



Astra Data Store預覽21.12文件

Astra Data Store

NetApp
May 31, 2022

目錄

Astra Data Store預覽21.12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	4
Astra Data Store預覽簡介	4
Astra Data Store預覽部署模式	6
叢集擴充	7
Astra Data Store預覽的儲存效率	8
Astra Data Store預覽的安全性	8
開始使用	10
Astra Data Store預覽需求	10
Astra Data Store預覽快速入門	13
安裝Astra Data Store預覽	14
設定Astra Data Store預覽元件	30
Astra Data Store預覽限制	41
Astra Data Store預覽的常見問題集	41
使用Astra資料儲存區	44
使用kveccli命令管理Astra Data Store預覽資產	44
部署測試應用程式	50
管理叢集	54
利用功能表監控指標Cloud Insights	66
使用Prometheus和Grafana監控指標	78
設定及監控事件記錄	80
使用Astra Control Center搭配Astra Data Store預覽	80
使用自動指令碼解除安裝Astra Data Store預覽	81
無需指令碼即可解除安裝Astra Data Store預覽	87
知識與支援	94
疑難排解	94
取得協助	94
自動支援監控	94
法律聲明	99
版權	99
商標	99
專利	99
隱私權政策	99
開放原始碼	99

Astra Data Store預覽21.12文件

版本資訊

我們很高興宣布推出Astra Data Store預覽版本的21.12。

- ["本版Astra Data Store預覽的新功能"](#)
- ["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。

本版Astra Data Store的新功能

我們很高興宣布推出Astra Data Store預覽版本的2021.12.12修補程式。

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此產品預覽版本的問題。

使用預設的活動力探針值進行**MongoDB**部署時、**Pod**會在當機迴圈中失敗

因應措施是在MongoDB部署規格中將LIF探針初始延遲時間設為600秒。

Astra Data Store預覽解除安裝程序可能會導致**Pod**處於終止狀態

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

請參閱 ["無需指令碼、即可手動解除安裝Astra Data Store預覽"](#) 或 ["使用指令碼自動解除安裝Astra Data Store預覽"](#)。

如需詳細資訊、請參閱

- ["已知限制"](#)

已知限制

已知限制指出產品預覽版本不支援的平台、裝置或功能、或是無法與產品正確互通的平台、裝置或功能。請仔細檢閱這些限制。

不支援移除一或多個節點的功能

Astra Data Store預覽支援更換故障節點、但不支援節點移除功能。

不支援新增或移除磁碟機的功能

Astra Data Store預覽支援更換故障磁碟機、但不支援新增或移除現有叢集的磁碟機。

Astra Data Store預覽功能未在啟用防火牆的情況下驗證

Astra Data Store預覽需要停用主機firewall。尚未驗證使用Calico HostEndpoint等CNI工具啟用防火牆。

升級或修補程式需要全新安裝

Astra Data Store預覽不適用於正式作業工作負載。

基於**Ubuntu**的裸機或**VM**直通部署需要**NVMe TLC SSD**

此限制不適用於RHEL、RCOOS或CentOS型部署。

如需詳細資訊、請參閱

- ["已知問題"](#)

概念

Astra Data Store預覽簡介

Astra Data Store預覽是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預覽、您可以執行下列動作：

- 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產品系列

Astra系列產品提供Kubernetes應用程式資料生命週期管理功能、可簡化狀態應用程式的作業。輕鬆保護、備份及移轉Kubernetes工作負載、並立即建立可運作的應用程式複本。

Astra產品包括：

- * Astra Control *：使用應用程式感知的資料管理工具、在公有雲和內部部署環境中管理、保護及移動Kubernetes工作負載
 - * Astra Control Service*：使用NetApp管理的服務來管理公有雲中的Kubernetes工作負載。
 - * Astra Control Center*：使用自我管理的軟體來管理內部部署Kubernetes工作負載的資料。
- * Astra Data Store preview *：使用Kubernetes原生的共享檔案服務來處理容器和VM工作負載、以進行企業資料管理。
- * Astra Trident *：使用容器儲存介面（Container Storage Interface、簡稱csi）相容的儲存設備資源配置與管理功能、來處理Kubernetes工作負載、並與NetApp儲存供應商合作。

請參閱 ["Astra系列簡介"](#)。



Astra Data Store預覽功能

Astra Data Store預覽提供端點對端點Kubernetes原生儲存與資料管理功能、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預覽入門

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預覽部署模式

Astra Data Store預覽會使用與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預覽：

- 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預覽是Astra產品系列的一部分。若要深入瞭解整個Astra系列、請參閱 "[Astra系列簡介](#)"。

Astra Data Store預覽生態系統

Astra Data Store預覽可搭配下列功能使用：

- * 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OpenShift Kubernetes叢集、搭配Astra Trident儲存後端ONTAP、搭配使用支援的功能包括：NetApp或Astra Data Store預覽儲存後端。

- *** 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 Data Store預覽上建立磁碟區。

- *** 《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預覽中所管理資源的相關資訊。Cloud Insights

Astra Data Store預覽介面

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預覽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預覽的相關資訊。
- *** kubectl命令***：若要使用Astra Data Store預覽、您可以直接使用kubectl命令。
- *** Kubernetes副檔名***：此外、您也可以使用Astra Data Store預覽Kubernetes API副檔名。

自訂資源定義（CRD）是Kubernetes REST API的延伸、是在部署Astra Data Store預覽運算子時所建立的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會預覽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預覽可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預覽可支援新增具有任何效能功能的節點、只要它們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 Data Store預覽需求"](#)。

