCenter。
4. 從左側導覽中選取\*儀表板\*。
5. 從Astra控制中心儀表板中、記下新的託管vCenter叢集、做為\*託管\*數的一部分。



受管理的叢集數包括vCenter和Kubernetes叢集。

6. 若要檢視叢集詳細資料、請按一下\*「託管」\*計數。

此時會出現「叢集」頁面。

### 建立自訂SCC（如果使用OpenShift）

如果您使用OpenShift、您可以選擇指派安全內容限制（SCC）來控制Pod可以執行的動作、並控制Pod可以存取的內容。

根據預設、任何容器的執行都會被授予受限的SCC、而且只會授予該SCC所定義的功能。受限的SCC不會提供VASA供應商Pod所需的權限。使用此程序可為VASA供應商部署所使用的服務帳戶提供所需的較高權限（列於範例中）。

將自訂SCC指派給Astra Data Store「NTV-system」命名空間的各種預設服務帳戶、此命名空間是特殊權限與節點輸出程式SCC的混合體。

只有在Red Hat OpenShift Container Platform（OCP）上部署時、才需要執行下列步驟。

1. 建立名為「VP\_backend\_特權\_scc.yaml」的自訂SCC：

```
kubectl create -f vp_backend_privileged_scc.yaml
```

範例：VP\_backend\_特權\_SCC.yaml

```

allowHostDirVolumePlugin: true
allowHostIPC: false
allowHostNetwork: true
allowHostPID: false
allowHostPorts: true
allowPrivilegeEscalation: true
allowPrivilegedContainer: true
allowedCapabilities:
  - '*'
allowedUnsafeSysctls:
  - '*'
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  type: RunAsAny
groups: []
kind: SecurityContextConstraints
metadata:
  name: vpbackend-privileged
priority: null
readOnlyRootFilesystem: false
requiredDropCapabilities: null
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: RunAsAny
seccompProfiles:
  - '*'
supplementalGroups:
  type: RunAsAny
users:
  - system:serviceaccount:ntv-system:default
  - system:serviceaccount:ntv-system:ntv-auth-svc
  - system:serviceaccount:ntv-system:ntv-autosupport
  - system:serviceaccount:ntv-system:ntv-compliance-svc
  - system:serviceaccount:ntv-system:ntv-datastore-svc
  - system:serviceaccount:ntv-system:ntv-metallb-controller
  - system:serviceaccount:ntv-system:ntv-metallb-speaker
  - system:serviceaccount:ntv-system:ntv-mongodb
  - system:serviceaccount:ntv-system:ntv-nfs-svc
  - system:serviceaccount:ntv-system:ntv-rabbitmq-svc
  - system:serviceaccount:ntv-system:ntv-storage-svc
  - system:serviceaccount:ntv-system:ntv-vault
  - system:serviceaccount:ntv-system:ntv-vault-admin
  - system:serviceaccount:ntv-system:ntv-vault-agent-injector
  - system:serviceaccount:ntv-system:ntv-vault-controller

```



```

- system:serviceaccount:ntv-system:ntv-vault-initializer
- system:serviceaccount:ntv-system:ntv-vcenter-svc
- system:serviceaccount:ntv-system:ntv-vm-management-svc
- system:serviceaccount:ntv-system:ntv-watcher-svc
- system:serviceaccount:ntv-system:ntv-vault-sa-vault-tls
- system:serviceaccount:ntv-system:ntv-gateway-svc
- system:serviceaccount:ntv-system:ntv-jobmanager-svc
- system:serviceaccount:ntv-system:ntv-vasa-svc
volumes:
- '*'

```

## 2. 使用「oc Get SCC」命令顯示新增的SCC：

```
oc get scc vpbackend-privileged
```

回應：

NAME	PRIV	CAPS	SELINUX	RUNASUSER	FSGROUP	SUPGROUP
vpbackend-privileged	true	["*"]	RunAsAny	RunAsAny	RunAsAny	RunAsAny
<no value>	false	["*"]				

## 在適用於VMware vSphere的Astra外掛程式中使用現有的儲存後端

使用Astra Control Center UI新增vCenter之後、請使用適用於VMware vSphere的Astra外掛程式新增Astra Data Store儲存後端。

