開始使用
Astra Control Center

NetApp
November 21, 2023

目錄

開始使用

Astra Control Center需求

開始驗證作業環境、應用程式叢集、應用程式、授權和網頁瀏覽器的整備度。

營運環境需求

Astra Control Center需要下列其中一種作業環境：

- Kubernetes 1.20至1.23
- Rancher 2.5.8、2.5.9或2.6（含RKE1）
- Red Hat OpenShift Container Platform 4.6-8、4.7、4.8或4.9
- VMware Tanzu Kubernetes Grid 1.4
- VMware Tanzu Kubernetes Grid整合版1.12.2

確保您選擇裝載Astra Control Center的作業環境符合環境正式文件中所述的基本資源需求。除了環境的資源需求之外、Astra Control Center還需要下列資源：

元件	需求
儲存後端容量	至少可提供500GB容量
工作節點	總共至少3個工作節點、每個節點有4個CPU核心和12GB RAM
FQDN位址	Astra Control Center的FQDN位址
Astra Trident	• 已安裝並設定Astra Trident 21.004或更新版本 • 如果Astra Data Store將用作儲存後端、則已安裝並設定Astra Trident 210.1或更新版本

這些需求假設Astra Control Center是營運環境中唯一執行的應用程式。如果環境正在執行其他應用程式、請相應調整這些最低需求。

- 映像登錄：您必須擁有現有的私有Docker映像檔登錄、才能將Astra Control Center建置映像推入其中。您需要提供映像登錄的URL、以便上傳映像。
- * Astra Trident / ONTAP S基 類組態*：Astra Control Center需要建立儲存類別、並將其設為預設儲存類別。Astra Control Center支援ONTAP Astra Trident提供的下列支援資訊驅動程式：
  - ONTAP-NAS
  - ONTAP-SAN
  - ONTAP-san經濟型

在OpenShift環境中進行應用程式複製時、Astra Control Center需要允許OpenShift掛載磁碟區並變更檔案的擁有權。因此、您必須設定ONTAP 一個不中斷的Volume匯出原則、才能執行這些作業。您可以使用下列命令來執行此作業：

1. 「匯出原則規則modify -vserver <儲存虛擬機器名稱>-policyname <原則名稱>-rueindex 1 -超級使用者sys」
2. 「匯出原則規則修改-vserver <儲存虛擬機器名稱>-policyname <原則名稱>-rueindex 1 -anon 65534」

如果您計畫將第二個OpenShift作業環境新增為受管理的運算資源、則必須確保啟用Astra Trident Volume Snapshot功能。若要使用Astra Trident啟用及測試Volume快照、 "請參閱官方的Astra Trident說明"。

VMware Tanzu Kubernetes Grid叢集需求

在VMware Tanzu Kubernetes Grid（TKG）或Tanzu Kubernetes Grid整合版（TKGi）叢集上裝載Astra Control Center時、請謹記下列考量事項。

- 停用要由Astra Control管理的任何應用程式叢集上的TKG或TKGi預設儲存類別強制。您可以編輯命名空間叢集上的「TanzuKubernetesCluster」資源來執行此作業。
- 您必須建立安全原則、讓Astra Control Center能夠在叢集中建立Pod。您可以使用下列命令來執行此作業：

```
kubectl config use-context <context-of-workload-cluster>
kubectl create clusterrolebinding default-tkg-admin-privileged-binding
--clusterrole=psp:vmware-system-privileged --group=system:authenticated
```

- 在TKG或TKGi環境中部署Astra Control Center時、請注意Astra Trident的特定需求。如需詳細資訊、請參閱 "Astra Trident文件"。

預設的VMware TKG和TKGi組態檔案權杖會在部署後10小時內過期。如果您使用Tanzu產品組合產品、則必須產生一個含有非過期權杖的Tanzu Kubernetes叢集組態檔、以避免Astra Control Center與託管應用程式叢集之間發生連線問題。如需相關指示、請造訪 "VMware NSxT-T資料中心產品文件。"

支援的儲存後端

Astra Control Center支援下列儲存後端。

- Astra資料儲存區
- NetApp ONTAP S9.5或更新AFF 版本的功能性和FAS 功能性系統
- NetApp Cloud Volumes ONTAP

應用程式叢集需求

Astra Control Center對於您計畫從Astra Control Center管理的叢集有下列需求。如果您打算管理的叢集是裝載Astra Control Center的作業環境叢集、則也適用這些需求。

- Kubernetes的最新版本 "Snapshot控制器元件" 已安裝
- 阿斯特拉部落 "volumesnapshotClass物件" 已由系統管理員定義
- 叢集上存在預設的Kubernetes儲存類別
- 至少有一個儲存類別設定為使用Astra Trident

> 您的應用程式叢集應該有一個「kubeconfig・yaml」檔案、只定義一個_context_元素。請參閱的Kubernetes文件 "建立Kbeconfig檔案的相關資訊"。

> 在Rancher環境中管理應用程式叢集時、請修改Rancher提供的「kubeconfig」檔案中應用程式叢集的預設內容、以使用控制面內容、而非Rancher API伺服器內容。如此可減少Rancher API伺服器的負載、並改善效能。

應用程式管理需求

Astra Control具備下列應用程式管理需求：

- 授權：若要使用Astra Control Center管理應用程式、您需要Astra Control Center授權。
- 命名空間：Astra Control要求應用程式不超過一個命名空間、但命名空間可以包含多個應用程式。
- * StorageClass *：如果您安裝的應用程式已明確設定StorageClass、且需要複製應用程式、則複製作業的目標叢集必須具有原本指定的StorageClass。將具有明確設定StorageClass的應用程式複製到沒有相同StorageClass的叢集、將會失敗。
- * Kubernetes資源*：使用未由Astra Control收集之Kubernetes資源的應用程式、可能沒有完整的應用程式資料管理功能。Astra Control會收集下列Kubernetes資源：

叢集角色	ClusterRoleBinding	組態對應
可關係工作	CustomResourceDesDefinition	CustomResource
示範	部署組態	HorizontalPodAutoscaler
入侵	互鎖Webhook	網路原則
PeristentVolume Claim	Pod	Podcast中斷預算
Podcast範本	ReplicaSet	角色
角色繫結	路由	秘密
服務	服務帳戶	狀態集
驗證Webhook		

支援的應用程式安裝方法

Astra Control支援下列應用程式安裝方法：

- 資訊清單檔案：Astra Control支援使用KUbectl從資訊清單檔案安裝的應用程式。例如：

```
kubectl apply -f myapp.yaml
```

- * Helm 3*：如果您使用Helm來安裝應用程式、Astra Control需要Helm版本3。完全支援使用Helm 3（或從Helm 2升級至Helm 3）來管理及複製安裝的應用程式。不支援管理以Helm 2安裝的應用程式。
- 操作員部署的應用程式：Astra Control支援以命名空間範圍運算子安裝的應用程式。以下是已針對此安裝模式驗證的一些應用程式：
  - "Apache K8ssandra"
  - "Jenkins CI"
  - "Percona XtraDB叢集"

> 運算子及其安裝的應用程式必須使用相同的命名空間；您可能需要修改運算子的部署.yaml檔案、以確保情況如此。

存取網際網路

您應該判斷是否有外部網際網路存取權。如果您沒有、部分功能可能會受到限制、例如從NetApp Cloud Insights接收監控和數據資料、或是將支援組合傳送至 "NetApp 支援網站"。

授權

Astra Control Center需要Astra Control Center授權才能提供完整功能。向NetApp取得評估授權或完整授權。若無授權、您將無法：

- 定義自訂應用程式
- 建立現有應用程式的快照或複本
- 設定資料保護原則

如果您想要試用Astra控制中心、您可以 "使用90天試用版授權"。

若要深入瞭解授權的運作方式、請參閱 "授權"。

內部部署**Kubernetes**叢集的入口

您可以選擇網路入侵Astra控制中心的用途類型。依預設、Astra Control Center會將Astra Control Center閘道（服務/網路）部署為整個叢集的資源。Astra Control Center也支援使用服務負載平衡器（如果環境允許）。如果您想要使用服務負載平衡器、但尚未設定一個、則可以使用MetalLB負載平衡器自動將外部IP位址指派給服務。在內部DNS伺服器組態中、您應該將Astra Control Center所選的DNS名稱指向負載平衡的IP位址。

> 如果您要在Tanzu Kubernetes Grid叢集上裝載Astra Control Center、請使用「kubecl Get nslb監控器-A」命令、查看您是否已設定服務監控器來接受入口流量。如果存在、則不應安裝MetalLB、因為現有的服務監視器將會覆寫任何新的負載平衡器組態。

如需詳細資訊、請參閱 "設定入口以進行負載平衡"。

網路需求

裝載Astra Control Center的作業環境會使用下列TCP連接埠進行通訊。您應確保這些連接埠可透過任何防火牆、並設定防火牆、以允許來自Astra網路的任何HTTPS輸出流量。有些連接埠需要在裝載Astra Control Center的環境與每個託管叢集之間進行連線（視情況而定）。

您可以在雙堆疊Kubernetes叢集中部署Astra Control Center、Astra Control Center則可管理已設定為雙堆疊作業的應用程式和儲存後端。如需雙堆疊叢集需求的詳細資訊、請參閱 "Kubernetes文件"。

來源	目的地	連接埠	傳輸協定	目的
用戶端PC	Astra控制中心	443..	HTTPS	UI / API存取：確保此連接埠在裝載Astra Control Center的叢集與每個受管理叢集之間都開啟
度量使用者	Astra Control Center工作節點	9090	HTTPS	度量資料通訊：確保每個託管叢集都能存取裝載Astra Control Center的叢集上的此連接埠（需要雙向通訊）
Astra控制中心	託管Cloud Insights版的服務	443..	HTTPS	通訊Cloud Insights
Astra控制中心	Amazon S3儲存貯體供應商	443..	HTTPS	Amazon S3儲存通訊
Astra控制中心	NetApp AutoSupport	443..	HTTPS	NetApp AutoSupport通訊

支援的網頁瀏覽器

Astra Control Center支援最新版本的Firefox、Safari和Chrome、最低解析度為1280 x 720。

下一步

檢視 "快速入門" 總覽：

Astra Control Center快速入門

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

歡迎試用！如果您想要試用Astra Control Center、可以使用90天試用版授權。請參閱 "授權資訊" 以取得詳細資料。

1 檢閱**Kubernetes**叢集需求

- Astra可搭配Kubernetes叢集、搭配Trident設定ONTAP 的支援功能、包括後端的功能、或是Astra Data Store儲存後端。
- 叢集必須以健全狀態執行、且至少有三個線上工作者節點。
- 叢集必須執行Kubernetes。

"深入瞭解Astra Control Center需求"。

2 下載並安裝**Astra Control Center**

- 從下載Astra Control Center "NetApp支援網站Astra Control Center下載頁面"。
- 在您的本機環境中安裝Astra Control Center。

  或者、使用Red Hat作業系統集線器安裝Astra Control Center。

"深入瞭解如何安裝Astra Control Center"。

3 完成一些初始設定工作

- 新增授權。
- 新增Kubernetes叢集、Astra Control Center會探索詳細資料。
- 新增ONTAP 一個功能不一的 "Astra資料儲存區" 儲存後端：
- 或者、您也可以新增物件存放區、以儲存應用程式備份。

"深入瞭解初始設定程序"。

4 使用**Astra**控制中心

完成Astra Control Center的設定之後、接下來您可以：

- 管理應用程式。 "深入瞭解如何管理應用程式"。
- 您也可以選擇連線至NetApp Cloud Insights 解決方案、在Astra Control Center UI內顯示系統健全狀況、容量和處理量的指標。 "深入瞭解連線Cloud Insights 到NetApp"。

5 請從本快速入門繼續

"安裝Astra Control Center"。

如需詳細資訊、請參閱

- "使用Astra Control API"

安裝總覽

選擇並完成下列其中一個Astra Control Center安裝程序：

- "使用標準程序安裝Astra Control Center"
- "（如果您使用Red Hat OpenShift）使用OpenShift作業系統集線器安裝Astra Control Center"
- "安裝Astra Control Center搭配Cloud Volumes ONTAP 一套功能性儲存後端"

使用標準程序安裝Astra Control Center

若要安裝Astra Control Center、請從NetApp支援網站下載安裝套件、並執行下列步驟、在您的環境中安裝Astra Control Center操作員和Astra Control Center。您可以使用此程序、在連線網際網路或無線環境中安裝Astra Control Center。

對於Red Hat OpenShift環境、您也可以使用 "替代程序" 使用OpenShift作業系統集線器安裝Astra Control Center。

您需要的產品

- "開始安裝之前、請先準備好環境以進行Astra Control Center部署"。
- 確保所有叢集操作員都處於健全狀態且可用。

  OpenShift範例：

  ```
  oc get clusteroperators
  ```

- 確保所有API服務均處於健全狀態且可供使用：

  OpenShift範例：

  ```
  oc get apiservices
  ```

- 您打算使用的Astra FQDN必須可路由傳送至此叢集。這表示您在內部DNS伺服器中有DNS項目、或是使用已註冊的核心URL路由。

關於這項工作

Astra Control Center安裝程序會執行下列作業：

- 將Astra元件安裝到「NetApp-acc」（或自訂命名）命名空間。
- 建立預設帳戶。
- 為此Astra Control Center執行個體建立預設的管理使用者電子郵件地址和預設的一次性密碼「ACC-<UUUID__of_installation_>」。此使用者被指派系統中的擁有者角色、第一次登入UI時即需要此角色。
- 協助您判斷所有Astra Control Center Pod都在執行中。
- 安裝Astra UI。

> （僅適用於Astra Data Store Early Access Program（EAP）版本）如果您打算使用Astra Control Center管理Astra Data Store並啟用VMware工作流程、 只能在「pCloud」命名空間上部署Astra Control Center、而不能在本程序步驟所述的「NetApp-acc」命名空間或自訂命名空間上部署。

> 請勿在整個安裝程序期間執行下列命令、以免刪除所有Astra Control Center Pod：「kubecl DELETE -f Astra _control_center_oper_deploy。yaml」

如果您使用的是Red Hat的Podman而非Docker Engine、則可以使用Podman命令來取代Docker命令。

步驟

若要安裝Astra Control Center、請執行下列步驟：

- 下載並解壓縮Astra Control Center套裝組合
- 安裝NetApp Astra kubecl外掛程式
- [將映像新增至本機登錄]
- [設定具有驗證需求之登錄的命名空間和機密]
- 安裝Astra Control Center操作員
- 設定Astra控制中心
- 完整的Astra控制中心和操作員安裝
- [驗證系統狀態]
- [設定入口以進行負載平衡]
- 登入Astra Control Center UI