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預覽的儲存效率

Astra Data Store預覽採用以NetApp ONTAP 技術為基礎的儲存效率技術、SolidFire 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：ONTAP 採用精簡配置的支援功能、可偵測已清零磁碟區的區域、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ONTAP 只有包含所要求資料的壓縮群組才能解壓縮、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 Data Store預覽的安全性

Astra Data Store預覽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預覽使用下列安全方法：

- 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 Data Store預覽需求

開始驗證您的環境是否符合Astra Data Store預覽要求。

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

Astra Data Store預覽僅支援使用Astra Trident SCSI驅動程式、為Kubernetes工作負載配置持續磁碟區。Astra Data Store未來版本將支援VM工作負載。



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

Kubernetes工作節點資源需求

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

資源	最低	最大值
資料磁碟機數量	<ul style="list-style-type: none">• 3（存在獨立的快取裝置）• 4（如果沒有快取裝置）	14
資料磁碟機大小	100GiB	4TiB
選用快取裝置的數量	1（8GiB或更大）	不適用
vCPU數量	10.	10.
RAM	35GiB	35GiB



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：

- 主機磁碟（開機）上的100GiB或更大可用空間、用於儲存Astra Data Store預覽記錄檔。
- 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。

硬體與軟體需求

Astra Data Store預覽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

- HPE 360
- HPE DL380

- Dell R640
- Dell R740

Astra Data Store預覽已通過下列磁碟機類型的驗證：

- 裸機部署：Astra Data Store預覽安裝在Kubernetes叢集上、直接安裝在Linux叢集上、不需要任何Hypervisor
 - SATA或NVMe TLC SSD
- 虛擬機器型部署：Astra Data Store預覽安裝在ESXi叢集所裝載的Linux VM上的Kubernetes叢集上
 - SATA、SAS或NVMe TLC SSD型資料存放區
 - 以虛擬磁碟或通道磁碟機呈現的磁碟機



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「`dick.enableuid=true`」新增至虛擬機器的進階設定。

軟體

- Hypervisor：Astra Data Store預覽已通過VMware VM部署與ESXi 7.0的驗證。Astra Data Store預覽不支援KVM型部署。
- Astra Data Store預覽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預覽已通過下列Kubernetes發佈版本的驗證：
 - Red Hat OpenShift 4.7
 - Google Anthos 1.7
 - Kubernetes 1.21
 - Kubernetes 1.20



Astra Data Store預覽需要Astra Trident版本210.1才能進行儲存資源配置和協調。請參閱 "[Astra Trident安裝說明](#)"。

網路需求

Astra Data Store預覽每個叢集需要一個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預覽管理介面應與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">• 每個節點兩（2）個：<ul style="list-style-type: none">◦ MI/CIP：每個節點的管理介面上有一（1）個預先設定的IP位址◦ 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
每個節點有兩個網路介面	<ul style="list-style-type: none">• 每個節點三個：<ul style="list-style-type: none">◦ MIP：每個節點的管理介面上有一（1）個預先設定的IP位址◦ 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◦ 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



對於這兩種組態、您應該省略叢集自訂資源（CR）檔案中的資料網路閘道欄位、即「astradscluster.yaml」。每個節點上現有的路由組態可容納所有位址。



這些組態不使用VLAN標記。

Astra Trident

Astra Data Store預覽需要應用程式Kubernetes叢集執行Astra Trident 210.1。Astra Data Store預覽可設定為 "儲存後端" 使用Astra Trident來配置持續磁碟區。

CNI組態

Astra Data Store預覽已通過下列CNI驗證：

- 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

這些CNI需要停用主機防火牆（firewalld）。

持續磁碟區共用需求

每個Astra Data Store預覽叢集都支援使用持續磁碟區來滿足安裝在該叢集上的任何應用程式的儲存需求。Kubernetes應用程式會使用透過NFSv4.1共用的持續磁碟區來存取檔案、這需要AUTH_SYS驗證方法。

授權

Astra Data Store預覽需要Astra Data Store預覽授權才能提供完整功能。 ["請在此註冊"](#) 取得Astra Data Store預覽授權。下載授權的指示將會在您註冊後寄送給您。

組態AutoSupport

Astra Data Store預覽需要AutoSupport 啟用功能才能連線AutoSupport 至該功能的後端。這可能是透過直接的網際網路存取或Proxy組態。

。 ["用於傳送強制遙測AutoSupport 功能套件的定期設定"](#) 不應變更。如果您停用傳送週期AutoSupport 性的更新套件、叢集將會鎖定、而且在重新啟用週期性設定之前、無法建立新的磁碟區。

下一步

檢視 ["快速入門"](#) 總覽：

以取得更多資訊

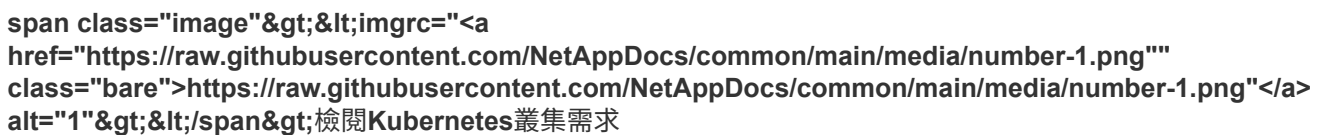
["Astra Data Store預覽限制"](#)

Astra Data Store預覽快速入門

本頁提供Astra Data Store預覽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！如果您想要嘗試Astra Data Store預覽、可以使用90天的預覽授權。

["請在此註冊"](#) 取得Astra Data Store預覽授權。

 <https://raw.githubusercontent.com/NetAppDocs/common/main/media/number-1.png>
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預覽部署中的每個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預覽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 Data Store預覽需求"](#)。