此程序完成下列動作：

- 將現有的儲存後端新增至選取的vCenter。
- 向選取的vCenter登錄VASA供應商。VASA供應商提供VMware與Astra Data Store之間的通訊。
- 將VASA供應商自我簽署的憑證新增至儲存後端。



您新增的vCenter有時可能需要10分鐘才會出現在儲存後端精靈中。



Astra資料儲存區不應與多個vCenter共用。

### 步驟

1. 存取適用於VMware vSphere的NetApp Astra外掛程式。
2. 從左側導覽中選取\* Astra Plugin for VMware vSpher\*、或從「捷徑」頁面選取\* Astra Plugin for VMware vSpher\*圖示。
3. 從「Astra Plugin for VMware vSphere Overview」（Astra外掛程式for VMware vSphere概述）頁面、選取\*

Use existing storage backend\*（使用現有的儲存後端或者、從左側導覽中選取\*儲存後端\*>\*新增\*、然後選取\*使用現有的儲存後端\*。

4. 選取現有的Astra Data Store作為儲存後端、然後選取\* Next\*。
5. 在VASA供應商頁面上、輸入VASA供應商名稱、IP位址（如果使用負載平衡器）、使用者名稱和密碼。



對於使用者名稱、您可以使用英數字元和底線。請勿輸入任何特殊字元。使用者名稱的第一個字母必須以字母字元開頭。

6. 指出您是否要部署負載平衡器、然後輸入用於存取VASA提供者的IP位址。IP必須是與節點IP分開的額外可路由的可用IP。啟用負載平衡器時、MetalLB會部署在Astra Data Store Kubernetes叢集中、並設定為分配可用的IP。



如果您使用Google Anthos叢集進行部署、請選擇不部署負載平衡器、因為Anthos已經以負載平衡器的形式執行了冶金。VASA供應商的CR（v1beta1\_vasaprovider.yaml）應將「mrarfin deploy」（mretb部署）旗標設為「假」。

如果您選擇不部署負載平衡器、則會假設負載平衡器已經部署並設定為為\*負載平衡器\*類型的Kubernetes服務分配IP。



此時部署中尚未部署VASA供應商。

7. 選擇\*下一步\*。
8. 在「憑證」頁面上、檢閱自我簽署憑證的憑證資訊。
9. 選擇\*下一步\*。
10. 檢閱摘要資訊。
11. 選取\*「Add\*」。

這會部署VASA供應商。

## 驗證Astra Plugin for VMware vSphere中的儲存後端

註冊Astra Data Store儲存後端後端之後、即會出現在Astra Plugin for VMware vSphere儲存後端清單中。

您可以決定儲存後端狀態和VASA供應商狀態。您也可以查看每個儲存後端的已用容量。

選取儲存後端之後、您也可以檢視已使用和可用的容量、資料減量比率、以及內部網路管理IP位址。

### 步驟

1. 在適用於VMware vSphere的NetApp Astra外掛程式中、從左側導覽中選取\*儲存後端\*。
2. 選取Astra Data Store儲存後端以查看「摘要」索引標籤。
3. 檢閱VASA供應商的已用及可用容量、資料減量比率及狀態。
4. 選取其他索引標籤以查看VM、資料存放區、主機和儲存節點的相關資訊。

## 使用適用於VMware vSphere的Astra外掛程式建立資料存放區

新增儲存後端並註冊適用於VMware vSphere的Astra外掛程式之後、即可在VMware中建立資料存放區。

您可以將資料存放區新增至資料中心、運算或主機叢集。



您無法使用相同的儲存後端、在同一個資料中心下建立多個資料存放區。

您可以使用NFS傳輸協定新增vVol資料存放區類型。

### 步驟

1. 存取適用於VMware vSphere的Astra外掛程式。
2. 從外掛程式功能表中、選取\* Create Datastor\*。
3. 輸入新的資料存放區名稱、類型（vVol）和傳輸協定（NFS）。
4. 選擇\*下一步\*。
5. 從「Storage（儲存設備）」頁面、選取您剛建立的Astra Data Store儲存後端。



