



# 開始使用

## Astra Data Store

NetApp  
May 16, 2022

# 目錄

開始使用 .....	1
Astra Data Store預覽需求 .....	1
Astra Data Store預覽快速入門 .....	4
安裝Astra Data Store預覽 .....	5
設定Astra Data Store預覽元件 .....	21
Astra Data Store預覽限制 .....	32
Astra Data Store預覽的常見問題集 .....	32

# 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"><li>• 3（存在獨立的快取裝置）</li><li>• 4（如果沒有快取裝置）</li></ul>	14
資料磁碟機大小	100GiB	4TiB
選用快取裝置的數量	1（8GiB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"><li>• 每個節點兩（2）個：<ul style="list-style-type: none"><li>◦ MI/CIP：每個節點的管理介面上有一（1）個預先設定的IP位址</li><li>◦ 資料IP：每個節點的一（1）個未使用或未設定的IP位址、與MIP位於同一子網路中</li></ul></li></ul>
每個節點有兩個網路介面	<ul style="list-style-type: none"><li>• 每個節點三個：<ul style="list-style-type: none"><li>◦ MIP：每個節點的管理介面上有一（1）個預先設定的IP位址</li><li>◦ CIP：每個節點的資料介面上有一（1）個預先設定的IP位址、與MIP位於不同的子網路中</li><li>◦ 資料IP：在CIP所在的同一子網路中、每個節點有一（1）個未使用或未設定的IP位址</li></ul></li></ul>



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

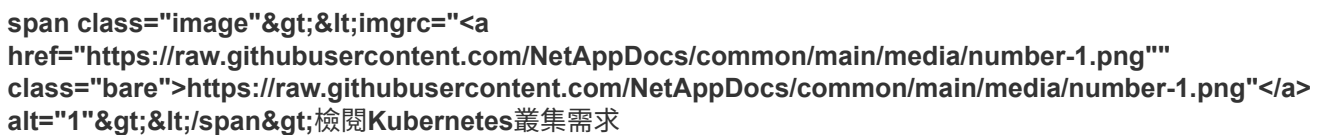
["Astra Data Store預覽限制"](#)

## 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。

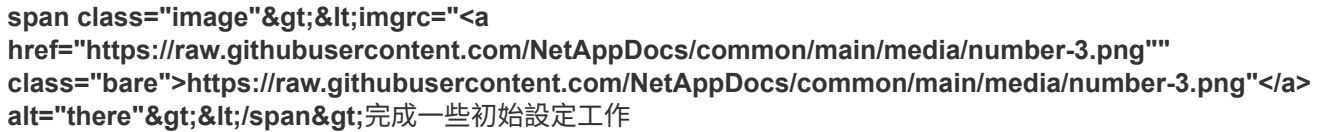
深入瞭解 ["Astra Data Store預覽需求"](#)。

 [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>  
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？\*

是的。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。

## Copyright Information

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

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

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

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

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

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

## Trademark Information

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