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

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

- 安裝Astra Data Store預覽叢集。
- 設定Astra Data Store預覽監控。
- 如果您使用Red Hat OpenShift、請在Red Hat OpenShift Container Platform（OCP）上安裝Astra Data Store預覽。

深入瞭解 ["安裝Astra Data Store預覽"](#)。

 <https://raw.githubusercontent.com/NetAppDocs/common/main/media/number-3.png>
<https://raw.githubusercontent.com/NetAppDocs/common/main/media/number-3.png>
 there"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預覽設定為儲存後端。
- 建立預設的Astra Data Store預覽儲存類別。

深入瞭解 ["初始設定程序"](#)。

完成Astra Data Store預覽設定之後、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預覽使用kubectl命令"](#)。
- 設定監控端點。深入瞭解 ["設定監控端點"](#)。

["安裝Astra Data Store預覽"](#)。

安裝Astra Data Store預覽

若要安裝Astra Data Store預覽、請從下載安裝套件 ["NetApp 支援網站"](#) 並完成本程序中所述的安裝步驟。

Astra Data Store deployment



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預覽部署"。
- 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、請預覽。
- 答 "NetApp授權檔案 (NLF) " 適用於Astra Data Store預覽。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預覽使用。
- 網際網路連線：Astra Data Store預覽不支援無線環境。需要網際網路連線、才能直接或透過Proxy連線至support.netapp.com。

Astra Data Store預覽安裝程序會引導您完成下列高層級步驟。

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

如果您想要啟用Astra Data Store預覽、以便使用含有機密的影像登錄、請參閱 "此KB"。

下載Astra Data Store預覽套裝組合並擷取影像

1. 登入 "[NetApp 支援網站](#)" 並下載Astra Data Store預覽套裝組合（「2021.12.01_ads.tar」）。
2. （可選）使用以下命令驗證套件的簽名：

```
openssl dgst -sha256 -verify 2021.12_ads.pub -signature 2021.12_ads.sig  
2021.12.01_ads.tar
```

3. 擷取影像：

```
tar -xvf 2021.12.01_ads.tar
```

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複製kbecl-astra二進位檔到安裝k8s kbecl二進位檔的標準路徑、例如：「usr/bin/」。Kustbecl-astrads是自訂的Kvecll擴充功能、可安裝及管理Astra Data Store預覽叢集。

```
cp -p ./bin/kubect1-astrads /usr/bin/.
```

2. 將Astra Data Store預覽影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預覽映像的登錄路徑、例如「po.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 $(ls images/*.tar) ; do
    astraImage=$(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}'~' ./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、可略過此程序。

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預覽元件。

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預覽叢集Pod所需的權限。使用此程序提供Astra Data Store預覽所需的權限（列於範例中）。

將自訂SCC指派給Astra Data Store預覽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預覽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預覽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

```

安裝Astra Data Store預覽作業者

1. 列出Astra Data Store預覽清單：

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

回應：

```
manifests/astradscluster.yaml
manifests/astradsoperator.yaml
manifests/astradsversion.yaml
manifests/monitoring_operator.yaml
```

2. 使用KECBECLL套用部署營運者：

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

回應：



命名空間回應可能會因執行標準安裝或OCP安裝而有所不同。

```
namespace/astrads-system created
customresourcedefinition.apiextensions.k8s.io/astradsautosupports.astrads.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/astradsdeployments.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/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/astradsqospolicies.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-leader-election-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-astradslicense-editor-role  
created  
clusterrole.rbac.authorization.k8s.io/astrads-astradslicense-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-autosupport-editor-role  
created  
clusterrole.rbac.authorization.k8s.io/astrads-autosupport-viewer-role  
created  
clusterrole.rbac.authorization.k8s.io/astrads-manager-role created  
clusterrole.rbac.authorization.k8s.io/astrads-metrics-reader created  
clusterrole.rbac.authorization.k8s.io/astrads-netappexportpolicy-editor-  
role created  
clusterrole.rbac.authorization.k8s.io/astrads-netappexportpolicy-viewer-  
role created  
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsdeployment-  
editor-role created  
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsdeployment-  
viewer-role created  
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsnfsoption-editor-  
role created  
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsnfsoption-viewer-  
role created  
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsnodeinfo-editor-  
role created  
clusterrole.rbac.authorization.k8s.io/astrads-netappsdsnodeinfo-viewer-  
role created  
clusterrole.rbac.authorization.k8s.io/astrads-proxy-role created  
rolebinding.rbac.authorization.k8s.io/astrads-leader-election-  
rolebinding created  
clusterrolebinding.rbac.authorization.k8s.io/astrads-manager-rolebinding  
created  
clusterrolebinding.rbac.authorization.k8s.io/astrads-proxy-rolebinding  
created  
configmap/astrads-autosupport-cm created  
configmap/astrads-firetap-cm created  
configmap/astrads-fluent-bit-cm created
```

```
configmap/astrads-kevents-asup created
configmap/astrads-metrics-cm created
service/astrads-operator-metrics-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預覽版本Yaml

1. 使用KECBECVL套用部署：

```
kubectl apply -f ./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-astradsversion-2jqnk	1/1	Running	0
2m7s			
astrads-ds-nodeinfo-astradsversion-dbk7v	1/1	Running	0
2m7s			
astrads-ds-nodeinfo-astradsversion-rn9tt	1/1	Running	0
2m7s			
astrads-ds-nodeinfo-astradsversion-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預覽授權

1. 請套用您在註冊預覽時取得的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	ADSCLUSTER	VALID	PRODUCT
EVALUATION	ENDDATE	VALIDATED	
p100000006	astrads-example-cluster	true	Astra Data Store Preview
true	2022-01-23	2021-11-04T14:38:54Z	

安裝Astra Data Store預覽叢集

1. 開啟Yaml檔案：