您無法使用具有現有資料存放區的儲存後端。

6. 選擇\*下一步\*。
7. 從「摘要」頁面檢閱資訊。
8. 選擇\* Create（建立）。



如果您遇到與失敗掃描或一般系統錯誤有關的錯誤、["重新掃描/同步vCenter上的儲存供應商"](#)然後再次嘗試建立資料存放區。

## 產生VM儲存原則

建立資料存放區之後、在建立VM之前、您應該使用REST API UI中的「/virtualization /API/v1/vCenter/VM儲存原則」來產生預先設計的VM儲存原則。

### 步驟

1. 請前往「https://<ads\_gateway\_ip>:8443」存取REST API UI頁面。
2. 前往API「POST /virtualization /api/auth /登入」、並提供使用者名稱、密碼和vCenter主機名稱。

回應：

```
{
  "vmware-api-session-id": "212f4d6447b05586ab1509a76c6e7da56d29cc5b",
  "vcenter-guid": "8e475060-b3c8-4267-bf0f-9d472d592d39"
}
```

3. 前往API「Get /virtualization /api/auth/validate-session」、完成下列步驟：

- a. 使用上面產生的「VMware-API- session-id」和「vCenter-guid」作為標頭。
- b. 選擇\*立即試用\*。

回應：（驗證被截短於下方）：

```
authorization: eyJhbGciOiJSUzI1NiIsInR...9h15DYYvClT3oA connection:
keep-alive content-type: application/json date: Wed, 18 May 2022
13:31:18 GMT server: nginx transfer-encoding: chunked
```

4. 前往API「/virtualization /API/v1/vCenter/VM儲存原則」、並將先前回應中產生的承載權杖新增為「授權」。

系統會顯示「200」回應、並產生三個VM儲存原則。

5. 確認vCenter儲存原則頁面上的新VM儲存原則（名為Bronze、Silver和Gold）。
6. 繼續建立VM。

## 下一步

接下來、您可能會想要執行下列工作：

- 建立VM。
- 掛載資料存放區。請參閱 ["掛載資料存放區"](#)。

## 以取得更多資訊

- ["Astra Control Center文件"](#)
- ["Astra系列簡介"](#)

## 監控VMware安裝的元件

您可以使用適用於VMware vSphere的NetApp Astra外掛程式來監控Astra Data Store安裝的元件。您可以監控系統的健全狀況、包括儲存後端、VASA供應商、VM和vVols。您也可以檢視容量和vCenter資訊。

### 使用適用於VMware vSphere儀表板的Astra外掛程式來監控系統健全狀況

使用VMware環境管理Astra Data Store、需要監控儲存後端與VASA供應商的整體健全狀況。

使用適用於VMware vSphere儀表板的NetApp Astra外掛程式、您可以查看下列資訊：

- 此vCenter中所有儲存後端的實體已用及可用容量。您可以將游標暫留在資訊上、以查看更多詳細資料。
- 健全且不良的儲存後端與VASA供應商
- 前10大VM的延遲、IOPS、處理量和容量使用率。

您可以從儀表板執行多項額外工作：

- 監控容量
- 使用現有的儲存後端。請參閱 ["設定儲存後端"](#)。
- 存取產品文件

#### 檢閱儀表板的步驟

1. 存取適用於VMware vSphere的Astra外掛程式。
2. 請從「總覽」頁面檢閱下列各節：
  - a. \*儲存後端\*區段：您可以按一下儲存後端與VASA供應商的狀態、以查看其狀態的詳細資料。您也可以按一下以檢視所有儲存後端。
  - b. \*儲存後端容量\*區段：檢閱所選vCenter中儲存後端的實體使用量和可用容量總計。若要變更vCenter伺服器、請按一下右上角的vCenter Server選項。
  - c. \*虛擬機器\*區段：檢視使用前10大容量的虛擬機器。



您可以按一下表格標題、顯示所需的資料、例如前10大延遲虛擬機器。

#### 在其他檢視中監控Astra Data Store的步驟

1. 存取下列檢視以監控Astra Data Store元件：
  - \*虛擬機器\*索引標籤：列出所有由Astra Data Store管理的虛擬機器、以及僅列出前10大虛擬機器的儀表板。
  - \* Storage \*向下切入：顯示與儲存系統相關聯的主機、虛擬機器和資料存放區。
  - \* VM Storage\*檢視：提供VASA供應商所建立vVols的詳細資料。

