使用Astra

Astra Control Center

NetApp
November 21, 2023

目錄

使用Astra

管理應用程式

開始管理應用程式

您先請 "將叢集新增至Astra Control管理"、您可以在叢集上安裝應用程式（Astra Control之外）、然後前往Astra Control的「應用程式」頁面、開始管理應用程式及其資源。

如需詳細資訊、請參閱 "應用程式管理需求"。

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。這些運算子通常採用「傳遞值」而非「傳遞參照」架構來設計。以下是一些遵循這些模式的營運者應用程式：
  - "Apache K8ssandra"
  - "Jenkins CI"
  - "Percona XtraDB叢集"

請注意、Astra Control可能無法複製以「傳遞參考」架構設計的操作員（例如、CockroachDB操作員）。在這些類型的複製作業中、複製的操作員會嘗試從來源操作員參考Kubernetes機密、儘管在複製程序中有自己的新秘密。由於Astra Control不知道來源營運者的Kubernetes機密資料、因此複製作業可能會失敗。

> 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。

在叢集上安裝應用程式

現在您已將叢集新增至Astra Control、您可以在叢集上安裝應用程式或管理現有的應用程式。任何範圍內的應用程式都可以管理命名空間。在Pod上線後、您可以使用Astra Control來管理應用程式。

如需從Helm圖表部署已驗證應用程式的協助、請參閱下列內容：

- "從Helm圖表部署MariaDB"
- "從Helm圖表部署MySQL"
- "從Helm圖表部署Postgres"
- "從Helm圖表部署Jenkins"

管理應用程式

Astra Control可讓您在命名空間層級或Kubernetes標籤上管理應用程式。

> 不支援與Helm 2一起安裝的應用程式。

您可以執行下列活動來管理應用程式：

- 管理應用程式
  - [依命名空間管理應用程式]
  - 依Kubernetes標籤管理應用程式
- [忽略應用程式]
- [取消管理應用程式]

> Astra Control本身並非標準應用程式、而是「系統應用程式」。 您不應嘗試自行管理Astra Control。依預設、Astra Control本身不會顯示用於管理。若要查看系統應用程式、請使用「顯示系統應用程式」篩選器。

如需如何使用Astra Control API管理應用程式的指示、請參閱 "Astra Automation和API資訊"。

> 資料保護作業（複製、備份、還原）及後續持續調整磁碟區大小之後、UI中會顯示新的磁碟區大小、延遲最多20分鐘。資料保護作業只需幾分鐘就能成功完成、您可以使用儲存後端的管理軟體來確認磁碟區大小的變更。

依命名空間管理應用程式

「應用程式」頁面的「探索到」區段會顯示這些命名空間中的命名空間、以及任何已安裝Helm的應用程式或自訂標記的應用程式。您可以選擇個別或在命名空間層級管理每個應用程式。所有這些都達到資料保護作業所需的精細度。

例如、您可能想要設定每週執行時間的「MARIA」備份原則、但您可能需要比這更頻繁地備份「MariaDB」（位於同一個命名空間）。根據這些需求、您需要分別管理應用程式、而非在單一命名空間下管理。

雖然Astra Control可讓您分別管理階層的兩個層級（命名空間和命名空間中的應用程式）、但最佳實務做法是選擇一個或另一個層級。如果在命名空間和應用程式層級同時執行動作、則Astra Control中所採取的動作可能會失敗。

步驟

1. 從左側導覽列選取*應用程式*。
2. 選取*探索*篩選器。

3. 檢視探索到的命名空間清單。展開命名空間以檢視應用程式及相關資源。

   Astra Control會在命名空間中顯示Helm應用程式和自訂標記的應用程式。如果有Helm標籤、則會以標籤圖示來指定。

4. 查看「群組」欄以查看應用程式執行的命名空間（以資料夾圖示指定）。
5. 決定要個別管理每個應用程式、還是在命名空間層級管理。
6. 在階層架構的所需層級找到您想要的應用程式、然後從「選項」功能表的「動作」欄位中選取「管理」。
7. 如果您不想管理應用程式、請從「動作」欄的「選項」功能表中選取「忽略」。