```
vim ./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。

- a. (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](#)。
- b. (必填) 規格：在「show」中變更下列必要值：
 - 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預覽節點一樣多。對於Astra Data Store預覽、這表示至少4個位址、或5個（如果您打算稍後將叢集擴充至5個節點）。
 - 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。



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

- c. (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。
- d. (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。



大部分的意見已從下列Yaml範例中移除。

```
apiVersion: astrads.netapp.io/v1alpha1
kind: AstraDSCluster
metadata:
  name: astrads-cluster-name
  namespace: astrads-system
spec:
  adsNodeConfig:
    cpu: 9
    memory: 34
  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 k8s label key that defines which protection
domain a node belongs to.
# adsProtectionDomainKey: ""
# [Optional] Provide a monitoring config to be used to setup/configure
a monitoring agent.
# monitoringConfig:
  # namespace: "netapp-monitoring"
  # repo: "[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 ./manifests/astradscluster.yaml
```

4. (僅適用於OCP) 如果已啟用SELinux、請在Astra Data Store預覽叢集中的節點上、重新標示下列目錄的「Linux」內容。

```
sudo chcon -R -t container_file_t  
/var/opt/netapp/firetap/rootfs/var/asup/notification/firetap/
```

```
sudo chcon -R -t container_file_t /var/netapp/firetap/firegen/persist/
```



這是因為「Linux」可防止這些目錄可寫入、導致支援Pod進入「CrashLooper (CrashLooper)」狀態、所以需要執行此步驟。此步驟需要在Astra Data Store預覽叢集中的所有節點上執行。

5. 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	1/1	Running	0	7h31m
astrads-deployment-support-788b859c65-2qjkn	3/3	Running	19	12d
astrads-ds-astrads-cluster-lab0dbc-j9jzc	1/1	Running	0	5d2h
astrads-ds-astrads-cluster-lab0dbc-k9wp8	1/1	Running	0	5d1h
astrads-ds-astrads-cluster-lab0dbc-pwk42	1/1	Running	0	5d2h
astrads-ds-astrads-cluster-lab0dbc-qhvc6	1/1	Running	0	8h
astrads-ds-nodeinfo-astradsversion-gcmj8	1/1	Running	1	12d
astrads-ds-nodeinfo-astradsversion-j826x	1/1	Running	3	12d
astrads-ds-nodeinfo-astradsversion-vdthh	1/1	Running	3	12d
astrads-ds-nodeinfo-astradsversion-xwgsf	1/1	Running	0	12d
astrads-ds-support-828vw	2/2	Running	2	5d2h
astrads-ds-support-cfzts	2/2	Running	0	8h
astrads-ds-support-nzkkrr	2/2	Running	15	7h49m
astrads-ds-support-xxbnp	2/2	Running	1	5d2h
astrads-license-controller-86c69f76bb-s6fb7	1/1	Running	0	8h
astrads-operator-79ff8fbb6d-vpz9m	1/1	Running	0	8h

6. 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	2021.10.0	p100000006	
10.x.x.x	10m			

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：

事件訊息	意義
成功選取4個控制面板節點以加入ADS叢集	Astra Data Store預覽操作員識別出足夠的節點、包括CPU、記憶體、儲存設備和網路、以建立Astra Data Store預覽叢集。
正在建立ADS叢集	Astra Data Store預覽叢集控制器已啟動叢集建立作業。
已建立ADS叢集	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>
```

設定Astra Data Store預覽監控

您可以設定Astra Data Store預覽以進行Astra Control Center監控或由其他遙測服務進行監控。

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。

設定Astra Control Center預覽的監控功能

只有在Astra資料儲存區預覽在Astra Control Center中以後端管理之後、才執行下列步驟。

1. 設定Astra Data Store預覽、以供Astra Control Center監控：

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

安裝監控操作員

（選用）如果Astra Data Store預覽無法匯入Astra Control Center、建議使用監控操作員。如果您的Astra Data Store預覽執行個體是獨立部署、使用Cloud Insights 支援功能來監控遙測、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。

1. 執行此安裝命令：

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

2. 設定Astra Data Store預覽以監控：

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

下一步

執行以完成部署 ["設定工作"](#)。

設定Astra Data Store預覽元件

安裝Astra Data Store預覽並解決一些環境先決條件之後、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：

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

安裝Astra Trident

若要進行Astra Data Store預覽、您需要安裝Astra Trident 21.10.1。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預覽部署"。
- 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.sna  
pshot.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：

- a. 將「叢集」值變更為Astra Data Store叢集的叢集名稱。
- b. 將「命名空間」值變更為您要用於建立磁碟區的命名空間。
- c. 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」。
- d. 將您要授予存取權的IP位址填入「AutoExportCIDR」清單。使用「0.0.0.0/0」允許所有人。
- e. 如需「kubeconfig」值、請執行下列動作：

- i. 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 > kubeconfig.json
```

- ii. 將編碼為基64、並使用基64輸出作為「kubeconfig」值：

編碼範例：

```
cat kubeconfig.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": "bronze"
  },
  "storage": [
    {
      "labels": {
        "performance": "extreme"
      },
      "defaults": {
        "qosPolicy": "bronze"
      }
    },
    {
      "labels": {
        "performance": "premium"
      },
      "defaults": {
        "qosPolicy": "bronze"
      }
    },
    {
      "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：

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: trident-csi
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
mountOptions:
  - vers=4.1

```

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 -A
```

回應：

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION	AGE		
trident-csi	csi.trident.netapp.io	Delete	Immediate
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: bronze
          size: ""
          snapshotDir: ""
          snapshotPolicy: ""
        labels:
          performance: extreme
          region: ""
          supportedTopologies: null
          zone: ""
      - defaults:
          exportPolicy: ""
          qosPolicy: bronze
          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:
  - trident-csi
  supportedTopologies: null
volumes: []

```

Astra Data Store預覽限制

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預覽版本具有下列資源限制。

資源	最低	最大值
Astra Data Store預覽叢集中的節點數	4.	5.
每個節點的持續磁碟區數目	不適用	10.
每個節點的持續磁碟區已配置總容量	不適用	1TiB
Volume大小	20億	1TiB
每個Volume的快照	0	256
每個Volume的複本	0	9.