### 檢閱儲存後端臨界值設定

儲存後端容量臨界值設定會在儲存後端的所有資料存放區上顯示警示通知時加以控制。

當您使用適用於VMware vSphere的Astra外掛程式來部署或新增儲存後端時、會設定下列預設臨界值：

- 90%已滿會產生紅色警示
- 80%已滿會產生黃色警示

您可以檢視系統在VMware中產生警示的層級。



對於Astra Data Store Early Access方案、如果在多個資料中心使用相同的儲存容器、您可能會看到資料存放區的警示不正確。

#### 步驟

1. 存取適用於VMware vSphere的NetApp Astra外掛程式。
2. 從左側導覽中選取\*設定\*。
3. 檢閱設定臨界值。

以取得更多資訊

- ["Astra Control Center文件"](#)
- ["Astra系列簡介"](#)

## 管理VMware安裝的Astra Data Store元件

您可以在vSphere環境和Astra Control Center中管理下列Astra Data Store元件：

- [\[Work with managed vCenters\]](#)
- [\[Manage VMs from vSphere\]](#)
- [\[Manage the storage backend\]](#)
- [\[Manage datastores\]](#)

## 與託管vCenter合作

您可以透過下列方式使用託管vCenter：

- [\[View vCenter details in Astra Control Center\]](#)
- [\[View vCenter details in Astra Control Center\]](#)
- [\[Unmanage a vCenter in Astra Control Center\]](#)

### 在Astra Control Center中檢視vCenter詳細資料

您可以決定與叢集相關的所有vCenter。

#### 步驟

1. 從Astra Control Center左側導覽中、選取\* Clusters\*。
2. 檢視vCenter清單。
3. 選擇\*保存\*。

### 在Astra Control Center中檢視vCenter詳細資料

您可能想要檢視系統和叢集的健全狀況。您可以查看Astra控制中心儀表板、判斷管理的叢集數量。

#### 步驟

1. 從Astra Control Center左側導覽中、選取\* Clusters\*。
2. 選取vCenter。
3. 檢視資訊。

### 在Astra Control Center中取消管理vCenter

如果您不再想讓Astra Control Center管理vCenter、可以取消管理。這會從Astra控制中心移除和取消登錄vCenter。



必須先從適用於VMware vSphere的Astra外掛程式中移除與此vCenter相關聯的叢集、儲存後端和VM。

#### 步驟

1. 從Astra Control Center左側導覽中、選取\* Clusters\*。
2. 從「叢集」頁面選取vCenter。



或者、選取多個vCenter、然後選取\*「全部取消管理」\*。

3. 右鍵單擊\* Actions（操作）菜單並選擇 UnManage（取消管理）\*。
4. 在「UnManage vCenter（取消管理vCenter）」頁面中、輸入「unManage（取消管理）」。
5. 選取\*是、取消管理vCenter\*。

## 從vSphere管理VM

您可以使用原生vSphere作業來管理與Astra Data Store相關的VM。

- "刪除VM"
- "重新命名VM"
- "調整VM大小"



在此版本中、您一次只能調整一個VM磁碟的大小。嘗試調整多個磁碟的大小將會失敗。

- "開啟或關閉VM"
- "暫停VM"
- "重設VM"

下列快照工作流程可用於使用原生vCenter作業的Astra Data Store：

- "快照Astra Data Store"
- "還原快照"
- "刪除快照"



Snapshot作業偶爾可能會失敗、並顯示vVol執行時間錯誤。如果發生這種情況、請重試此作業。

## 管理儲存後端

您可以移除儲存後端。移除儲存後端並不會破壞它、也不會刪除Astra Data Store產品本身；它只會從VMware取消註冊VASA供應商、並取消連結該vCenter的儲存後端。



如果VASA供應商已啟用並部署在vCenter外部、您只能移除Astra Data Store。如果將儲存後端當成資源配置資料存放區程序的一部分使用、您將無法移除儲存後端。

如果Astra Data Store未與多個vCenter連結、則移除後、VASA供應商將會取消註冊並解除安裝。

## 步驟

1. 存取適用於VMware vSphere的Astra外掛程式。
2. 從左側導覽中選取\*儲存後端\*。
3. 在Storage後端頁面上、按一下儲存後端動作功能表、然後選取\*移除\*。
4. 輸入VASA供應商的使用者名稱和密碼。
5. 選擇\*移除\*。