下載並解壓縮**Astra Control Center**套裝組合

1. 請從下載Astra Control Center套裝組合（「Astra控制中心-[版本].tar.gz」） "NetApp 支援網站"。
2. 從下載Astra Control Center認證與金鑰的壓縮檔 "NetApp 支援網站"。
3. （可選）使用以下命令驗證套件的簽名：

```
openssl dgst -sha256 -verify astra-control-center[version].pub
-signature <astra-control-center[version].sig astra-control-
center[version].tar.gz
```

4. 擷取影像：

```
tar -vxzf astra-control-center-[version].tar.gz
```

安裝**NetApp Astra kubecl**外掛程式

NetApp Astra「kubecll」命令列外掛程式可在執行與部署及升級Astra Control Center相關的一般工作時節省時間。

您需要的產品

NetApp為不同的CPU架構和作業系統提供外掛程式的二進位檔。執行此工作之前、您必須先瞭解您的CPU和作業系統。在Linux和Mac作業系統上、您可以使用「uname -A」命令來收集此資訊。

步驟

1. 列出可用的NetApp Astra「kubecll」外掛程式二進位檔、並記下作業系統和CPU架構所需的檔案名稱：

```
ls kubectl-astra/
```

2. 將檔案複製到與標準「kubecll」公用程式相同的位置。在此範例中、「kubecll」公用程式位於「usr/local/bin」目錄中。將「<二進位名稱>」取代為您所需的檔案名稱：

```
cp kubectl-astra/<binary-name> /usr/local/bin/kubectl-astra
```

將映像新增至本機登錄

1. 變更至Astra目錄：

```
cd acc
```

2. 將Astra Control Center映像目錄中的檔案新增至本機登錄。

   請參閱以下自動載入影像的範例指令碼。

   a. 登入您的登錄：

      Docker：

      ```
      docker login [your_registry_path]
      ```

      Podcast：

      ```
      podman login [your_registry_path]
      ```

   b. 使用適當的指令碼來載入映像、標記映像、並[Subforte_image_local_register_pip]將映像推送到本機登錄：

      Docker：

```
export REGISTRY=[Docker_registry_path]
for astraImageFile in $(ls images/*.tar) ; do
  # Load to local cache. And store the name of the loaded image
trimming the 'Loaded images: '
  astraImage=$(docker load --input ${astraImageFile} | sed 's/Loaded
image: //')
  astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
  # Tag with local image repo.
  docker tag ${astraImage} ${REGISTRY}/${astraImage}
  # Push to the local repo.
  docker push ${REGISTRY}/${astraImage}
done
```

Podcast：

```
export REGISTRY=[Registry_path]
for astraImageFile in $(ls images/*.tar) ; do
  # Load to local cache. And store the name of the loaded image trimming
the 'Loaded images: '
  astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image(s): //')
  astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
  # Tag with local image repo.
  podman tag ${astraImage} ${REGISTRY}/${astraImage}
  # Push to the local repo.
  podman push ${REGISTRY}/${astraImage}
done
```

設定具有驗證需求之登錄的命名空間和機密

1. 如果您使用需要驗證的登錄、則需要執行下列動作：

   a. 建立「NetApp-acc operator：

   ```
   kubectl create ns netapp-acc-operator
   ```

   回應：

   ```
   namespace/netapp-acc-operator created
   ```

   b. 建立「NetApp-acc operator」命名空間的秘密。新增Docker資訊並執行下列命令：

```
kubectl create secret docker-registry astra-registry-cred -n netapp-
acc-operator --docker-server=[your_registry_path] --docker
-username=[username] --docker-password=[token]
```

回應範例：

```
secret/astra-registry-cred created
```

c. 建立「NetApp-acc」（或自訂命名）命名空間。

```
kubectl create ns [netapp-acc or custom namespace]
```

回應範例：

```
namespace/netapp-acc created
```

d. 為「NetApp-acc」（或自訂命名）命名空間建立秘密。新增Docker資訊並執行下列命令：

```
kubectl create secret docker-registry astra-registry-cred -n [netapp-
acc or custom namespace] --docker-server=[your_registry_path]
--docker-username=[username] --docker-password=[token]
```

回應

```
secret/astra-registry-cred created
```

a. [Substete_kubeconfig _secret]（選用）如果您希望叢集在安裝後由Astra Control Center自動管理、請確定您在Astra Control Center命名空間中提供了要使用此命令部署的kubeconfig作為機密：

```
kubectl create secret generic [acc-kubeconfig-cred or custom secret
name] --from-file=<path-to-your-kubeconfig> -n [netapp-acc or custom
namespace]
```

安裝**Astra Control Center**操作員

1. 編輯Astra Control Center營運者部署Yaml（「Astra _control_center_operer_deploy」、以參照您的本機登錄和機密。

```
vim astra_control_center_operator_deploy.yaml
```

a. 如果您使用需要驗證的登錄、請將預設行「imagePullSecrets：[]」改為：

```
imagePullSecrets:
- name: <name_of_secret_with_creds_to_local_registry>
```

b. 將「kube-RBAC代理」映像的「[your _register_path]」變更為您將映像推入的登錄路徑 上一步。

c. 將「acc oper-manager-manager」映像的「[your _register_path]」變更為您將映像推入的登錄路徑 上一步。

d. （若為使用Astra Data Store預覽的安裝）請參閱此已知問題 "儲存類別資源配置工具、以及您需要對Y反 洗錢進行的其他變更"。

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: [your_registry_path]/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        command:
        - /manager
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        image: [your_registry_path]/acc-operator:[version x.y.z]
        imagePullPolicy: IfNotPresent
      imagePullSecrets: []
```

2. 安裝Astra Control Center操作員：

```
kubectl apply -f astra_control_center_operator_deploy.yaml
```

回應範例：

```
namespace/netapp-acc-operator created
customresourcedefinition.apiextensions.k8s.io/astracontrolcenters.astra.
netapp.io created
role.rbac.authorization.k8s.io/acc-operator-leader-election-role created
clusterrole.rbac.authorization.k8s.io/acc-operator-manager-role created
clusterrole.rbac.authorization.k8s.io/acc-operator-metrics-reader
created
clusterrole.rbac.authorization.k8s.io/acc-operator-proxy-role created
rolebinding.rbac.authorization.k8s.io/acc-operator-leader-election-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-manager-
rolebinding created
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-proxy-
rolebinding created
configmap/acc-operator-manager-config created
service/acc-operator-controller-manager-metrics-service created
deployment.apps/acc-operator-controller-manager created
```

設定**Astra**控制中心

1. 編輯Astra Control Center自訂資源（CR）檔案（「Astra_control_center_min.yaml」）、以建立帳戶、AutoSupport 供參考、登錄及其他必要的組態：

   > 如果您的環境需要額外的自訂功能、您可以使用「Astra_control_center.yaml」作為替代的CR。「Astra_control_center_min.yaml」是預設的CR、適用於大部分的安裝。

   ```
   vim astra_control_center_min.yaml
   ```

   > 在初始Astra Control Center部署之後、無法變更由CR設定的內容。

   > 如果您使用不需要授權的登錄、則必須刪除「imageRegistry」中的「秘密」行、否則安裝將會失敗。

   a. 將「[your _register_path]（您的登錄路徑）」變更為您在上一個步驟中推送影像的登錄路徑。
   b. 將「帳戶名稱」字串變更為您要與帳戶建立關聯的名稱。
   c. 將「astraAddress」字串變更為您要在瀏覽器中用來存取Astra的FQDN。請勿在地址中使用「http：//」或「https：//」。複製此FQDN以供在中使用 後續步驟。
   d. 將「電子郵件」字串變更為預設的初始系統管理員地址。複製此電子郵件地址以供在中使用 後續步驟。
   e. 如果網站沒有網際網路連線、請將AutoSupport 「已註冊」改為「假」、否則連線網站則保留「真」。
   f. （選用）新增與帳戶相關之使用者的名字「firstName」和姓氏「lastName」。您可以在UI中立即或稍後

執行此步驟。

g. （可選）如果安裝需要、請將「儲存類別」值變更為其他Trident storageClass資源。

h. （選用）如果您希望叢集在安裝後由Astra Control Center自動管理、而且您已經擁有 已建立包含此叢集之Kbeconfig的秘密下、在這個名為「astraKubeConfigSecret："Acc－kubeconfig－cred or custom secret name"的Yaml檔案中新增欄位、以提供密碼名稱

i. 請完成下列其中一個步驟：

- 其他入侵控制器（擷取類型：一般）：這是Astra控制中心的預設動作。部署Astra Control Center之後、您需要設定入口控制器、以URL顯示Astra Control Center。

  預設的Astra Control Center安裝會將其閘道（「服務/網路」）設定為「ClusterIP」類型。此預設安裝需要您額外設定Kubernetes IngresController / Ingress、才能將流量路由傳送至該控制器。如果您想要使用入口、請參閱 "設定入口以進行負載平衡"。

- 服務負載平衡器（擷取類型：**AccTraefik**）：如果您不想安裝IngressController或建立Ingress資源、請將「ingressType」設為「AccTraefik」。

  這會將Astra Control Center「truefik」閘道部署為Kubernetes負載平衡器類型服務。

  Astra Control Center使用類型為「loadbalancer」（Astra Control Center命名空間中的「shvC/truefik」）的服務、並要求指派可存取的外部IP位址。如果您的環境允許負載平衡器、但您尚未設定負載平衡器、則可以使用MetalLB或其他外部服務負載平衡器、將外部IP位址指派給服務。在內部DNS伺服器組態中、您應該將Astra Control Center所選的DNS名稱指向負載平衡的IP位址。

  ⓘ 如需有關「負載平衡器」和入口服務類型的詳細資訊、請參閱 "需求"。

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  astraKubeConfigSecret: "acc-kubeconfig-cred or custom secret name"
  ingressType: "Generic"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
```

完整的**Astra**控制中心和操作員安裝

1. 如果您尚未在上一步中執行此操作、請建立「NetApp-acc」（或自訂）命名空間：

```
kubectl create ns [netapp-acc or custom namespace]
```

回應範例：

```
namespace/netapp-acc created
```

2. 在「NetApp-acc」（或您的自訂）命名空間中安裝Astra Control Center：

```
kubectl apply -f astra_control_center_min.yaml -n [netapp-acc or custom
namespace]
```

回應範例：

```
astracontrolcenter.astra.netapp.io/astra created
```

驗證系統狀態

> 如果您偏好使用OpenShift、您可以使用相似的相關命令來進行驗證步驟。

1. 驗證是否已成功安裝所有系統元件。

```
kubectl get pods -n [netapp-acc or custom namespace]
```

每個Pod的狀態應為「執行中」。部署系統Pod可能需要幾分鐘的時間。

回應範例：

```
NAME                                        READY   STATUS    RESTARTS
AGE
acc-helm-repo-5f75c5f564-bzqmt              1/1     Running   0
11m
activity-6b8f7cccb9-mlrn4                   1/1     Running   0
9m2s
api-token-authentication-6hznt              1/1     Running   0
8m50s
api-token-authentication-qpfgb              1/1     Running   0
8m50s
```

```
api-token-authentication-sqnb7             1/1     Running   0
8m50s
asup-5578bbdd57-dxkbp                      1/1     Running   0
9m3s
authentication-56bff4f95d-mspmq            1/1     Running   0
7m31s
bucketservice-6f7968b95d-9rrrl             1/1     Running   0
8m36s
cert-manager-5f6cf4bc4b-82khn              1/1     Running   0
6m19s
cert-manager-cainjector-76cf976458-sdrbc   1/1     Running   0
6m19s
cert-manager-webhook-5b7896bfd8-2n45j      1/1     Running   0
6m19s
cloud-extension-749d9f684c-8bdhq           1/1     Running   0
9m6s
cloud-insights-service-7d58687d9-h5tzw     1/1     Running   2
8m56s
composite-compute-968c79cb5-nv7l4          1/1     Running   0
9m11s
composite-volume-7687569985-jg9gg          1/1     Running   0
8m33s
credentials-5c9b75f4d6-nx9cz               1/1     Running   0
8m42s
entitlement-6c96fd8b78-zt7f8               1/1     Running   0
8m28s
features-5f7bfc9f68-gsjnl                  1/1     Running   0
8m57s
fluent-bit-ds-h88p7                        1/1     Running   0
7m22s
fluent-bit-ds-krhnj                        1/1     Running   0
7m23s
fluent-bit-ds-l5bjj                        1/1     Running   0
7m22s
fluent-bit-ds-lrclb                        1/1     Running   0
7m23s
fluent-bit-ds-s5t4n                        1/1     Running   0
7m23s
fluent-bit-ds-zpr6v                        1/1     Running   0
7m22s
graphql-server-5f5976f4bd-vbb4z            1/1     Running   0
7m13s
identity-56f78b8f9f-8h9p9                  1/1     Running   0
8m29s
influxdb2-0                                1/1     Running   0
11m
```

```
krakend-6f8d995b4d-5khkl                    1/1     Running   0
7m7s
license-5b5db87c97-jmxzc                    1/1     Running   0
9m
login-ui-57b57c74b8-6xtv7                   1/1     Running   0
7m10s
loki-0                                      1/1     Running   0
11m
monitoring-operator-9dbc9c76d-8znck         2/2     Running   0
7m33s
nats-0                                      1/1     Running   0
11m
nats-1                                      1/1     Running   0
10m
nats-2                                      1/1     Running   0
10m
nautilus-6b9d88bc86-h8kfb                   1/1     Running   0
8m6s
nautilus-6b9d88bc86-vn68r                   1/1     Running   0
8m35s
openapi-b87d77dd8-5dz9h                     1/1     Running   0
9m7s
polaris-consul-consul-5ljfb                 1/1     Running   0
11m
polaris-consul-consul-s5d5z                 1/1     Running   0
11m
polaris-consul-consul-server-0              1/1     Running   0
11m
polaris-consul-consul-server-1              1/1     Running   0
11m
polaris-consul-consul-server-2              1/1     Running   0
11m
polaris-consul-consul-twmpq                 1/1     Running   0
11m
polaris-mongodb-0                           2/2     Running   0
11m
polaris-mongodb-1                           2/2     Running   0
10m
polaris-mongodb-2                           2/2     Running   0
10m
polaris-ui-84dc87847f-zrg8w                 1/1     Running   0
7m12s
polaris-vault-0                             1/1     Running   0
11m
polaris-vault-1                             1/1     Running   0
11m
```

```
polaris-vault-2                            1/1     Running   0
11m
public-metrics-657698b66f-67pgt            1/1     Running   0
8m47s
storage-backend-metrics-6848b9fd87-w7x8r   1/1     Running   0
8m39s
storage-provider-5ff5868cd5-r9hj7          1/1     Running   0
8m45s
telegraf-ds-dw4hg                          1/1     Running   0
7m23s
telegraf-ds-k92gn                          1/1     Running   0
7m23s
telegraf-ds-mmxjl                          1/1     Running   0
7m23s
telegraf-ds-nhs8s                          1/1     Running   0
7m23s
telegraf-ds-rj7lw                          1/1     Running   0
7m23s
telegraf-ds-tqrkb                          1/1     Running   0
7m23s
telegraf-rs-9mwgj                          1/1     Running   0
7m23s
telemetry-service-56c49d689b-ffrzx         1/1     Running   0
8m42s
tenancy-767c77fb9d-g9ctv                   1/1     Running   0
8m52s
traefik-5857d87f85-7pmx8                   1/1     Running   0
6m49s
traefik-5857d87f85-cpxgv                   1/1     Running   0
5m34s
traefik-5857d87f85-lvmlb                   1/1     Running   0
4m33s
traefik-5857d87f85-t2xlk                   1/1     Running   0
4m33s
traefik-5857d87f85-v9wpf                   1/1     Running   0
7m3s
trident-svc-595f84dd78-zb8l6               1/1     Running   0
8m54s
vault-controller-86c94fbf4f-krttq          1/1     Running   0
9m24s
```