Astra Data Store預覽不支援VM工作負載。VMware vVol工作負載支援將於未來的版本中提供。



Astra Data Store預覽的效能受到限制、不應用於效能特性分析。

Astra Data Store預覽的常見問題集

尋找有關Astra Data Store預覽安裝、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預覽正式作業嗎？*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、但由於Astra Data Store預覽版本並不適用於正式作業工作負載。

我可以使用**Astra Data Store**預覽虛擬機器工作負載嗎？Astra Data Store預覽版本僅限於Kubernetes上執行的應用程式、無論是在裸機或虛擬機器上。未來的版本將支援Kubernetes和ESXi虛擬機器上的應用程式。請參閱["Astra資料儲存區需求"](#)。

- Astra Data Store預覽是否因其運作而與其他NetApp產品有任何相依關係？*是。Astra Data Store預覽需要在工作負載Kubernetes叢集上部署NetApp SCSI驅動程式Astra Trident版本210.1及更新版本。深入瞭解["Astra資料儲存區需求"](#)。

將Astra Data Store預覽叢集當作儲存後端的應用程式可以使用 ["Astra控制中心"](#) 版本21.12可運用應用程式感知的資料管理功能、包括資料保護、災難恢復及Kubernetes工作負載移轉。

*如何管理Astra Data Store預覽叢集？*您可以使用kubectl命令及Kubernetes API擴充功能來管理Astra Data Store預覽資產。

「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預覽叢集指標？*您可以使用Cloud Insights 支援功能來監控Astra Data Store預

覽指標。請參閱 ["利用功能表監控指標Cloud Insights"](#)。

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叢集中使用Astra Data Store預覽功能、以及其他的儲存供應商嗎？*可以。Astra Data Store預覽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預覽中移除Kubernetes叢集、是否會解除安裝Astra Trident？*如果您解除安裝Astra Data Store預覽、則不會從叢集中解除安裝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預覽是否需要授權？*是、Astra Data Store預覽需要NetApp授權檔案（NLF）。

請參閱 ["Astra資料儲存區需求"](#)。

- Astra Data Store預覽授權的有效時間有多長？* Astra Data Store預覽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預覽

*我可以在裸機或虛擬機器上執行的Kubernetes叢集上安裝Astra Data Store預覽嗎？*是。Astra Data Store預覽可安裝在執行裸機或ESXi VM的Kubernetes叢集上。請參閱 ["Astra Data Store預覽需求"](#)。

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

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

*我的Kubernetes叢集大於5個工作節點。我可以在其中安裝Astra Data Store預覽嗎？*是。Astra Data Store預覽叢集可部署在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5個工作節點。

- Astra Data Store預覽是否支援從私有登錄進行離線安裝？*是。Astra Data Store預覽可從本機登錄離線安裝。請參閱 ["安裝Astra Data Store預覽"](#)。不過、Astra Data Store預覽需要連線（直接或透過Proxy）至NetApp AutoSupport 支援區（support.netapp.com）才能繼續營運。

我需要網際網路連線才能使用**Astra Data Store**預覽嗎？Astra Data Store預覽需要連線至NetApp AutoSupport 支援後端、才能定期傳送必要的遙測AutoSupport 功能套件。這種連線方式可以是直接連線、也可以透過Proxy連線。如果缺少此連線功能或AutoSupport 停用了此功能、叢集將會鎖定、新的Volume建立功能將會停用、直到定期重新上傳套件為止。

- Astra Data Store預覽使用哪些角色和權限？*您必須是KUBE管理員、才能部署Astra Data Store預覽操作員。

Astra Data Store預覽含有一組名為「astrads-ds-nodeinfo-astradsversion」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預覽儲存叢集。

*我需要更新哪些資訊清單檔案才能安裝Astra Data Store預覽？*請從下載的Astra Data Store預覽套裝組合下載 ["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預覽"](#)。

疑難排解與支援

透過Astra Data Store預覽、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"

預覽版本要求您的系統必須連線至雲端、並整合至NetApp Active IQ VMware解決方案及AutoSupport VMware工具。

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"](#) 如需有關設定及下載Astra Data Store預覽支援記錄套裝組合的說明、請參閱。

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

*在Astra Data Store預覽上配置磁碟區時、會使用什麼NFS版本？*根據預設、Astra Data Store預覽支援NFS v4.1、適用於所有為Kubernetes應用程式配置的磁碟區。

為什麼即使我已設定**Astra Data Store**預覽搭配大容量磁碟機、我仍無法取得更大的持續磁碟區？Astra Data Store預覽會將節點上所有磁碟區的最大容量限制為1 TiB、以及Astra Data中所有節點的最多5 TiB 儲存預覽叢集。

請參閱 ["Astra Data Store預覽需求"](#) 和 ["Astra Data Store預覽限制"](#)。

升級Astra Data Store預覽

*我可以從Astra Data Store預覽版本升級嗎？*否Astra Data Store預覽不適用於正式作業工作負載、而Astra Data Store預覽軟體的新版本需要全新安裝。