## 管理資料存放區

您可以使用原生vCenter作業來管理vSphere環境中的Astra Data Store、並使用Astra外掛程式擴充功能來管理資料存放區：

- ["建立資料存放區"](#)
- [\[Mount a datastore\]](#)
- [\[Delete a datastore\]](#)

## 掛載資料存放區

使用適用於VMware vSphere的Astra外掛程式、您可以將資料存放區掛載到一或多個其他主機上。

## 步驟

1. 從vCenter的資料中心庫存中選取Astra Data Store的資料存放區。
2. 在資料存放區上按一下滑鼠右鍵、然後選取\* Astra Plugin for VMware vSphere\*>\* Mount Datastore\*。
3. 從「Mount datastore on hosts」（在主機上掛載資料存放區）頁面中、選取要掛載資料存放區的主機。



如果您要在所有主機上掛載資料存放區、請勾選\*所有主機上的掛載\*。

4. 選取\*掛載\*。

啟動作業之後、您可以在vSphere Client的「近期工作」面板中追蹤進度。



如果您遇到與失敗掃描或一般系統錯誤有關的錯誤、["重新掃描/同步vCenter上的儲存供應商"](#) 然後再次嘗試建立資料存放區。

## 刪除資料存放區

使用適用於VMware vSphere的Astra外掛程式、您可以刪除資料存放區。



若要刪除資料存放區、必須先移除資料存放區上的所有VM。

## 步驟

1. 從vCenter的資料中心庫存中選取資料存放區。
2. 在資料存放區上按一下滑鼠右鍵、然後選取\* Astra Plugin\*>\* Delete Datastore\*。
3. 在「刪除資料存放區」頁面中、確認資訊、或採取其他建議動作、以便刪除資料存放區。



4. 選擇\*刪除\*。

以取得更多資訊

- ["Astra Control Center文件"](#)
- ["Astra系列簡介"](#)

## 從VMware整合式環境中解除安裝Astra Data Store

您可以從vSphere環境中解除安裝Astra Data Store及其相關元件。

請參閱 ["這些指示"](#) 解除安裝Astra Data Store。

以取得更多資訊

- ["Astra Control Center文件"](#)
- ["Astra系列簡介"](#)

# 知識與支援

## 疑難排解

瞭解如何解決您可能遇到的一些常見問題。

[https://kb.netapp.com/Advice\\_and\\_Troubleshooting/Cloud\\_Services/Astra](https://kb.netapp.com/Advice_and_Troubleshooting/Cloud_Services/Astra)

## 取得協助

NetApp以多種方式支援Astra Data Store。全年無休提供免費的自助服務支援選項、例如知識庫（KB）文章和Slack通路。



您可以取得Astra Data Store的社群技術支援。使用建立案例 "[NetApp支援網站（NSS）](#)" 不適用於此版本。您可以透過意見反應選項與支援部門聯絡、或使用Slack通路進行自助服務。

### 自助服務支援選項

這些選項全年無休免費提供：

- "[知識庫（需要登入）](#)"

搜尋與Astra Data Store相關的文章、常見問題集或中斷修復資訊。

- 文件

這是您目前正在檢視的文件網站。

- "[NetApp「Container」 Slack通路](#)"

前往「Container」通路、與同儕和專家交流。

- 意見反應電子郵件

傳送電子郵件至[astra.feedback@netapp.com](mailto:astra.feedback@netapp.com)、讓我們知道您的想法、想法或疑慮。

### 如需詳細資訊、請參閱

- "[如何將檔案上傳至NetApp（需要登入）](#)"
- "[NetApp知識庫文章](#)"

## 自動支援監控

此功能可監控Astra Data Store系統的執行時間和資訊、並將訊息傳送給NetApp支援部門。AutoSupport視您的組態而定、您可以監控這些系統元件：

- 控制面

- 儲存設備

在期間預設啟用支援AutoSupport "[Astra Data Store叢集安裝](#)" 或者AutoSupport 在叢集套用了某個來源自訂資源（CR）之後。啟用後AutoSupport、即可自動將S還原（ASUP）套裝組合上傳至 "[NetApp支援網站（NSS）](#)" 或可供手動下載。