2. （選用）若要確保安裝完成、您可以使用下列命令來查看「acc operator」記錄。

```
kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-
operator -c manager -f
```

> 「accHost」叢集登錄是最後一項作業、如果失敗、就不會導致部署失敗。如果記錄中指出叢集登錄失敗、您可以透過新增叢集工作流程再次嘗試登錄 "在UI中" 或API。

3. 當所有Pod都在執行時、請擷取由Astra Control Center營運者安裝的「適用鍵盤」執行個體、以驗證安裝是否成功。

```
kubectl get acc -o yaml -n [netapp-acc or custom namespace]
```

4. 在Yaml中、勾選回應中的「tatus.deploymentState`」欄位、以取得「部署」值。如果部署失敗、則會改為顯示錯誤訊息。
5. 若要取得登入Astra Control Center時使用的一次性密碼、請複製「stats.uuid」值。密碼為「ACC-」、後面接著UUID值（「ACC-[UUUID]」、或是在本範例中為「ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f」）。

Y反 洗錢詳細資料範例

```
name: astra
   namespace: netapp-acc
   resourceVersion: "104424560"
   selfLink: /apis/astra.netapp.io/v1/namespaces/netapp-
acc/astracontrolcenters/astra
   uid: 9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f
 spec:
   accountName: Example
   astraAddress: astra.example.com
   astraVersion: 21.12.60
   autoSupport:
     enrolled: true
     url: https://support.netapp.com/asupprod/post/1.0/postAsup
   crds: {}
   email: admin@example.com
   firstName: SRE
   imageRegistry:
     name: registry_name/astra
     secret: astra-registry-cred
   lastName: Admin
 status:
   accConditionHistory:
     items:
     - astraVersion: 21.12.60
       condition:
         lastTransitionTime: "2021-11-23T02:23:59Z"
         message: Deploying is currently in progress.
         reason: InProgress
         status: "False"
         type: Ready
       generation: 2
       observedSpec:
         accountName: Example
         astraAddress: astra.example.com
         astraVersion: 21.12.60
         autoSupport:
           enrolled: true
           url: https://support.netapp.com/asupprod/post/1.0/postAsup
         crds: {}
         email: admin@example.com
         firstName: SRE
         imageRegistry:
           name: registry_name/astra
```

```
          secret: astra-registry-cred
        lastName: Admin
      timestamp: "2021-11-23T02:23:59Z"
    - astraVersion: 21.12.60
      condition:
        lastTransitionTime: "2021-11-23T02:23:59Z"
        message: Deploying is currently in progress.
        reason: InProgress
        status: "True"
        type: Deploying
      generation: 2
      observedSpec:
        accountName: Example
        astraAddress: astra.example.com
        astraVersion: 21.12.60
        autoSupport:
          enrolled: true
          url: https://support.netapp.com/asupprod/post/1.0/postAsup
        crds: {}
        email: admin@example.com
        firstName: SRE
        imageRegistry:
          name: registry_name/astra
          secret: astra-registry-cred
        lastName: Admin
      timestamp: "2021-11-23T02:23:59Z"
    - astraVersion: 21.12.60
      condition:
        lastTransitionTime: "2021-11-23T02:29:41Z"
        message: Post Install was successful
        observedGeneration: 2
        reason: Complete
        status: "True"
        type: PostInstallComplete
      generation: 2
      observedSpec:
        accountName: Example
        astraAddress: astra.example.com
        astraVersion: 21.12.60
        autoSupport:
          enrolled: true
          url: https://support.netapp.com/asupprod/post/1.0/postAsup
        crds: {}
        email: admin@example.com
        firstName: SRE
        imageRegistry:
```

```
          name: registry_name/astra
          secret: astra-registry-cred
        lastName: Admin
      timestamp: "2021-11-23T02:29:41Z"
    - astraVersion: 21.12.60
      condition:
        lastTransitionTime: "2021-11-23T02:29:41Z"
        message: Deploying succeeded.
        reason: Complete
        status: "False"
        type: Deploying
      generation: 2
      observedGeneration: 2
      observedSpec:
        accountName: Example
        astraAddress: astra.example.com
        astraVersion: 21.12.60
        autoSupport:
          enrolled: true
          url: https://support.netapp.com/asupprod/post/1.0/postAsup
        crds: {}
        email: admin@example.com
        firstName: SRE
        imageRegistry:
          name: registry_name/astra
          secret: astra-registry-cred
        lastName: Admin
      observedVersion: 21.12.60
      timestamp: "2021-11-23T02:29:41Z"
    - astraVersion: 21.12.60
      condition:
        lastTransitionTime: "2021-11-23T02:29:41Z"
        message: Astra is deployed
        reason: Complete
        status: "True"
        type: Deployed
      generation: 2
      observedGeneration: 2
      observedSpec:
        accountName: Example
        astraAddress: astra.example.com
        astraVersion: 21.12.60
        autoSupport:
          enrolled: true
          url: https://support.netapp.com/asupprod/post/1.0/postAsup
        crds: {}
```

```
        email: admin@example.com
        firstName: SRE
        imageRegistry:
          name: registry_name/astra
          secret: astra-registry-cred
        lastName: Admin
      observedVersion: 21.12.60
      timestamp: "2021-11-23T02:29:41Z"
    - astraVersion: 21.12.60
      condition:
        lastTransitionTime: "2021-11-23T02:29:41Z"
        message: Astra is deployed
        reason: Complete
        status: "True"
        type: Ready
      generation: 2
      observedGeneration: 2
      observedSpec:
        accountName: Example
        astraAddress: astra.example.com
        astraVersion: 21.12.60
        autoSupport:
          enrolled: true
          url: https://support.netapp.com/asupprod/post/1.0/postAsup
        crds: {}
        email: admin@example.com
        firstName: SRE
        imageRegistry:
          name: registry_name/astra
          secret: astra-registry-cred
        lastName: Admin
      observedVersion: 21.12.60
      timestamp: "2021-11-23T02:29:41Z"
  certManager: deploy
  cluster:
    type: OCP
    vendorVersion: 4.7.5
    version: v1.20.0+bafe72f
  conditions:
  - lastTransitionTime: "2021-12-08T16:19:55Z"
    message: Astra is deployed
    reason: Complete
    status: "True"
    type: Ready
  - lastTransitionTime: "2021-12-08T16:19:55Z"
    message: Deploying succeeded.
```

```
      reason: Complete
      status: "False"
      type: Deploying
    - lastTransitionTime: "2021-12-08T16:19:53Z"
      message: Post Install was successful
      observedGeneration: 2
      reason: Complete
      status: "True"
      type: PostInstallComplete
    - lastTransitionTime: "2021-12-08T16:19:55Z"
      message: Astra is deployed
      reason: Complete
      status: "True"
      type: Deployed
    deploymentState: Deployed
    observedGeneration: 2
    observedSpec:
      accountName: Example
      astraAddress: astra.example.com
      astraVersion: 21.12.60
      autoSupport:
        enrolled: true
        url: https://support.netapp.com/asupprod/post/1.0/postAsup
      crds: {}
      email: admin@example.com
      firstName: SRE
      imageRegistry:
        name: registry_name/astra
        secret: astra-registry-cred
      lastName: Admin
    observedVersion: 21.12.60
    postInstall: Complete
    uuid: 9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f
  kind: List
  metadata:
   resourceVersion: ""
   selfLink: ""
```

設定入口以進行負載平衡

您可以設定Kubernetes入口控制器來管理外部服務存取、例如叢集中的負載平衡。

本程序說明如何設定入口控制器（「擷取類型：一般」）。這是Astra Control Center的預設動作。部署Astra Control Center之後、您需要設定入口控制器、以URL顯示Astra Control Center。

> 如果您不想設定入口控制器、可以設定「擷取類型：AccTraefik」。Astra Control Center使用類型為「loadbalancer」（Astra Control Center命名空間中的「shvC/truefik」）的服務、並要求指派可存取的外部IP位址。如果您的環境允許負載平衡器、但您尚未設定負載平衡器、則可以使用MetalLB或其他外部服務負載平衡器、將外部IP位址指派給服務。在內部DNS伺服器組態中、您應該將Astra Control Center所選的DNS名稱指向負載平衡的IP位址。如需有關「負載平衡器」和入口服務類型的詳細資訊、請參閱 "需求"。

這些步驟會因您使用的入口控制器類型而有所不同：

- Nginx入口控制器
- OpenShift入口控制器

您需要的產品

- 必要的 "入口控制器" 應已部署。
- 。 "入口等級" 應已建立對應於入口控制器的。
- 您使用的Kubernetes版本介於v1.19和v1.22之間、甚至包括在內。

適用於**Nginvin**像 控制器的步驟

1. 建立類型的秘密 "8a637503539b25b68130b6e8003579d9" 如所述、在「NetApp-acc」（或自訂命名）命名空間中取得TLS私密金鑰和憑證 "TLS機密"。
2. 使用「v1beta1」（Kubernetes版本低於或1.22）或「v1」資源類型、部署「NetApp-acc」（或自訂命名）命名空間中的入口資源、以取代已過時或新架構：
   a. 對於已過時的「v1Beta1」架構、請遵循以下範例：

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-acc
  namespace: [netapp-acc or custom namespace]
  annotations:
    kubernetes.io/ingress.class: [class name for nginx controller]
spec:
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: [ACC address]
    http:
      paths:
      - backend:
        serviceName: traefik
        servicePort: 80
        pathType: ImplementationSpecific
```

b. 如需「v1」新架構、請遵循以下範例：

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: netapp-acc-ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: [class name for nginx controller]
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: <ACC addess>
    http:
      paths:
        - path:
          backend:
            service:
              name: traefik
              port:
                number: 80
          pathType: ImplementationSpecific
```

OpenShift入口控制器的步驟

1. 取得您的憑證、取得可供OpenShift路由使用的金鑰、憑證和CA檔案。
2. 建立OpenShift路由：

```
oc create route edge --service=traefik
--port=web -n [netapp-acc or custom namespace]
--insecure-policy=Redirect --hostname=<ACC address>
--cert=cert.pem --key=key.pem
```

登入**Astra Control Center UI**

安裝Astra Control Center之後、您將變更預設管理員的密碼、並登入Astra Control Center UI儀表板。

步驟

1. 在瀏覽器中、輸入您在「Astra Address」（astrAddress）中使用的FQDN、位於「Astra控制中心_min.yaml」（當）字段中 您安裝了Astra Control Center。
2. 出現提示時、請接受自我簽署的憑證。

> 您可以在登入後建立自訂憑證。

3. 在Astra Control Center登入頁面中、輸入您在「Astra Control Center_min.yaml」CR中使用的「電子郵件」值 您安裝了Astra Control Center，然後是一次性密碼（「ACC-[UUUID]」）。

> 如果您輸入錯誤密碼三次、系統將鎖定管理員帳戶15分鐘。

4. 選擇*登入*。
5. 出現提示時變更密碼。

> 如果這是您第一次登入、但您忘記密碼、而且尚未建立其他管理使用者帳戶、請聯絡NetApp支援部門以取得密碼恢復協助。

6. （選用）移除現有的自我簽署TLS憑證、並以取代 "由憑證授權單位（CA）簽署的自訂TLS憑證"。

疑難排解安裝

如果有任何服務處於「錯誤」狀態、您可以檢查記錄。尋找400到500範圍內的API回應代碼。這些都表示發生故障的地點。

步驟

1. 若要檢查Astra控制中心的操作員記錄、請輸入下列內容：

```
kubectl logs --follow -n netapp-acc-operator $(kubectl get pods -n
netapp-acc-operator -o name)  -c manager
```

下一步

執行以完成部署 "設定工作"。

使用OpenShift作業系統集線器安裝Astra Control Center

如果您使用Red Hat OpenShift、可以使用Red Hat認證的操作員來安裝Astra Control Center。請使用此程序從安裝Astra Control Center "Red Hat生態系統目錄" 或使用Red Hat OpenShift Container Platform。

完成此程序之後、您必須返回安裝程序、才能完成 "剩餘步驟" 以驗證安裝是否成功並登入。

您需要的產品

- "開始安裝之前、請先準備好環境以進行Astra Control Center部署"。
- 從OpenShift叢集、確保所有叢集操作員都處於正常狀態（「Available」（可用）為「true」（真））：

```
oc get clusteroperators
```

- 從OpenShift叢集、確保所有API服務都處於健全狀態（「Available」（可用）為「true」）：

```
oc get apiservices
```

- 您已在資料中心建立Astra Control Center的FQDN位址。
- 您有必要的權限和存取權、可以存取Red Hat OpenShift Container Platform來執行所述的安裝步驟。

步驟

- 下載並解壓縮Astra Control Center套裝組合
- 安裝NetApp Astra kubecl外掛程式
- [將映像新增至本機登錄]
- [尋找操作員安裝頁面]
- [安裝操作員]
- 安裝Astra Control Center

下載並解壓縮**Astra Control Center**套裝組合

1. 請從下載Astra Control Center套裝組合（「Astra控制中心-[版本].tar.gz」） "NetApp 支援網站"。
2. 從下載Astra Control Center認證與金鑰的壓縮檔 "NetApp 支援網站"。
3. （可選）使用以下命令驗證套件的簽名：

```
openssl dgst -sha256 -verify astra-control-center[version].pub
-signature <astra-control-center[version].sig astra-control-
center[version].tar.gz
```