使用Astra資料儲存區

使用kvecII命令管理Astra Data Store預覽資產

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

若要瞭解如何部署範例應用程式、請參閱 ["部署測試應用程式"](#)。

如需下列叢集維護資訊、請參閱 ["管理叢集"](#)：

- 將節點置於維護模式
- 更換磁碟機
- 新增節點
- 更換節點

您需要的是 **#8217** ；需要的是什麼

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

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

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

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

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

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

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

回應：

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

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

```
kubectl get ads -A
```

回應：

```

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

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

NAMESPACE      NAME

```

```

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

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

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

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

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

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

```


3. 使用其中一個簡短名稱來顯示叢集中磁碟區的目前狀態：

```
kubectl get adsvo -A
```

回應：

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

使用KECBECVL副檔名上的說明選項

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

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

```
kubectl astrads -h
```

回應：

```
A kubectl plugin for inspecting your AstraDS deployment

Usage:
  astrads [command]

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

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

operation, this flag can be

groups.

`--cache-dir string`

cache")

`--certificate-authority string`
certificate authority

`--client-certificate string`
for TLS

`--client-key string`

`--cluster string`

cluster to use

`--context string`

context to use

`-h, --help`

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

your HTTPS connections insecure

`--kubeconfig string`
use for CLI requests.

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

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

should contain a

1s, 2m, 3h).

timeout requests.

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

`--token string`
to the API server

`--user string`
to use

repeated to specify multiple

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

Path to a cert file for the

Path to a client certificate file

Path to a client key file for TLS

The name of the kubeconfig

The name of the kubeconfig

help for astrads

If true, the server's

for validity. This will make

Path to the kubeconfig file to

If present, the namespace scope

The length of time to wait

server request. Non-zero values

corresponding time unit (e.g.

A value of zero means don't

(default "0")

The address and port of the

Bearer token for authentication

The name of the kubeconfig user

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

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

回應：

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

Usage:

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

Examples:

```
# Control plane collection
```

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

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

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

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

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

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

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

NOTE:

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

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

Flags:

```
-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          This should be a positive integer  
(default "manual")         Specify the callhome event to trigger.
```

```
-f, --forceUpload           Configure an AutoSupport to upload if  
it is in the compressed state  
and not  
uploading because it was created with  
the 'local' option or if  
automatic uploads of AutoSupports is  
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
--nodes string          the 'compressed' state
                        Specify nodes to collect for storage
component. (default "all")
-t, --startTime string  StartTime is the starting time for
collection of AutoSupport.
                        This should be in the ISO 8601 date
                        time format.
                        Example format accepted:
                        2021-01-01T15:20:25Z, 2021-01-
01T15:20:25-05:00
-u, --usermessage string  UserMessage is the additional message
to include in the
                        AutoSupport subject.
                        (default "Manual event trigger from
CLI")

```

部署測試應用程式

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

在此範例中、我們使用Helm儲存庫部署Bitnami的MongoDB圖表。

您需要的是 **#8217** ；需要的是什麼

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

步驟

1. 從Bitnami新增Helm repo：

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

2. 部署MongoDB：

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

3. 檢查MongoDB Pod的狀態：

```

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

```

4. 驗證MongoDB使用的持續磁碟區宣告 (PVC)：

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

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

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

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

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

```

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

```

```

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

```

```

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

```

管理叢集

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

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

您需要的是 **#8217** ；需要的是什麼

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

新增節點

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

步驟

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


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

回應：

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

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

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

2. 否則、只要新增節點即可：

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

3. 確認已新增節點：

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

將節點置於維護模式

當您需要執行主機維護或套件升級時、應將節點置於維護模式。



節點必須已是Astra Data Store預覽叢集的一部分。

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

步驟

1. 顯示節點詳細資料：

```
~% kubectl get nodes
```

回應：

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

2. 確保節點尚未處於維護模式：

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

回應（維護模式中沒有節點）：

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

3. 啟用維護模式。

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

範例：

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

4. 列出節點。

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

回應：

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

5. 檢查維護模式的狀態：

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

回應：

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

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

6. 節點維護完成後、請停用維護模式：

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

7. 確保節點不再處於維護模式：

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

更換節點

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

步驟

1. 列出所有節點：

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

回應：

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

2. 描述叢集：

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

回應：

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

3. 驗證故障節點上的「Node HA（節點HA）」是否標記為「假」：

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

回應：

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

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

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

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

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

回應：

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

```

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

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

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

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

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

```

```

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

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

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

```

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

5. 驗證節點是否已從叢集移除：

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

回應：

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

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

回應：

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

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

回應：

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

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

回應：

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

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

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

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

回應：

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

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

回應：

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

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


回應：

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

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

回應：

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

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

回應：

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

更換磁碟機

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

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

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

回應：

```

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

```

故障磁碟機CR會在叢集中自動建立、名稱對應於故障磁碟機的UUID。

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

回應：