#### 選項

- [\[AutoSupport triggers and scenarios\]](#)
- [\[Configure custom control plane AutoSupport collection\]](#)
- [\[Configure custom storage AutoSupport collection\]](#)
- [\[List ASUPs in the system\]](#)
- [\[Download an ASUP Bundle\]](#)
- [\[Upload a core file\]](#)

## 可觸發及情境AutoSupport

下列方式觸發了各種套裝組合：AutoSupport

- 定期：ASUP套裝組合是以在CR中定義的時間間隔建立。
- 使用者觸發：您可以手動建立自己的ASUP來查看記錄。
- \* Coredump \*：如果節點上有核心傾印、則會產生ASUP、核心會傳送至NetApp以供進一步調查。
- 基於呼叫首頁事件：ASUP是由作業系統的特定呼叫首頁事件所產生。
- \* Kubernetes事件為基礎\*：ASUP是由控制面板中的特定Kubernetes事件所產生。

這些觸發案例產生下列AutoSupport 其中一種功能類型：

- \* ControlPlane AutoSupport Section\*：Astra Data Store控制平面記錄與CRS的集合。
- 儲存**AutoSupport** 功能：儲存報告與效能資料的集合。
- 核心傾印**AutoSupport** 功能：系統核心傾印的集合。

## 設定自訂控制面板AutoSupport 的資料收集

您可以建立可AutoSupport 報告控制面板事件的自訂資料收集組態。大多數安裝在預設情況下、都會在期間啟用定期事件報告 "[Astra Data Store叢集安裝](#)"。本程序說明如何根據AutoSupport 您所選的參數來設定報告的功能：

#### 步驟

1. 自訂下列命令以建立控制面板集合CR：

```
kubectl astasds asup collect -c controlplane --namespace=astrads-system
```

- a. 定義自訂參數：

- '<myASUPName>'：AutoSupport 要產生的Setcr名稱。

- 「-e <事件名稱>」：觸發集合的事件名稱。事件名稱應預先定義在component.yaml（安裝於支援控制器）中。

範例：

```
kubectl astrasds asup collect -c controlplane custom-asup-name -e debug
--namespace=astrads-system
```

a. 視系統需求新增其他參數：

- 「叢集」：在多叢集環境中需要此旗標。
- 「-localCollection」：啟用本機收藏。預設值為「假」。
- 「-forceupload」：啟用強制上傳。預設值為「假」。
- 「-retry」：啟用重試。預設值為「假」。

## 設定自訂儲存AutoSupport 功能的收藏

您可以建立AutoSupport 可報告儲存元件事件的自訂資料收集組態。大多數安裝在預設情況下、都會在期間啟用定期事件報告 "[Astra Data Store叢集安裝](#)"。本程序說明如何根據AutoSupport 您所選的參數來設定報告的功能：

步驟

1. 自訂下列命令以建立儲存集合CR：

```
kubectl astrasds asup collect -c storage --namespace=astrads-system
```

a. 定義自訂參數：

- '<myASUPName>'：AutoSupport 要產生的Setcr名稱。
- 「-e <事件名稱>」：觸發集合的事件名稱。事件名稱應預先定義在component.yaml（安裝於支援控制器）中。

效能事件範例：

```
kubectl-astrads asup collect -c storage -e performance example-perf-
storage-asup
```

- t <ISO\_format>-d <hides>'：在指定的時間內、為所有節點收集儲存ASUP。使用標準ISO日期時間格式（「-t」）、以小時為單位的持續時間（「-d」）。例如：

```
kubectl astrads asup collect -c storage -t 2021-01-01T15:00:00Z -d 24
```

- -nodes<nodename>'：收集指定節點的儲存ASUP。例如：

```
kubectl astrads asup collect -c storage --nodes example1
```

- -nodes nodename1、nodename2、nodename3：收集指定節點的儲存ASUP：

```
kubectl astrads asup collect -c storage --nodes  
example1,example2,example3
```

a. 視系統需求新增其他參數：

- 「叢集」：在多叢集環境中需要此旗標。
- 「-localCollection」：啟用本機收藏。預設值為「假」。
- 「-forceupload」：啟用強制上傳。預設值為「假」。
- 「-retry」：啟用重試。預設值為「假」。

## 列出系統中的ASUP

使用下列命令、依名稱列出系統中的ASUP：

```
kubectl astrads asup list --namespace=astrads-system
```