4. 擷取影像：

```
tar -vxzf astra-control-center-[version].tar.gz
```

安裝**NetApp Astra kubecl**外掛程式

NetApp Astra「kubecll」命令列外掛程式可在執行與部署及升級Astra Control Center相關的一般工作時節省時間。

您需要的產品

NetApp為不同的CPU架構和作業系統提供外掛程式的二進位檔。執行此工作之前、您必須先瞭解您的CPU和作業系統。在Linux和Mac作業系統上、您可以使用「uname -A」命令來收集此資訊。

步驟

1. 列出可用的NetApp Astra「kubecll」外掛程式二進位檔、並記下作業系統和CPU架構所需的檔案名稱：

```
ls kubectl-astra/
```

2. 將檔案複製到與標準「kubecll」公用程式相同的位置。在此範例中、「kubecll」公用程式位於「usr/local/bin」目錄中。將「<二進位名稱>」取代為您所需的檔案名稱：

```
cp kubectl-astra/<binary-name> /usr/local/bin/kubectl-astra
```

將映像新增至本機登錄

1. 變更至Astra目錄：

```
cd acc
```

2. 將Astra Control Center映像目錄中的檔案新增至本機登錄。

> 請參閱以下自動載入影像的範例指令碼。

a. 登入您的登錄：

Docker：

```
docker login [your_registry_path]
```

Podcast：

```
podman login [your_registry_path]
```

b. 使用適當的指令碼來載入映像、標記映像、並[Subforte_image_local_register_pip]將映像推送到本機登錄：

Docker：

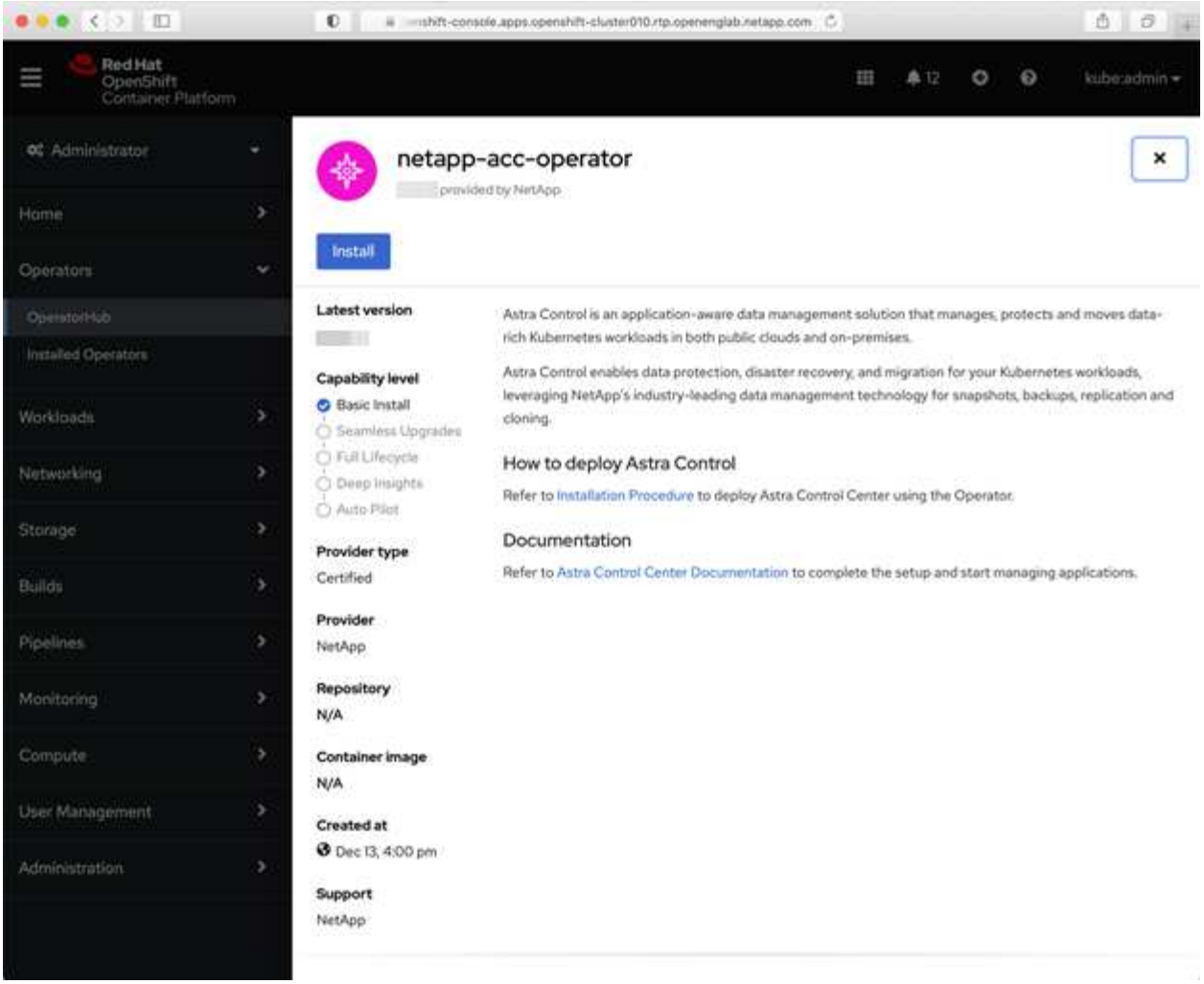
```
export REGISTRY=[Docker_registry_path]
for astraImageFile in $(ls images/*.tar) ; do
  # Load to local cache. And store the name of the loaded image
trimming the 'Loaded images: '
  astraImage=$(docker load --input ${astraImageFile} | sed 's/Loaded
image: //')
  astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
  # Tag with local image repo.
  docker tag ${astraImage} ${REGISTRY}/${astraImage}
  # Push to the local repo.
  docker push ${REGISTRY}/${astraImage}
done
```

Podcast：

```
export REGISTRY=[Registry_path]
for astraImageFile in $(ls images/*.tar) ; do
  # Load to local cache. And store the name of the loaded image trimming
the 'Loaded images: '
  astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image(s): //')
  astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
  # Tag with local image repo.
  podman tag ${astraImage} ${REGISTRY}/${astraImage}
  # Push to the local repo.
  podman push ${REGISTRY}/${astraImage}
done
```

尋找操作員安裝頁面

1. 請完成下列其中一個程序、以存取操作員安裝頁面：
   - 從Red Hat Openshift Web主控台



：

i. 登入OpenShift Container Platform UI。

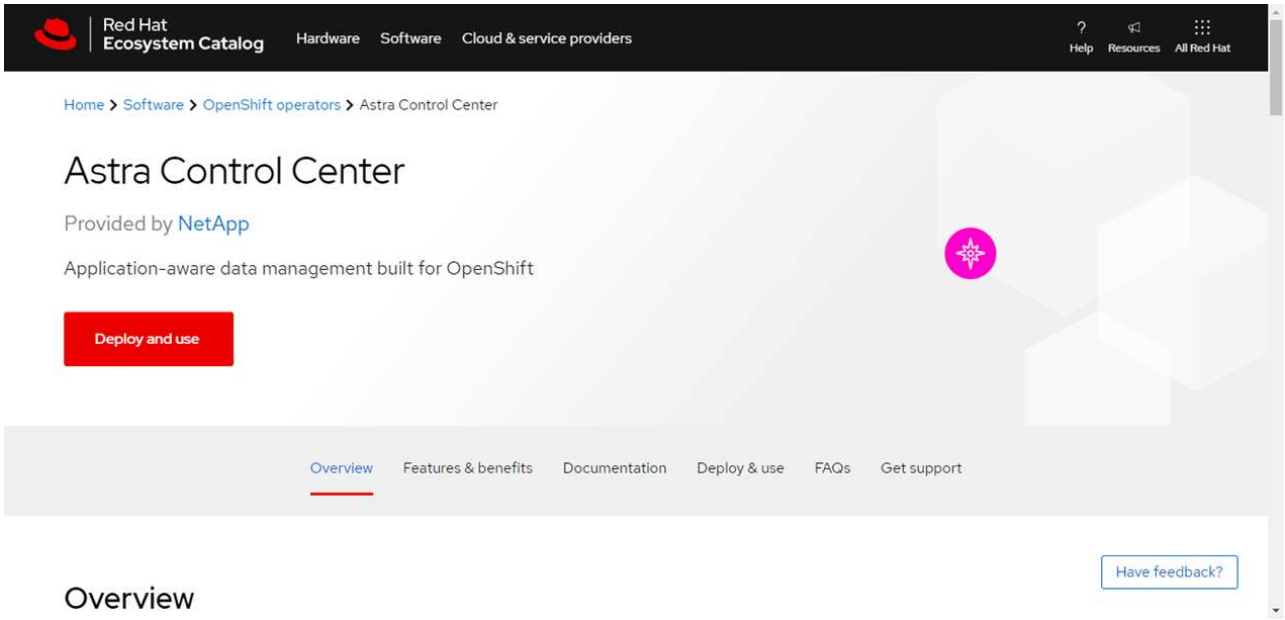
ii. 從側功能表中、選取*運算子>運算子中樞*。

iii. 選擇NetApp Astra Control Center營運者。

iv. 選擇*安裝*。

。從Red Hat生態系統目錄

：



i. 選擇NetApp Astra Control Center "營運者"。

ii. 選擇*部署和使用*。

安裝操作員

1. 完成*安裝操作員*頁面並安裝操作員：

> 此運算子可用於所有叢集命名空間。

a. 在操作員安裝過程中、系統會自動建立運算子命名空間或「NetApp-acc operator」命名空間。

b. 選取手動或自動核准策略。

> 建議手動核准。每個叢集只能執行單一運算子執行個體。

c. 選擇*安裝*。

> 如果您選擇手動核准策略、系統會提示您核准此操作員的手動安裝計畫。

2. 從主控台移至「作業系統集線器」功能表、確認操作員已成功安裝。

安裝**Astra Control Center**

1. 在Astra Control Center操作員的詳細資料檢視中、從主控台選取所提供API區段中的「Create instance（建立執行個體）」。

2. 填寫「Create適用的」表單欄位：

a. 保留或調整Astra Control Center名稱。

b. （選用）啟用或停用自動支援。建議保留「自動支援」功能。

c. 輸入Astra Control Center位址。請勿在地址中輸入「http：//」或「https：//」。

d. 輸入Astra Control Center版本、例如21.12.60。

e. 輸入帳戶名稱、電子郵件地址和管理員姓氏。
f. 保留預設的Volume回收原則。
g. 在*映像登錄*中、輸入您的本機容器映像登錄路徑。請勿在地址中輸入「http：//」或「https：//」。
h. 如果您使用需要驗證的登錄、請輸入密碼。
i. 輸入管理員名字。
j. 設定資源擴充。
k. 保留預設的儲存類別。
l. 定義客戶需求日處理偏好設定。

3. 選取「Create」（建立）。

下一步

確認Astra Control Center安裝成功、然後完成 "剩餘步驟" 以登入。此外、您也可以執行來完成部署 "設定工作"。

安裝Astra Control Center搭配Cloud Volumes ONTAP 一套功能性儲存後端

有了Astra Control Center、您就能在混合雲環境中使用自我管理的Kubernetes叢集和Cloud Volumes ONTAP 實例來管理應用程式。您可以在內部部署的Kubernetes叢集或雲端環境中的其中一個自我管理Kubernetes叢集上部署Astra Control Center。

有了其中一項部署、您就能使用Cloud Volumes ONTAP 下列其中一項部署、以下列方式執行應用程式資料管理作業：將NetApp當成儲存後端。您也可以將S3儲存區設定為備份目標。

若要在Amazon Web Services（AWS）和Microsoft Azure中安裝Astra Control Center搭配Cloud Volumes ONTAP 使用整套儲存後端、請視您的雲端環境而定、執行下列步驟。

- 在Amazon Web Services中部署Astra Control Center
- 在Microsoft Azure中部署Astra Control Center

在Amazon Web Services中部署Astra Control Center

您可以在Amazon Web Services（AWS）公有雲上的自我管理Kubernetes叢集上部署Astra Control Center。

部署Astra Control Center僅支援自我管理的OpenShift Container Platform（OCP）叢集。

AWS所需的功能

在AWS中部署Astra Control Center之前、您需要下列項目：

- Astra Control Center授權。請參閱 "Astra Control Center授權要求"。
- "符合Astra Control Center的要求"。
- NetApp Cloud Central帳戶
- Red Hat OpenShift Container Platform（OCP）權限（位於命名空間層級以建立Pod）
- AWS認證資料、存取ID和秘密金鑰、具備可讓您建立儲存區和連接器的權限

- AWS帳戶彈性容器登錄（ECR）存取與登入
- 存取Astra Control UI所需的AWS託管區域和Route 53項目

AWS的作業環境需求

Astra Control Center需要下列AWS作業環境：

- Red Hat OpenShift Container Platform 4.8.