```

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

```

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

回應：

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

步驟

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

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

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

回應：

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

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

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



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

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

3. 若要重新執行磁碟機更換、請使用之前的命令「-force」：

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

以取得更多資訊

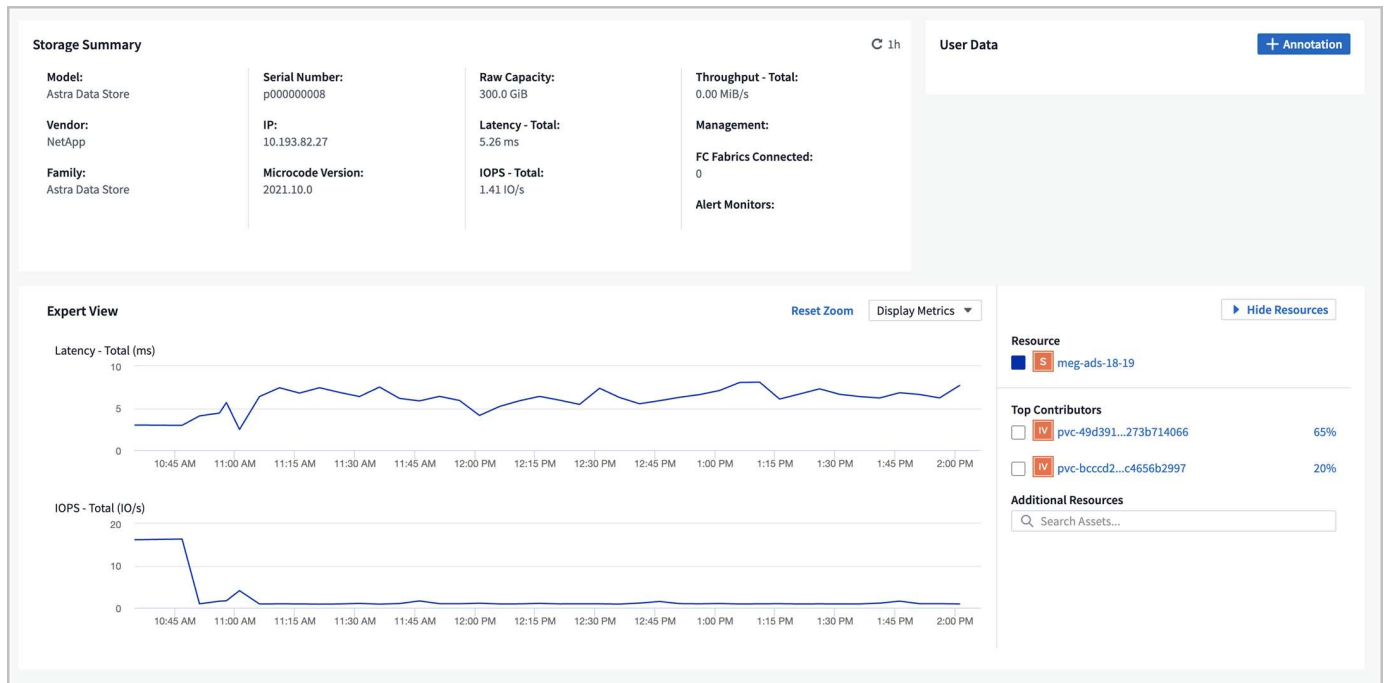
- ["使用kveccli命令管理Astra Data Store預覽資產"](#)

利用功能表監控指標Cloud Insights

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

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

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



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

完成Cloud Insights 連線先決條件工作

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

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

收集下列資訊：

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

擷取單元儲存

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

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

下載並執行安裝指令碼

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

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

然後、會以下列方式傳送指標：

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

顯示安裝程式指令碼說明

安裝程式指令碼的完整說明文字如下所示：

顯示安裝程式指令碼說明文字：

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

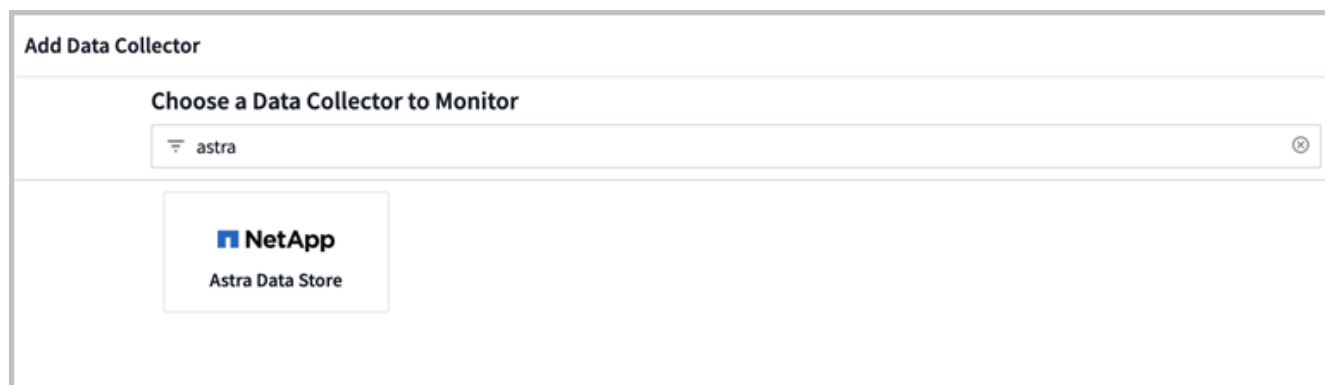
回應：

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

執行安裝指令碼


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

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




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


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



Select a Data Collector



Configure Collector




NetApp
Astra Data Store

Configure Collector

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

What Operating System or Platform Are You Using?


Kubernetes

Select existing API Access Token or create a new one

default_ads_api_key1 (...d0gHof)

[+ API Access Token](#)

[Production Best Practices ?](#)

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

- 1

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

Copy Installer Script

☐ Reveal Installer Script

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

CI_DOMAIN_NAME="f49uaky.gstabler-ads.cloudinsights-dev.netapp.com"
CI_API_KEY="eyJraWQ1OiI5OTk5IiwidHlwIjoI5ldUIiwiaWxnbG9zIjoI5FMzODQifQ.eyJjcVhdG9yTG9naW41OiJhZG1pb2IiImRpc3BsYXl0YW1lIjoIZGVmYXVsdF9hZHNfYXBPX2tleTEgKG9uIGJlaGFsZiBvZiBhZG1pb2Ipbik"