回應範例：

NAMESPACE	NAME	SEQUENCE	NUMBER	EVENT
SIZE	STATE	LOCAL	COLLECTION	
astrads-system	storage-callhome.reboot.unknown-...	1		
	callhome.reboot.unknown	0	uploaded	astrads-ds-support-tdl2h:
astrads-system	storage-callhome.reboot.unknown-...	2		
	callhome.reboot.unknown	0	uploaded	astrads-ds-support-xx6n8:
astrads-system	storage-callhome.reboot.unknown-...	3		
	callhome.reboot.unknown	0	uploaded	astrads-ds-support-qghnx:

## 下載ASUP產品組合

您可以使用此命令下載本機收集的ASUP套裝組合。使用「-o」指定目前工作目錄以外的位置：

```
./kubectl-astrasds asup download <ASUP_bundle_name> -o <location>
```

## 上傳核心檔案

如果服務當機、AutoSupport 系統會在當機時建立一個包含相關記憶體內容的資訊（ASUP）訊息（稱為核心檔案）。Astra Data Store會自動將ASUP訊息上傳給NetApp Support、但您需要手動上傳核心檔案、使其與ASUP

訊息相關聯。

## 步驟

1. 使用下列「kubectl」命令來檢視ASUP訊息：

```
kubectl astrads asup list --namespace=astrads-system
```

您應該會看到類似下列的輸出：

NAMESPACE	NAME	SEQUENCE NUMBER	EVENT
SIZE	STATE	LOCAL COLLECTION	
astrads-system	storage-coredump-2021...	1	coredump
197848373	compressed	astrads-ds-support-sxxn7:/var/...	

2. 使用下列「kubectl」命令、從ASUP訊息下載核心檔案。使用「-o」選項來指定下載檔案的目的地目錄。

```
kubectl astrads asup download storage-coredump-20211216t140851311961680  
-o <absolute_path_to_destination_directory>
```



在極少數情況下、您可能無法下載核心檔案、因為其他核心檔案已經取代。發生這種情況時、命令會傳回「無法stat：無此類檔案或目錄」錯誤。如果看到此錯誤、您可以 ["取得協助"](#)。

3. 開啟網頁瀏覽器並瀏覽至 ["NetApp驗證檔案上傳工具"](#)、如果您尚未登入、請輸入您的NetApp支援認證資料。
4. 選取「我沒有個案編號」核取方塊。
5. 在\*最近的地區\*功能表中、選取最靠近您的區域。
6. 選取\*上傳\*按鈕。
7. 瀏覽並選取您先前下載的核心檔案。

隨即開始上傳。上傳完成時、會出現一則成功訊息。

如需詳細資訊、請參閱

- ["如何將檔案上傳至NetApp（需要登入）"](#)

# 舊版Astra Data Store

您可以取得先前版本的文件。

- ["Astra Data Store 212.12文件"](#)

# 法律聲明

法律聲明提供版權聲明、商標、專利等存取權限。

## 版權

<http://www.netapp.com/us/legal/copyright.aspx>

## 商標

NetApp、NetApp 標誌及 NetApp 商標頁面上列出的標章均為 NetApp、Inc. 的商標。其他公司與產品名稱可能為其各自所有者的商標。

<http://www.netapp.com/us/legal/netapptmlist.aspx>

## 專利

如需最新的 NetApp 擁有專利清單、請參閱：

<https://www.netapp.com/us/media/patents-page.pdf>

## 隱私權政策

<https://www.netapp.com/us/legal/privacypolicy/index.aspx>

## 開放原始碼

通知檔案提供有關 NetApp 軟體所使用之協力廠商版權與授權的資訊。

["Astra Data Store EAP版本注意事項"](#)



## 版權資訊

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

衍生自受版權保護之NetApp資料的軟體必須遵守下列授權與免責聲明：

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

NetApp保留隨時變更本文所述之任何產品的權利、恕不另行通知。除非NetApp以書面明確同意、否則NetApp不承擔因使用本文所述產品而產生的任何責任或責任。使用或購買本產品並不代表NetApp擁有任何專利權利、商標權利或任何其他智慧財產權。

本手冊所述產品可能受到一或多個美國國家/地區的保護專利、國外專利或申請中。

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

## 商標資訊

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