> 確保您選擇裝載Astra Control Center的作業環境符合環境正式文件中所述的基本資源需求。

除了環境的資源需求之外、Astra Control Center還需要下列資源：

元件	需求
後端**NetApp Cloud Volumes ONTAP** 功能儲存容量	至少提供300 GB
工作者節點（**AWS EC2**需求）	總共至少3個工作節點、每個節點有4個vCPU核心和12GB RAM
負載平衡器	服務類型「負載平衡器」可用於將入口流量傳送至作業環境叢集中的服務
FQDN	將Astra Control Center的FQDN指向負載平衡IP位址的方法
Astra Trident（安裝於**NetApp Cloud Manager**的**Kubernetes**叢集探索中）	Astra Trident 21.004或更新版本已安裝並設定、且NetApp ONTAP 的版本9.5或更新為儲存後端
映像登錄	您必須擁有現有的私有登錄、例如AWS Elastic Container登錄、才能將Astra Control Center建置映像推入其中。您需要提供映像登錄的URL、以便上傳映像。 Astra Control Center託管叢集和託管叢集必須能夠存取相同的映像登錄、才能使用還原型映像來備份和還原應用程式。
Astra Trident / ONTAP Estra組態	Astra Control Center需要建立儲存類別、並將其設為預設儲存類別。Astra Control Center支援下列ONTAP 將Kubernetes叢集匯入NetApp Cloud Manager時所建立的支援功能。這些資料由Astra Trident提供： • 「vsaworkingworking環境-<←ha-NAS csi.trident.netapp.io`」 • 「vsaworkingworking環境-<←ha-san csi.trident.netapp.io`」 • 「vsaworkingworking環境-<<>-se-NAS csi.trident.netapp.io`」 • 「vsaworkingworking環境-<←se-san csi.trident.netapp.io`」

> 這些需求假設Astra Control Center是營運環境中唯一執行的應用程式。如果環境正在執行其他應用程式、請相應調整這些最低需求。

> AWS登錄權杖會在12小時內過期、之後您必須更新Docker映像登錄機密。

AWS部署總覽

以下是安裝Astra Control Center for AWS的程序總覽、Cloud Volumes ONTAP 其中包含以作為儲存後端的功能。

以下將詳細說明每個步驟。

1. 確保您擁有足夠的IAM權限。
2. 在AWS上安裝RedHat OpenShift叢集。
3. 設定AWS。
4. 設定NetApp Cloud Manager。
5. 安裝Astra Control Center。

確保您擁有足夠的IAM權限

確保您擁有足夠的IAM角色和權限、可讓您安裝RedHat OpenShift叢集和NetApp Cloud Manager Connector。

請參閱 "初始 AWS 認證資料"。

在AWS上安裝RedHat OpenShift叢集

在AWS上安裝RedHat OpenShift Container Platform叢集。

如需安裝指示、請參閱 "在OpenShift Container Platform的AWS上安裝叢集"。

設定AWS

接下來、設定AWS以建立虛擬網路、設定EC2運算執行個體、建立AWS S3儲存區、建立彈性容器登錄（ECR）以裝載Astra Control Center映像、然後將映像推送至此登錄。

請遵循AWS文件完成下列步驟。請參閱 "AWS安裝文件"。

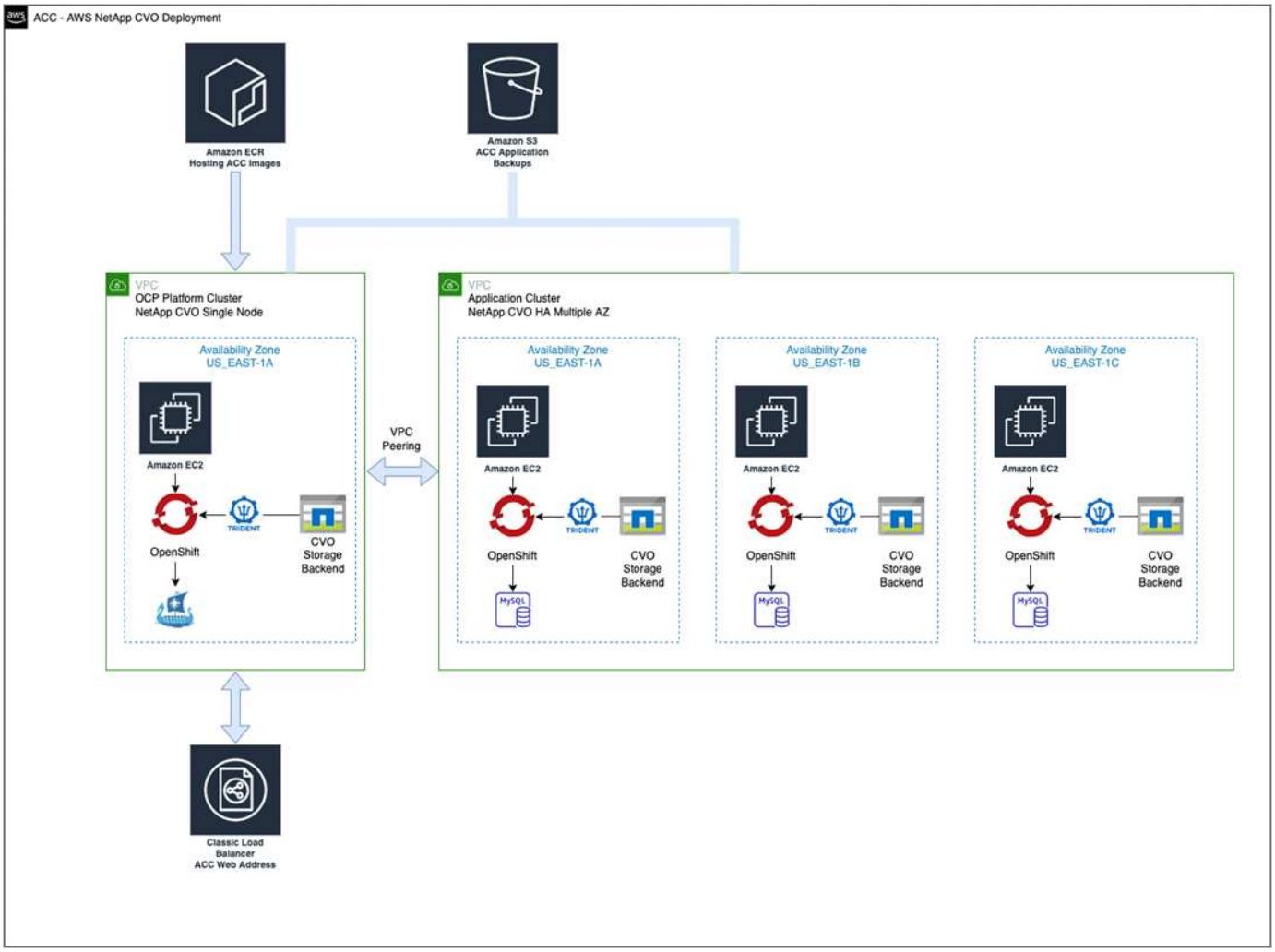
1. 建立AWS虛擬網路。
2. 檢閱EC2運算執行個體。這可以是AWS中的裸機伺服器或VM。
3. 如果執行個體類型尚未符合主節點和工作節點的Astra最低資源需求、請在AWS中變更執行個體類型以符合Astra需求。 請參閱 "Astra Control Center需求"。
4. 建立至少一個AWS S3儲存區來儲存備份。
5. 建立AWS彈性Container登錄（ECR）、以裝載所有的主動定速控制系統映像。

   > 如果您未建立ECR、Astra Control Center將無法從含有Cloud Volumes ONTAP AWS後端的支援的叢集存取監控資料。此問題是因為您嘗試使用Astra Control Center探索及管理的叢集無法存取AWS ECR。

6. 將Acc映像推送到您定義的登錄。

> AWS Elastic Container登錄（ECR）權杖會在12小時後過期、導致跨叢集複製作業失敗。從Cloud Volumes ONTAP 針對AWS設定的功能進行的功能區管理儲存後端時、就會發生此問題。若要修正此問題、請再次向ECR驗證、並產生新的秘密、讓複製作業順利恢復。

以下是AWS部署範例：



設定**NetApp Cloud Manager**

使用Cloud Manager建立工作區、將連接器新增至AWS、建立工作環境、以及匯入叢集。

請依照Cloud Manager文件完成下列步驟。請參閱下列內容：

- "開始使用Cloud Volumes ONTAP AWS的功能"。
- "使用Cloud Manager在AWS中建立連接器"

步驟

1. 將您的認證資料新增至Cloud Manager。
2. 建立工作區。
3. 新增AWS的連接器。選擇AWS做為供應商。
4. 為您的雲端環境建立工作環境。

a. 位置：「Amazon Web Services（AWS）」

b. 類型：Cloud Volumes ONTAP 「EHA」

5. 匯入OpenShift叢集。叢集將連線至您剛建立的工作環境。

   a. 選擇* K8s*>*叢集清單*>*叢集詳細資料*、即可檢視NetApp叢集詳細資料。

   b. 請注意右上角的Trident版本。

   c. 請注意Cloud Volumes ONTAP 、顯示NetApp為資源配置程式的叢集儲存類別。

      這會匯入您的Red Hat OpenShift叢集、並將其指派為預設儲存類別。您可以選取儲存類別。Trident會在匯入和探索程序中自動安裝。

6. 請注意此Cloud Volumes ONTAP 功能部署中的所有持續磁碟區和磁碟區。

> 可作為單一節點或高可用度運作。Cloud Volumes ONTAP如果已啟用HA、請記下在AWS中執行的HA狀態和節點部署狀態。

安裝Astra Control Center

遵循標準 "Astra Control Center安裝說明"。

在Microsoft Azure中部署Astra Control Center

您可以將Astra Control Center部署在Microsoft Azure公有雲上的自我管理Kubernetes叢集上。

Azure的必備功能

在Azure中部署Astra Control Center之前、您需要下列項目：

- Astra Control Center授權。請參閱 "Astra Control Center授權要求"。
- "符合Astra Control Center的要求"。
- NetApp Cloud Central帳戶
- Red Hat OpenShift Container Platform（OCP）4.8.
- Red Hat OpenShift Container Platform（OCP）權限（位於命名空間層級以建立Pod）
- Azure認證、具備可讓您建立儲存區和連接器的權限

Azure的營運環境需求

確保您選擇裝載Astra Control Center的作業環境符合環境正式文件中所述的基本資源需求。

除了環境的資源需求之外、Astra Control Center還需要下列資源：

請參閱 "Astra Control Center營運環境需求"。

元件	需求
後端**NetApp Cloud Volumes ONTAP** 功能儲存容量	至少提供300 GB

元件	需求
工作者節點（**Azure**運算需求）	總共至少3個工作節點、每個節點有4個vCPU核心和12GB RAM
負載平衡器	服務類型「負載平衡器」可用於將入口流量傳送至作業環境叢集中的服務
FQDN（**Azure DNS**區域）	將Astra Control Center的FQDN指向負載平衡IP位址的方法
Astra Trident（安裝於**NetApp Cloud Manager**的**Kubernetes**叢集探索中）	Astra Trident 21.004或更新版本已安裝並設定、NetApp ONTAP 版本9.5或更新版本將作為儲存後端使用
映像登錄	您必須擁有現有的私有登錄、例如Azure Container登錄（ACR）、才能將Astra Control Center建置映像推送至該登錄。您需要提供映像登錄的URL、以便上傳映像。 您必須啟用匿名存取、才能拉出還原映像進行備份。
Astra Trident / ONTAP Estra組態	Astra Control Center需要建立儲存類別、並將其設為預設儲存類別。Astra Control Center支援下列ONTAP 將Kubernetes叢集匯入NetApp Cloud Manager時所建立的支援功能。這些資料由Astra Trident提供： • 「vsaworkingworking環境-<←ha-NAS csi.trident.netapp.io`」 • 「vsaworkingworking環境-<←ha-san csi.trident.netapp.io`」 • 「vsaworkingworking環境-<<>-se-NAS csi.trident.netapp.io`」 • 「vsaworkingworking環境-<←se-san csi.trident.netapp.io`」

這些需求假設Astra Control Center是營運環境中唯一執行的應用程式。如果環境正在執行其他應用程式、請相應調整這些最低需求。

Azure部署總覽

以下是安裝Astra Control Center for Azure的程序總覽。

以下將詳細說明每個步驟。

1. 在Azure上安裝RedHat OpenShift叢集。
2. 建立Azure資源群組。
3. 確保您擁有足夠的IAM權限。
4. 設定Azure。
5. 設定NetApp Cloud Manager。
6. 安裝及設定Astra Control Center。

在Azure上安裝RedHat OpenShift叢集

第一步是在Azure上安裝RedHat OpenShift叢集。

如需安裝說明、請參閱上的 RedHat 說明文件 "在 Azure 上安裝 OpenShift 叢集" 和 "安裝 Azure 帳戶"。

建立Azure資源群組

建立至少一個Azure資源群組。

> OpenShift可能會建立自己的資源群組。此外、您也應該定義Azure資源群組。請參閱OpenShift文件。

您可能想要建立平台叢集資源群組和目標應用程式OpenShift叢集資源群組。

確保您擁有足夠的IAM權限

確保您擁有足夠的IAM角色和權限、可讓您安裝RedHat OpenShift叢集和NetApp Cloud Manager Connector。

請參閱 "Azure 認證與權限"。

設定Azure

接下來、設定Azure以建立虛擬網路、設定運算執行個體、建立Azure Blob容器、建立Azure Container Register（ACR）來裝載Astra Control Center映像、然後將映像推送至此登錄。

請依照Azure文件完成下列步驟。請參閱 "在Azure上安裝OpenShift叢集"。

1. 建立Azure虛擬網路。
2. 檢閱運算執行個體。這可以是Azure中的裸機伺服器或VM。
3. 如果執行個體類型尚未符合主節點和工作節點的Astra最低資源需求、請變更Azure中的執行個體類型以符合Astra要求。 請參閱 "Astra Control Center需求"。
4. 建立至少一個Azure Blob容器來儲存備份。
5. 建立儲存帳戶。您需要儲存帳戶來建立容器、以便在Astra Control Center中作為儲存庫。
6. 建立儲存貯體存取所需的機密。
7. 建立Azure Container登錄（ACR）、以裝載所有Astra Control Center映像。
8. 設定Docker推/拉所有Astra Control Center影像的ACR存取。
9. 輸入下列指令碼、將Acc映像推入此登錄：

   ```
   az acr login -n <AZ ACR URL/Location>
   This script requires ACC manifest file and your Azure ACR location.
   ```

   範例：

```
manifestfile=astra-control-center-<version>.manifest
AZ_ACR_REGISTRY=<target image repository>
ASTRA_REGISTRY=<source ACC image repository>

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRYY/$image
    docker push $AZ_ACR_REGISTRY/$image
done < astra-control-center-22.04.41.manifest
```