```
- 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*l should point to the *kubernetes* cluster to be monitored.
- 8

Complete Setup

8. 指令碼完成後、按一下*完成設定*。

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



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

安裝指令碼會在組態成功時產生類似的輸出：

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

代理程式CR範例

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

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

```

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

```

編輯Cloud Insights 此鏈接

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

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

更新Kubernetes API權杖

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

更新Cloud Insights 程式：更新程式碼

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

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

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

此區段如下所示：

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

6. 儲存並結束編輯器視窗。

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

中斷Cloud Insights 與該功能的連線

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

移除Astra Data Store預覽收集器

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

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

移除擷取單元、Telegraf和Fluent位元

1. 編輯Agent CR:

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

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

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

7. 如果您將本機目錄用於擷取單元PV、而非儲存資源配置程式、請刪除PV：

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

然後、刪除正在執行擷取單元的節點上的實際目錄。

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

監控操作員會更新Telegraf和Fluent位元組態、並移除擷取單元。

Open Metrics API說明

以下是可用來從Astra Data Store預覽收集度量的API清單。

- 「說明」行說明指標。
- 「類型」行指出度量是量表還是計數器。

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

```

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

```

```

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

```

```

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

```

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

使用Prometheus和Grafana監控指標

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

您需要的是 **#8217** ；需要的是什麼

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

設定Prometheus

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

步驟

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

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

3. 啟動Prometheus服務。

設定Grafana

您可以設定Grafana以顯示Prometheus收集的指標。

步驟

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

```
apiVersion: 1

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

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

匯入Grafana儀表板範本

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

步驟

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

設定及監控事件記錄

若要監控事件管理系統（EMS）記錄、您可以執行下列高層級工作：

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

在Astra Data Store預覽叢集自訂資源（CR）中設定監控

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

輸入：

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

其中：

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

設定Cloud Insights 功能

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

串流事件記錄至Elastic

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

輸入：

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



彈性主機名稱可以是IP位址。

使用Astra Control Center搭配Astra Data Store預覽

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

設定Astra Control Center進行Astra Data Store預覽

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

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



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

您可以在Astra控制中心做什麼

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

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

以取得更多資訊

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

使用自動指令碼解除安裝Astra Data Store預覽

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

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

您需要的是 **#8217** ；需要的是什麼

- root系統管理權限

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

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

移除現有的工作負載和繫結

在解除安裝Astra Data Store預覽之前、您必須先移除下列項目

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

如此可確保Kubernetes環境保持乾淨狀態、這在重新安裝時非常重要。

解除安裝Astra Data Store叢集

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

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

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

3. 執行下列指令碼、指出您要解除安裝的項目：

```
./uninstall.sh

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

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

否則、如果您想要解除安裝所有項目、請輸入「uninstall.sh」



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

5. 出現提示時、請確認您要繼續、然後輸入「eraseDDATA」

回應：

```
./uninstall.sh all

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

```

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

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

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

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

```

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

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

```

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

```

驗證刪除astrad-system命名空間

請確定下列命令沒有傳回任何結果：

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

確保容器未在工作節點上執行

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

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

刪除OpenShift Container Platform資源

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

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

完成標準的解除安裝程序之後、請完成下列步驟。

1. 移除SCC資源：

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

2. 移除角色繫結資源：

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



請忽略這些步驟中的「找不到資源」錯誤。

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

Astra Data Store預覽解除安裝程序疑難排解

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

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

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

無需指令碼即可解除安裝Astra Data Store預覽

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

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

您需要的是 **#8217** ；需要的是什麼

- root系統管理權限

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

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

移除現有的工作負載和繫結

在解除安裝Astra Data Store預覽之前、您必須先移除下列項目

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

如此可確保Kubernetes環境保持乾淨狀態、這在重新安裝時非常重要。

解除安裝Astra Data Store預覽叢集和控制面板

請依照下列步驟手動解除安裝Astra Data Store預覽。

刪除磁碟區並匯出原則

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



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

步驟

1. 刪除磁碟區：

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

2. 刪除匯出原則：

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

刪除Astra Data Store預覽叢集

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



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

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

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

1. 刪除叢集：

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

刪除授權

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

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

刪除Astra Data Store預覽安裝

刪除叢集中的控制器、運算子、命名空間和支援Pod。

1. 刪除Astra Data Store預覽安裝物件：

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

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

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

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

3. 刪除剩餘成品和運算子yaml檔案：

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

驗證刪除astrad-system命名空間

請確定下列命令沒有傳回任何結果：

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

確保容器未在工作節點上執行

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

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

刪除OpenShift Container Platform資源

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

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

完成標準的解除安裝程序之後、請完成下列步驟。

1. 移除SCC資源：

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

2. 移除角色繫結資源：

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



請忽略這些步驟中的「找不到資源錯誤」。

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

手動刪除範例

以下是執行手動解除安裝指令碼的範例。

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

```

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

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

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

```

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

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

$ kubectl get ns | grep astrads-system

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

Astra Data Store預覽解除安裝程序疑難排解

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

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

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

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

取得協助

NetApp以多種方式支援Astra Data Store預覽。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預覽的社群技術支援。使用建立案例 "[NetApp支援網站（NSS）](#)" 無法用於預覽版本。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預覽相關的文章、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、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預覽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預覽叢集安裝](#)" 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預覽控制平面記錄和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預覽叢集安裝](#)"。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預覽叢集安裝](#)"。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-astrads 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 Preview會自動將ASUP訊息上傳至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）"](#)

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 21.12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。