10. 設定DNS區域。

設定NetApp Cloud Manager

使用Cloud Manager建立工作區、將連接器新增至Azure、建立工作環境、以及匯入叢集。

請依照Cloud Manager文件完成下列步驟。請參閱 "Azure中的Cloud Manager入門"。

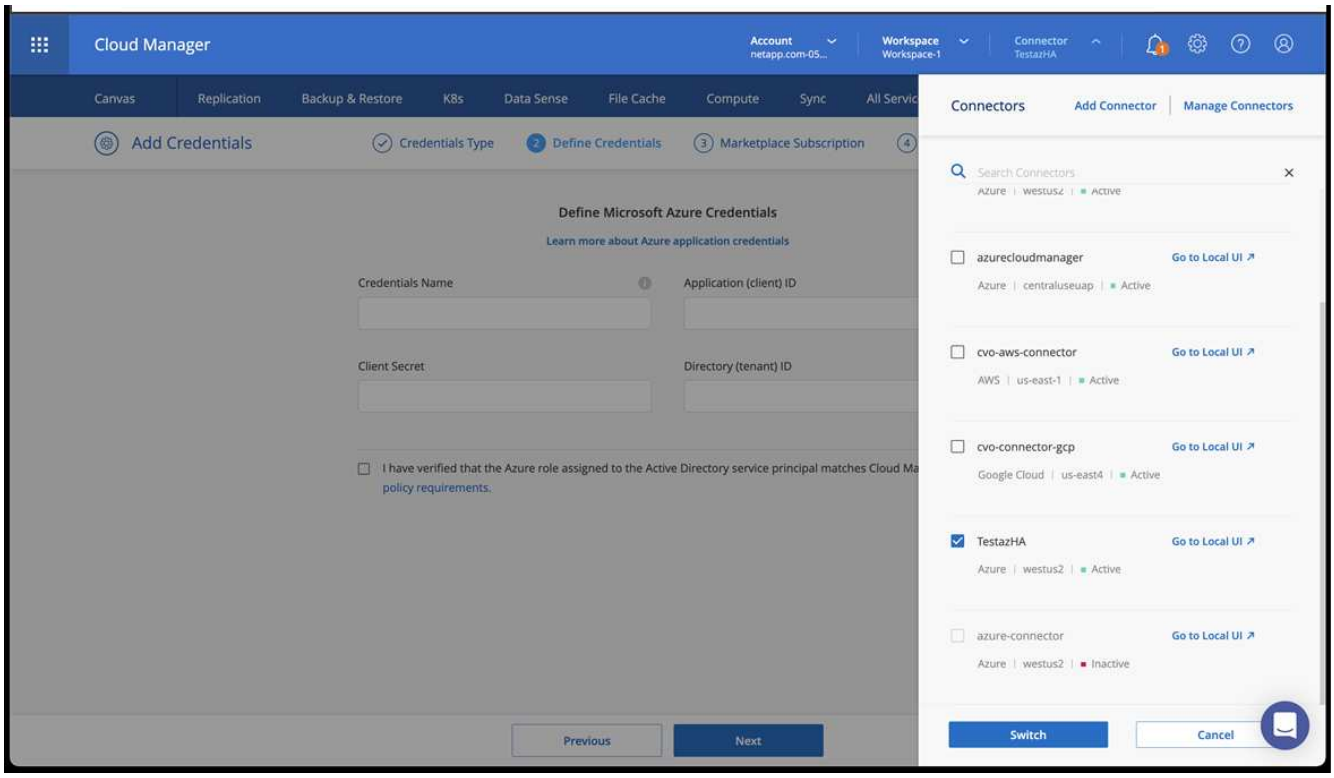
您需要的產品

以所需的IAM權限和角色存取Azure帳戶

步驟

1. 將您的認證資料新增至Cloud Manager。
2. 新增Azure連接器。請參閱 "Cloud Manager原則"。
   a. 選擇* Azure *作為供應商。
   b. 輸入Azure認證資料、包括應用程式ID、用戶端機密和目錄（租戶）ID。

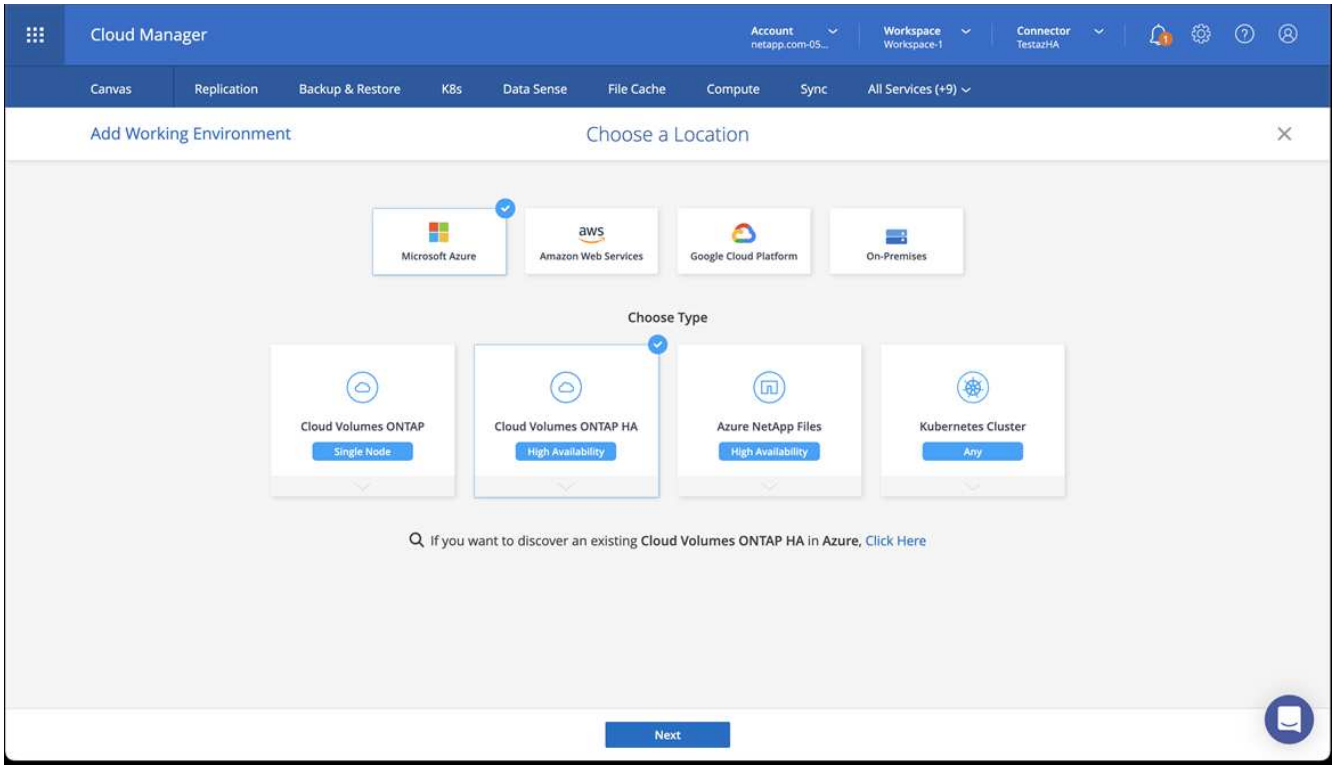
      請參閱 "從Cloud Manager在Azure中建立連接器"。
3. 確認連接器正在執行、並切換至該連接器。



4. 為您的雲端環境建立工作環境。

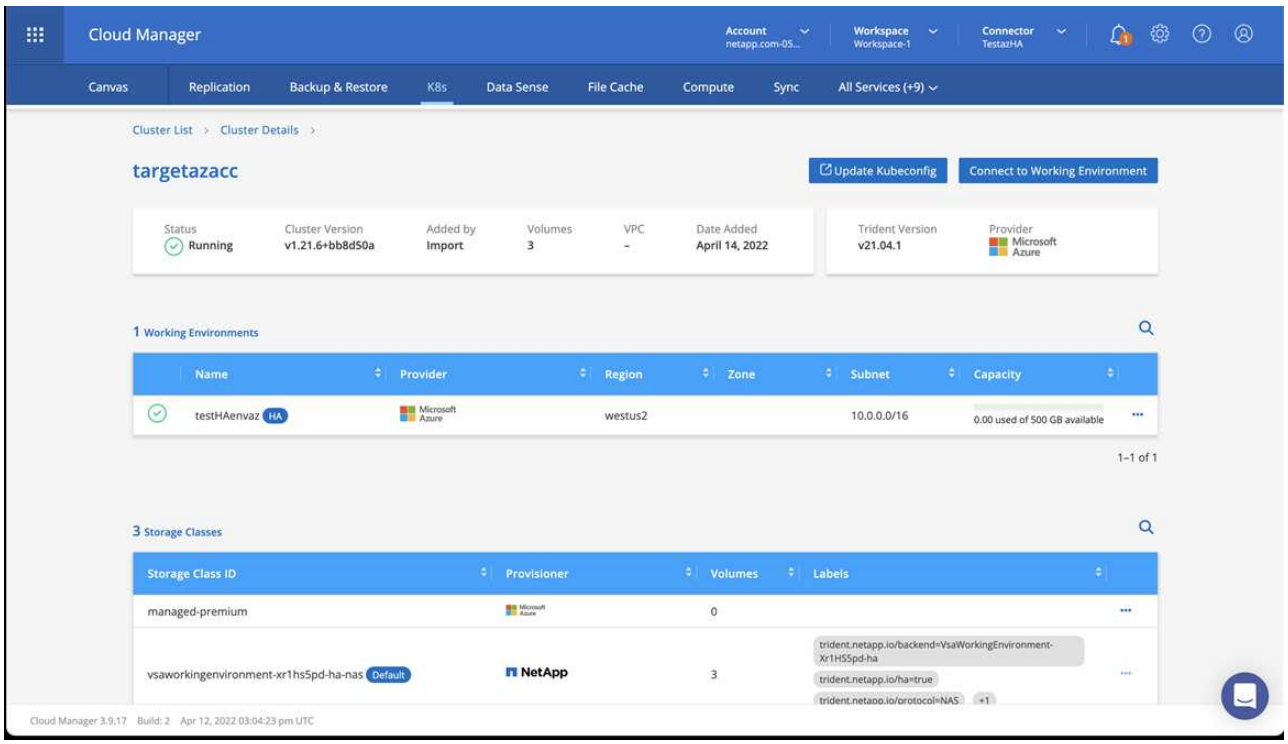
   a. 位置：「Microsoft Azure」。

   b. 輸入：Cloud Volumes ONTAP 「EHA」。



5. 匯入OpenShift叢集。叢集將連線至您剛建立的工作環境。

a. 選擇* K8s*>*叢集清單*>*叢集詳細資料*、即可檢視NetApp叢集詳細資料。



b. 請注意右上角的Trident版本。

c. 請注意Cloud Volumes ONTAP 、顯示NetApp為資源配置程式的叢集儲存類別。

這會匯入您的Red Hat OpenShift叢集、並指派預設的儲存類別。您可以選取儲存類別。Trident會在匯入和探索程序中自動安裝。

6. 請注意此Cloud Volumes ONTAP 功能部署中的所有持續磁碟區和磁碟區。
7. 可作為單一節點或高可用度運作。Cloud Volumes ONTAP如果已啟用HA、請記下Azure中執行的HA狀態和節點部署狀態。

安裝及設定**Astra Control Center**

使用標準安裝Astra Control Center "安裝說明"。

使用Astra Control Center新增Azure儲存庫。請參閱 "設定Astra Control Center並新增鏟斗"。

設定Astra控制中心

Astra Control Center支援ONTAP 並監控將支援及Astra Data Store做為儲存後端。安裝Astra Control Center、登入UI並變更密碼之後、您將需要設定授權、新增叢集、管理儲存設備及新增儲存區。

工作

- 新增Astra Control Center授權
- [新增叢集]
- [新增儲存後端]

- [新增儲存庫]

新增Astra Control Center授權

您可以使用UI或新增授權 "API" 以獲得完整的Astra控制中心功能。若無授權、您使用Astra Control Center的使用僅限於管理使用者及新增叢集。

如需如何計算授權的詳細資訊、請參閱 "授權"。

> 若要更新現有的評估或完整授權、請參閱 "更新現有授權"。

Astra Control Center授權會使用Kubernetes CPU單元來測量CPU資源。授權必須考量指派給所有受管理Kubernetes叢集之工作節點的CPU資源。新增授權之前、您必須先從取得授權檔案（NLF） "NetApp 支援網站"。

您也可以試用Astra Control Center搭配評估授權、從下載授權之日起90天內使用Astra Control Center。您可以註冊以免費試用 "請按這裡"。

> 如果您的安裝量成長到超過授權的CPU單元數量、Astra Control Center會防止您管理新的應用程式。超過容量時會顯示警示。

您需要的產品

當您從下載Astra Control Center時 "NetApp 支援網站"下載NetApp授權檔案（NLF）。請確定您有權存取此授權檔案。

步驟

1. 登入Astra Control Center UI。
2. 選擇*帳戶*>*授權*。
3. 選擇*新增授權*。
4. 瀏覽至您下載的授權檔案（NLF）。
5. 選擇*新增授權*。

「帳戶>*授權*」頁面會顯示授權資訊、到期日、授權序號、帳戶ID及使用的CPU單位。

> 如果您擁有評估授權、請務必儲存您的帳戶ID、以免在Astra Control Center故障時發生資料遺失（如果您未傳送ASUP）。

新增叢集

若要開始管理應用程式、請新增Kubernetes叢集、並將其當作運算資源來管理。您必須為Astra Control Center新增叢集、才能探索Kubernetes應用程式。對於Astra Data Store、您想要新增Kubernetes應用程式叢集、其中包含使用Astra Data Store所配置磁碟區的應用程式。

> 我們建議Astra Control Center先管理部署於上的叢集、再將其他叢集新增至Astra Control Center進行管理。需要管理初始叢集、才能傳送Kubmetrics資料和叢集相關資料、以供進行度量和疑難排解。您可以使用*新增叢集*功能、以Astra控制中心來管理叢集。

當Astra Control管理叢集時、它會追蹤叢集的預設儲存類別。如果您使用「kubecll」命令變更儲存類別、Astra Control會還原變更。若要變更由Astra Control管理之叢集中的預設儲存類別、請使用下列其中一種方法：

- 使用Astra Control API「PUT /managedClusters'端點」、並使用「DefaultStorageClass」參數指派不同的預設儲存類別。
- 使用Astra Control Web UI指派不同的預設儲存類別。請參閱 [變更預設儲存類別]。

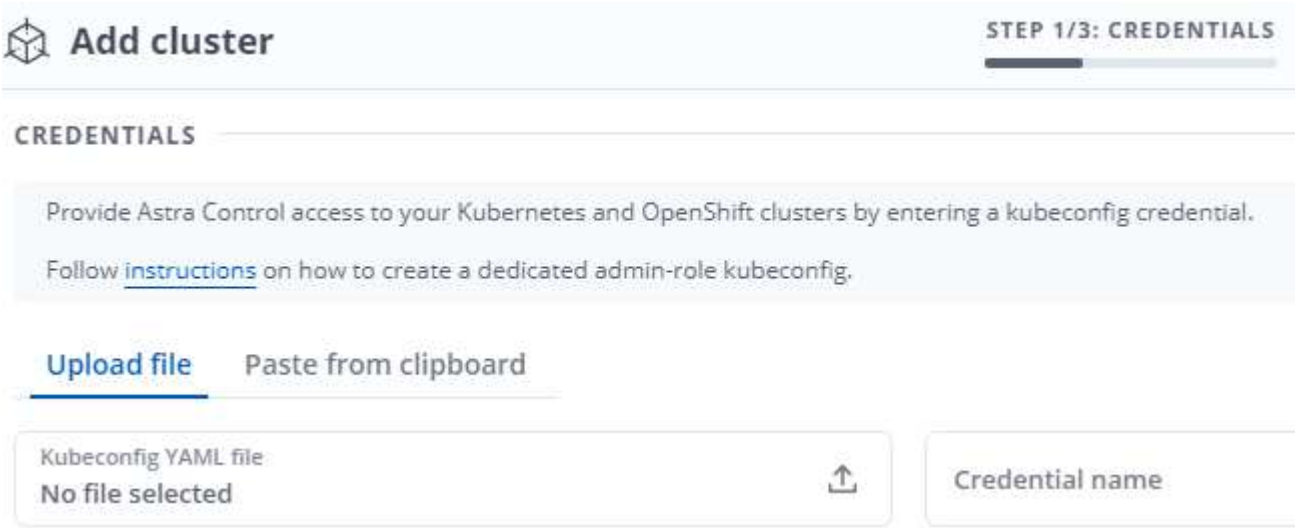
您需要的產品

- 新增叢集之前、請先檢閱並執行必要的 "必要工作"。

步驟

1. 在Astra Control Center UI的* Dashboard 中、選取「叢集」區段中的「Add*」。
2. 在打開的* Add Cluster.yaml視窗中、上傳「kubeconfig．yaml」檔案、或貼上「kubeconfig．yaml」檔案的內容。

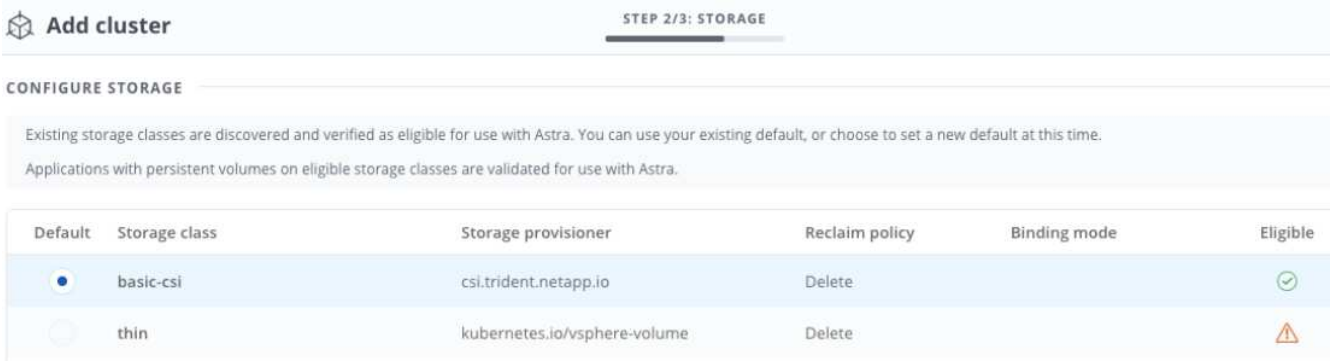
「kubeconfig．yaml」檔案應包含*一個叢集*的叢集認證資料。



如果您建立自己的「kubeconfig」檔案、您應該只定義其中*一個*內容元素。請參閱 "Kubernetes文件" 以取得建立「Kbeconfig」檔案的相關資訊。

3. 提供認證名稱。根據預設、認證名稱會自動填入為叢集名稱。
4. 選擇*設定儲存設備*。
5. 選取要用於此Kubernetes叢集的儲存類別、然後選取* Review *。

您應該選擇以ONTAP 不受資料儲存或Astra Data Store支援的Trident儲存類別。



6. 檢閱資訊、如果一切正常、請選取*新增叢集*。

結果

叢集會進入*探索*狀態、然後變更為*執行中*。您已成功新增Kubernetes叢集、現在正在Astra Control Center中進行管理。

> 在Astra Control Center中新增要管理的叢集之後、可能需要幾分鐘的時間來部署監控操作員。在此之前、通知圖示會變成紅色、並記錄*監控代理程式狀態檢查失敗*事件。您可以忽略這一點、因為當Astra Control Center取得正確狀態時、問題就能解決。如果幾分鐘內仍無法解決問題、請前往叢集、然後執行「ocGet pod -n NetApp-監 控」作為起點。您需要查看監控操作員記錄、以偵錯問題。

新增儲存後端

您可以新增儲存後端、以便Astra Control管理其資源。您可以在託管叢集上部署儲存後端、或使用現有的儲存後端。

將Astra Control中的儲存叢集管理為儲存後端、可讓您在持續磁碟區（PV）與儲存後端之間建立連結、以及取得額外的儲存指標。

現有**Astra Data Store**部署所需的資源

- 您已新增Kubernetes應用程式叢集和基礎運算叢集。

> 在您新增適用於Astra Data Store的Kubernetes應用程式叢集、並由Astra Control管理之後、叢集會在探索到的後端清單中顯示為「Unmanaged」。接下來、您必須新增包含Astra Data Store的運算叢集、並作為Kubernetes應用程式叢集的基礎。您可以從UI的*後端*執行此動作。選取叢集的「動作」功能表、選取「管理」、然後選取 "新增叢集"。在「Unmanaged」叢集狀態變更為Kubernetes叢集名稱之後、您可以繼續新增後端。

全新**Astra Data Store**部署所需的資源

- 您有 "已上傳您要部署的安裝套件版本" 至Astra Control可存取的位置。
- 您已新增要用於部署的Kubernetes叢集。
- 您已上傳 Astra Data Store授權 部署至Astra Control可存取的位置。

選項

- [部署儲存資源]
- [使用現有的儲存後端]

部署儲存資源

您可以部署新的Astra Data Store、並管理相關的儲存後端。

步驟

1. 從儀表板或後端功能表瀏覽：
   - 從*儀表板*：從「資源摘要」中、從「儲存後端」窗格中選取連結、然後從「後端」區段中選取「新增」。
   - 從*後端*：
     i. 在左側導覽區域中、選取*後端*。
     ii. 選取*「Add*」。
2. 在「部署」索引標籤中選取「* Astra Data Store*部署」選項。
3. 選取要部署的Astra Data Store套件：
   a. 輸入Astra Data Store應用程式的名稱。
   b. 選擇您要部署的Astra Data Store版本。

   > 如果您尚未上傳想要部署的版本、可以使用*新增套件*選項、或結束精靈並使用 "套件管理" 上傳安裝套裝組合。

4. 選取您先前已上傳的Astra Data Store授權、或使用*新增授權*選項上傳授權以搭配應用程式使用。

   > 具有完整權限的Astra Data Store授權會與Kubernetes叢集相關聯、而且這些相關的叢集應該會自動顯示。如果沒有託管叢集、您可以選取*新增叢集*選項、將其中一個新增至Astra Control管理。對於Astra Data Store授權、如果授權與叢集之間沒有關聯、您可以在精靈的下一頁定義此關聯。

5. 如果您尚未將Kubernetes叢集新增至Astra Control管理、則必須從* Kubernetes叢集*頁面執行此動作。從清單中選取現有的叢集、或選取*新增基礎叢集*、將叢集新增至Astra Control管理。
6. 選取Kubernetes叢集的部署範本大小、以提供Astra Data Store的資源。

   > 挑選範本時、請針對較大的工作負載選擇具有較多記憶體和核心的較大節點、或針對較小的工作負載選擇較多節點。您應該根據授權允許的內容來選取範本。每個範本選項都會針對每個節點的記憶體、核心和容量、建議符合範本模式的合格節點數量。

7. 設定節點：
   a. 新增節點標籤以識別支援此Astra Data Store叢集的工作節點集區。

   > 在開始部署或部署之前、必須將標籤新增至叢集中用於Astra Data Store部署的各個節點。

   b. 手動設定每個節點的容量（GiB）、或選取允許的最大節點容量。
   c. 設定叢集中允許的節點數目上限、或允許叢集上的節點數目上限。
8. （僅限Astra Data Store完整授權）輸入您要用於Protection Domain的標籤金鑰。

> 為每個節點的金鑰建立至少三個唯一的標籤。例如、如果您的金鑰是「astra.datastore.protection.domain`」、您可以建立下列標籤：「astra.datastore.protection.domain=domain1`,`astra.datastore.protection.domain=domain2`」和「astra.datastore.protection.domain=domain3`」。

9. 設定管理網路：
   a. 輸入Astra Data Store內部管理的管理IP位址、該位址與工作節點IP位址位於同一子網路。
   b. 選擇使用相同的NIC進行管理和資料網路、或分別設定。
   c. 輸入資料網路IP位址集區、子網路遮罩和閘道、以供儲存存取。
10. 檢查組態、然後選取*部署*開始安裝。

結果

成功安裝之後、後端會在後端清單中以「Available（可用）」狀態顯示、並顯示作用中的效能資訊。

> 您可能需要重新整理頁面、以便顯示後端。

使用現有的儲存後端

您可以將探索到ONTAP 的功能區或Astra Data Store儲存後端納入Astra Control Center管理。

步驟

1. 從儀表板或後端功能表瀏覽：
   - 從*儀表板*：從「資源摘要」中、從「儲存後端」窗格中選取連結、然後從「後端」區段中選取「新增」。
   - 從*後端*：
     i. 在左側導覽區域中、選取*後端*。
     ii. 從託管叢集的探索後端選取*管理*、或選取*新增*來管理其他現有後端。
2. 選取*使用現有*索引標籤。
3. 視後端類型而定、執行下列其中一項：
   - * Astra資料儲存區*：
     i. 選擇* Astra Data Store*。
     ii. 選取受管理的運算叢集、然後選取* Next*。
     iii. 確認後端詳細資料、然後選取*「Add storage backend*（新增儲存後端*）」。
   - 《》*：ONTAP
     i. 選擇* ONTAP 《》《*》。
     ii. 輸入ONTAP 該系統的管理員認證資料、然後選取* Review *。
     iii. 確認後端詳細資料、然後選取*「Add storage backend*（新增儲存後端*）」。

結果

後端會以「可用」狀態顯示在清單中、並顯示摘要資訊。

> 您可能需要重新整理頁面、以便顯示後端。

新增儲存庫

如果您想要備份應用程式和持續儲存設備、或是想要跨叢集複製應用程式、則必須新增物件存放區資源庫供應商。Astra Control會將這些備份或複製儲存在您定義的物件存放區中。

當您新增貯體時、Astra Control會將一個貯體標示為預設的貯體指標。您建立的第一個儲存區會成為預設儲存區。

如果您要將應用程式組態和持續儲存設備複製到同一個叢集、則不需要儲存庫。

請使用下列任一種貯體類型：

- NetApp ONTAP 產品S3
- NetApp StorageGRID 產品S3
- 一般S3

> 雖然Astra Control Center支援Amazon S3做為通用S3儲存區供應商、但Astra Control Center可能不支援所有聲稱Amazon S3支援的物件儲存區廠商。

如需如何使用Astra Control API新增儲存區的指示、請參閱 "Astra Automation和API資訊"。

步驟

1. 在左側導覽區域中、選取*鏟斗*。
   a. 選取*「Add*」。
   b. 選取貯體類型。

      > 新增儲存庫時、請選擇正確的儲存庫供應商、並提供該供應商的適當認證資料。例如、UI接受NetApp ONTAP S3作為類型並接受StorageGRID 驗證、但這將導致所有未來使用此儲存庫的應用程式備份與還原失敗。

   c. 建立新的貯體名稱、或輸入現有的貯體名稱和選用說明。

      > 庫位名稱和說明會顯示為備份位置、您可以在建立備份時稍後再選擇。此名稱也會在保護原則組態期間顯示。

   d. 輸入S3端點的名稱或IP位址。
   e. 如果您希望此儲存區成為所有備份的預設儲存區、請勾選「將此儲存區設為此私有雲端的預設儲存區」選項。

      > 此選項不會出現在您所建立的第一個儲存區。

   f. 請繼續新增 認證資訊。

新增**S3**存取認證

隨時新增S3存取認證。

步驟

1. 從「庫位」對話方塊中、選取「新增」或「使用現有」索引標籤。
    a. 在Astra Control中輸入認證與其他認證不同的名稱。
    b. 從剪貼簿貼上內容、輸入存取ID和秘密金鑰。

變更預設儲存類別

您可以變更叢集的預設儲存類別。

步驟

1. 在Astra Control Center Web UI中、選取* Clusters*。
2. 在「叢集」頁面上、選取您要變更的叢集。
3. 選擇* Storage*（儲存設備）選項卡。
4. 選擇*儲存類別*類別。
5. 針對您要設為預設的儲存類別、選取「動作」功能表。
6. 選擇*設為預設*。

接下來呢？

現在您已經登入Astra Control Center並新增叢集、就可以開始使用Astra Control Center的應用程式資料管理功能。

- "管理使用者"
- "開始管理應用程式"
- "保護應用程式"
- "複製應用程式"
- "管理通知"
- "連線Cloud Insights 至"
- "新增自訂TLS憑證"

如需詳細資訊、請參閱

- "使用Astra Control API"
- "已知問題"

新增叢集的必要條件

在新增叢集之前、您應確保符合先決條件。您也應該執行資格檢查、以確保叢集已準備好新增至Astra控制中心。

新增叢集之前所需的功能

- 下列叢集類型之一：
  - 執行OpenShift 4.6-8、4.7、4.8或4.9的叢集
  - 使用RKE1執行Rancher 2.5.x、2.5.x或2.6的叢集
  - 執行Kubernetes 1.20至1.23的叢集
  - 執行VMware Tanzu Kubernetes Grid 1.4的叢集
  - 執行VMware Tanzu Kubernetes Grid整合版1.12.2的叢集

  請確定您的叢集有一個或多個工作節點、且至少有1GB RAM可供執行遙測服務。

  > 如果您計畫將第二個OpenShift 4.6、4.7或4.8叢集新增為受管理的運算資源、則應確保已啟用Astra Trident Volume Snapshot功能。請參閱官方的Astra Trident "說明" 使用Astra Trident啟用及測試Volume Snapshot。

- Astra Trident儲存類設定為 "支援的儲存後端" （任何類型的叢集都需要）
- 支援ONTAP 的支援功能不只是超級使用者和使用者ID、更能使用Astra Control Center來備份和還原應用程式。在支援指令行中執行下列命令ONTAP ：「匯出原則規則modify -vserver <儲存虛擬機器名稱>-policyname <原則名稱>-rueindex 1 -sm超級 使用者sym --anon 65534」
- 由系統管理員定義的Astra Trident「volumesnapshotClass」物件。瞭解Astra Trident "說明" 使用Astra Trident啟用及測試Volume Snapshot。
- 請確定您只為Kubernetes叢集定義單一預設儲存類別。

執行資格檢查

執行下列資格檢查、確保您的叢集已準備好新增至Astra控制中心。

步驟

1. 檢查Trident版本。

   ```
   kubectl get tridentversions -n trident
   ```

   如果存在Trident、您會看到類似下列的輸出：

   ```
   NAME       VERSION
   trident    21.04.0
   ```

   如果Trident不存在、您會看到類似下列的輸出：

   ```
   error: the server doesn't have a resource type "tridentversions"
   ```

> 如果未安裝Trident或安裝的版本不是最新版本、您必須先安裝最新版本的Trident、才能繼續進行。請參閱 "Trident文件" 以取得相關指示。

2. 檢查儲存類別是否使用支援的Trident驅動程式。置備程式名稱應為「csi.trident.netapp.io`」。請參閱下列範例：

```
kubectl get sc
NAME                    PROVISIONER                    RECLAIMPOLICY
VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
ontap-gold (default)    csi.trident.netapp.io          Delete
Immediate           true                   5d23h
thin                    kubernetes.io/vsphere-volume   Delete
Immediate           false                  6d
```

建立管理角色**KECBEConfig**

在執行步驟之前、請先確定機器上有下列項目：

- 安裝了「kubecll」v1.19或更新版本
- 具有作用中內容叢集管理權限的作用中Kbeconfig

步驟

1. 建立服務帳戶、如下所示：

   a. 建立名為「astracontinal-service-account.yaml」的服務帳戶檔案。

      視需要調整名稱和命名空間。如果在此處進行變更、您應該在下列步驟中套用相同的變更。

```
<strong>astracontrol-service-account.yaml</strong>
```

+

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astracontrol-service-account
  namespace: default
```

   a. 套用服務帳戶：

```
kubectl apply -f astracontrol-service-account.yaml
```

2. （選用）如果叢集使用限制的Pod安全性原則、而該原則不允許建立具有權限的Pod、或允許Pod容器內的處理程序以root使用者身分執行、請為叢集建立自訂的Pod安全性原則、讓Astra Control能夠建立及管理Pod。如需相關指示、請參閱 "建立自訂Pod安全性原則"。
3. 授予叢集管理權限、如下所示：

   a. 建立名為「astracontletil-clusterrolebind.yaml」的「ClusterRoeBinding」檔案。

      視需要在建立服務帳戶時調整任何已修改的名稱和命名空間。

```
<strong>astracontrol-clusterrolebinding.yaml</strong>
```

+

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astracontrol-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: astracontrol-service-account
  namespace: default
```

   a. 套用叢集角色繫結：

```
kubectl apply -f astracontrol-clusterrolebinding.yaml
```

4. 列出服務帳戶機密、將「<內容>」取代為正確的安裝內容：

```
kubectl get serviceaccount astracontrol-service-account --context
<context> --namespace default -o json
```

輸出的結尾應類似於下列內容：

```
"secrets": [
{ "name": "astracontrol-service-account-dockercfg-vhz87"},
{ "name": "astracontrol-service-account-token-r59kr"}
]
```

"secretts "陣列中每個元素的索引以0開頭。在上述範例中、「astracontlivter-service-account-dockercfg-vhz87」的索引為0、而「astracontlive-service-account-toke-r59kr"的索引則為1。在輸出中、記下含有「權杖」一詞的服務帳戶名稱索引。

5. 產生以下的Kbeconfig：

   a. 建立「cree-kupeconfig．sh」檔案。將下列指令碼開頭的「toke_index」取代為正確的值。

```
<strong>create-kubeconfig.sh</strong>
```

```
# Update these to match your environment.
# Replace TOKEN_INDEX with the correct value
# from the output in the previous step. If you
# didn't change anything else above, don't change
# anything else here.

SERVICE_ACCOUNT_NAME=astracontrol-service-account
NAMESPACE=default
NEW_CONTEXT=astracontrol
KUBECONFIG_FILE='kubeconfig-sa'

CONTEXT=$(kubectl config current-context)

SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.secrets[TOKEN_INDEX].name}')
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.data.token}')

TOKEN=$(echo ${TOKEN_DATA} | base64 -d)

# Create dedicated kubeconfig
# Create a full copy
kubectl config view --raw > ${KUBECONFIG_FILE}.full.tmp

# Switch working context to correct context
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp config use-context
${CONTEXT}

# Minify
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp \
  config view --flatten --minify > ${KUBECONFIG_FILE}.tmp
```

```
# Rename context
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  rename-context ${CONTEXT} ${NEW_CONTEXT}

# Create token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-credentials ${CONTEXT}-${NAMESPACE}-token-user \
  --token ${TOKEN}

# Set context to use token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token
-user

# Set context to correct namespace
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --namespace ${NAMESPACE}

# Flatten/minify kubeconfig
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  view --flatten --minify > ${KUBECONFIG_FILE}

# Remove tmp
rm ${KUBECONFIG_FILE}.full.tmp
rm ${KUBECONFIG_FILE}.tmp
```

b. 請輸入命令以將其套用至Kubernetes叢集。

```
source create-kubeconfig.sh
```

6. （選用）將KUbeconfig重新命名為有意義的叢集名稱。保護您的叢集認證資料。

```
chmod 700 create-kubeconfig.sh
mv kubeconfig-sa.txt YOUR_CLUSTER_NAME_kubeconfig
```

接下來呢？

現在您已經確認已符合先決條件、您已經準備好了 "新增叢集"。

如需詳細資訊、請參閱

- "Trident文件"
- "使用Astra Control API"

新增自訂TLS憑證

您可以移除現有的自我簽署TLS憑證、並以由憑證授權單位（CA）簽署的TLS憑證取代。

您需要的產品

- Kubernetes叢集已安裝Astra Control Center
- 管理存取叢集上的命令Shell以執行「kubecll」命令
- 來自CA的私密金鑰和憑證檔案

移除自我簽署的憑證

移除現有的自我簽署TLS憑證。

1. 使用SSH、以管理使用者身分登入裝載Astra Control Center的Kubernetes叢集。
2. 使用下列命令尋找與目前憑證相關的TLS密碼、並以Astra Control Center部署命名空間取代「<ACC-deployment-namespace>」：

```
kubectl get certificate -n <ACC-deployment-namespace>
```

3. 使用下列命令刪除目前安裝的機密與憑證：

```
kubectl delete cert cert-manager-certificates -n <ACC-deployment-
namespace>
kubectl delete secret secure-testing-cert -n <ACC-deployment-namespace>
```

新增憑證

新增由CA簽署的TLS憑證。

1. 使用下列命令以CA的私密金鑰和憑證檔案建立新的TLS秘密，並以適當的資訊取代括弧<>中的引數：

```
kubectl create secret tls <secret-name> --key <private-key-filename>
--cert <certificate-filename> -n <ACC-deployment-namespace>
```

2. 使用以下命令和範例編輯叢集自訂資源定義（CRD）檔案、並將「pec.selfSigned`」值變更為「spec.ca.secretName`」、以參照您先前建立的TLS密碼：

```
kubectl edit clusterissuers.cert-manager.io/cert-manager-certificates -n
<ACC-deployment-namespace>
....

#spec:
#  selfSigned: {}

spec:
  ca:
    secretName: <secret-name>
```

3. 使用下列命令和範例輸出來驗證變更是否正確、而且叢集已準備好驗證憑證、並以Astra Control Center部署命名空間取代「<ACC-deployment-namedes>」：

```
kubectl describe clusterissuers.cert-manager.io/cert-manager-
certificates -n <ACC-deployment-namespace>
....

Status:
  Conditions:
    Last Transition Time:  2021-07-01T23:50:27Z
    Message:               Signing CA verified
    Reason:                KeyPairVerified
    Status:                True
    Type:                  Ready
Events:                    <none>
```

4. 使用下列範例建立「create.yaml」檔案、並以適當的資訊取代括弧<>中的預留位置值：

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: <certificate-name>
  namespace: <ACC-deployment-namespace>
spec:
  secretName: <certificate-secret-name>
  duration: 2160h # 90d
  renewBefore: 360h # 15d
  dnsNames:
  - <astra.dnsname.example.com> #Replace with the correct Astra Control
Center DNS address
  issuerRef:
    kind: ClusterIssuer
    name: cert-manager-certificates
```

5. 使用下列命令建立憑證：

```
kubectl apply -f certificate.yaml
```

6. 使用下列命令和範例輸出來驗證憑證是否已正確建立、以及是否已使用您在建立期間所指定的引數（例如名稱、持續時間、續約期限及DNS名稱）。

```
kubectl describe certificate -n <ACC-deployment-namespace>
....

Spec:
  Dns Names:
    astra.example.com
  Duration:  125h0m0s
  Issuer Ref:
    Kind:        ClusterIssuer
    Name:        cert-manager-certificates
  Renew Before:  61h0m0s
  Secret Name:   <certificate-secret-name>
Status:
  Conditions:
    Last Transition Time:  2021-07-02T00:45:41Z
    Message:               Certificate is up to date and has not expired
    Reason:                Ready
    Status:                True
    Type:                  Ready
  Not After:               2021-07-07T05:45:41Z
  Not Before:              2021-07-02T00:45:41Z
  Renewal Time:            2021-07-04T16:45:41Z
  Revision:                1
Events:                    <none>
```

7. 編輯「入口CRD TLS」選項、使用下列命令和範例指向新的憑證密碼、並以適當的資訊取代方括弧<>中的預留位置值：

```
kubectl edit ingressroutes.traefik.containo.us -n <ACC-deployment-
namespace>
....

# tls:
#    options:
#      name: default
#    secretName: secure-testing-cert
#    store:
#      name: default

 tls:
    options:
      name: default
    secretName: <certificate-secret-name>
    store:
      name: default
```

8. 使用網頁瀏覽器瀏覽至Astra Control Center的部署IP位址。
9. 確認憑證詳細資料與您安裝的憑證詳細資料相符。
10. 匯出憑證並將結果匯入網頁瀏覽器中的憑證管理程式。

建立自訂Pod安全性原則

Astra Control需要在其管理的叢集上建立及管理Kubernetes Pod。如果您的叢集使用限制的Pod安全性原則、而該原則不允許建立具有權限的Pod、或允許Pod容器內的處理程序以root使用者身分執行、則您需要建立限制較少的Pod安全性原則、以啟用Astra Control來建立及管理這些Pod。

步驟

1. 為叢集建立比預設限制更少的Pod安全性原則、並將其儲存在檔案中。例如：

```yaml
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: astracontrol
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - '*'
  volumes:
  - '*'
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    rule: 'RunAsAny'
```

2. 為Pod安全性原則建立新角色。

```
kubectl-admin create role psp:astracontrol \
    --verb=use \
    --resource=podsecuritypolicy \
    --resource-name=astracontrol
```

3. 將新角色繫結至服務帳戶。

```
kubectl-admin create rolebinding default:psp:astracontrol \
    --role=psp:astracontrol \
    --serviceaccount=astracontrol-service-account:default
```

Astra Control Center的常見問題集

如果您只是想要快速回答問題、這個常見問題集就能幫上忙。

總覽

以下各節提供使用Astra Control Center時可能會遇到的其他問題解答。如需進一步的說明、請聯絡astra.feedback@netapp.com

存取Astra Control Center

什麼是**Astra Control URL**？

Astra Control Center使用本機驗證和每個環境的專屬URL。

對於URL、請在瀏覽器中輸入您在安裝Astra Control Center時、於Astra_control_center_min.yaml自訂資源定義（CRD）檔案的SPEC.astraAddress欄位中所設定的完整網域名稱（FQDN）。電子郵件是您在Astra_control_center_min.yaml CRD的spec.email欄 位中設定的值。

我正在使用評估授權。如何變更為完整授權？

您可以取得NetApp授權檔案（NLF）、輕鬆變更為完整授權。

步驟

- 從左側導覽中、選取*帳戶*>*授權*。
- 選擇*新增授權*。
- 瀏覽至您下載的授權檔案、然後選取*「Add*（新增*）」。

我正在使用評估授權。我還能管理應用程式嗎？

是的、您可以使用評估授權測試管理應用程式功能。

正在登錄Kubernetes叢集

新增**Astra Control**之後、我需要將工作節點新增至**Kubernetes**叢集。我該怎麼辦？

新的工作者節點可新增至現有的資源池。Astra Control會自動探索這些功能。如果在Astra Control中看不到新節點、請檢查新的工作節點是否執行支援的映像類型。您也可以使用「kubectl Get nodes」命令來驗證新工作節點的健全狀況。

如何正確地取消管理叢集？

1. "從Astra Control取消應用程式管理"。
2. "從Astra Control取消管理叢集"。

從**Astra Control**移除**Kubernetes**叢集之後、應用程式和資料會發生什麼變化？

從Astra Control移除叢集不會對叢集的組態（應用程式和持續儲存）進行任何變更。在該叢集上執行的任何Astra Control快照或應用程式備份都無法還原。由Astra Control所建立的持續儲存備份仍在Astra Control之

內、但無法還原。

> 透過任何其他方法刪除叢集之前、請務必先從Astra Control移除叢集。使用另一個工具刪除叢集時、如果叢集仍由Astra Control進行管理、可能會對Astra Control帳戶造成問題。

*當我取消管理叢集時、NetApp Trident是否會自動從叢集解除安裝？*當您從Astra Control Center取消管理叢集時、Trident不會自動從叢集解除安裝。若要解除安裝Trident、您需要 "請遵循Trident文件中的下列步驟"。

管理應用程式

- Astra Control是否能部署應用程式？*

Astra Control不會部署應用程式。應用程式必須部署在Astra Control之外。

停止從**Astra Control**管理應用程式之後、應用程式會發生什麼事？

將刪除任何現有的備份或快照。應用程式與資料仍可繼續使用。資料管理作業無法用於未受管理的應用程式、或屬於它的任何備份或快照。

- Astra Control能否管理非NetApp儲存設備上的應用程式？*

不可以雖然Astra Control可以探索使用非NetApp儲存設備的應用程式、但它無法管理使用非NetApp儲存設備的應用程式。

*我應該自行管理Astra Control嗎？*不、您不應該自行管理Astra Control、因為它是「系統應用程式」。

*不良的Pod是否會影響應用程式管理？*如果託管應用程式的Pod處於不良狀態、Astra Control將無法建立新的備份與複製。

資料管理作業

我的帳戶中有我沒有建立的快照。他們來自何處？

在某些情況下、Astra Control會自動建立快照、做為備份、複製或還原程序的一部分。

我的應用程式使用數個**PV**。**Astra Control**是否會對所有這些**PVCs**執行快照和備份？

是的。Astra Control在應用程式上執行的快照作業包括繫結至應用程式PVCS的所有PV快照。

我可以直接透過不同的介面或物件儲存設備來管理**Astra Control**所拍攝的快照嗎？

不可以Astra Control所拍攝的快照與備份、只能透過Astra Control進行管理。