   例如、如果您想要一起管理「MARIA」命名空間下的所有應用程式、使其具有相同的快照和備份原則、您可以管理命名空間、並忽略命名空間中的應用程式。

8. 若要查看託管應用程式清單、請選取*託管*作為顯示篩選器。

> 您剛新增的應用程式可能會在「受保護的」欄下顯示警告圖示、表示尚未備份且尚未排程備份。

9. 若要查看特定應用程式的詳細資料、請選取應用程式名稱。

結果

您選擇管理的應用程式現在可從*「託管」*索引標籤取得。任何忽略的應用程式都會移至*忽略*索引標籤。理想情況下、探索到的索引標籤會顯示零應用程式、以便在安裝新應用程式時、更容易找到及管理。

依Kubernetes標籤管理應用程式

Astra Control在應用程式頁面頂端包含一個名為*定義自訂應用程式*的動作。您可以使用此動作來管理以Kubernetes標籤識別的應用程式。 "深入瞭解如何透過Kubernetes標籤定義自訂應用程式"。

步驟

1. 從左側導覽列選取*應用程式*。
2. 選擇*定義*。
3. 在*定義自訂應用程式*對話方塊中、提供管理應用程式所需的資訊：
   a. 新應用程式：輸入應用程式的顯示名稱。
   b. 叢集：選取應用程式所在的叢集。
   c. *命名空間：*選取應用程式的命名空間。
   d. *標籤：*輸入標籤或從下列資源中選取標籤。
   e. 選取的資源：檢視及管理您要保護的選定Kubernetes資源（Pod、機密、持續磁碟區等）。
      - 展開資源並選取標籤數量、即可檢視可用的標籤。
      - 選取其中一個標籤。

        選擇標籤後、標籤會顯示在*標籤*欄位中。Astra Control也會更新*未選取的資源*區段、以顯示與所選標籤不符的資源。

   f. 未選取的資源：確認您不想保護的應用程式資源。
4. 選擇*定義自訂應用程式*。

結果

Astra Control可管理應用程式。您現在可以在*託管*索引標籤中找到它。

忽略應用程式

如果發現應用程式、它會顯示在探索到的清單中。在此案例中、您可以清除探索到的清單、以便更容易找到新安裝的應用程式。或者、您可能會有正在管理的應用程式、之後決定不再管理這些應用程式。如果您不想管理這些應用程式、您可以指出應該忽略這些應用程式。

此外、您也可能想要在一個命名空間下同時管理應用程式（命名空間管理）。您可以忽略要從命名空間中排除的應用程式。

步驟

1. 從左側導覽列選取*應用程式*。
2. 選擇*已探索*做為篩選條件。
3. 選取應用程式。
4. 從「動作」欄的「選項」功能表中、選取「忽略」。
5. 若要取消忽略、請選取*取消忽略*。

取消管理應用程式

當您不再想要備份、快照或複製應用程式時、可以停止管理應用程式。

> 如果您取消管理應用程式、先前建立的任何備份或快照都將遺失。

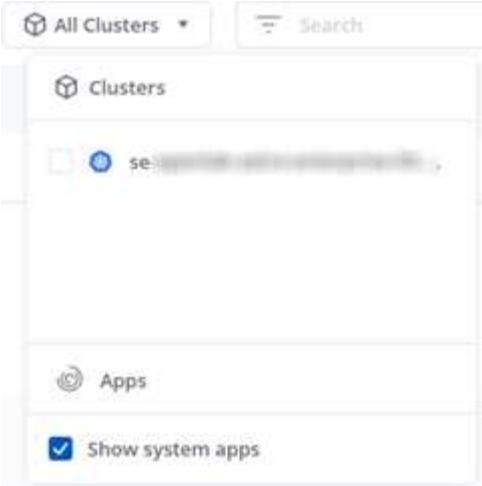
步驟

1. 從左側導覽列選取*應用程式*。
2. 選擇*託管*作爲篩選器。
3. 選取應用程式。
4. 從「動作」欄的「選項」功能表中、選取「取消管理」。
5. 檢閱資訊。
6. 輸入「unManage（取消管理）」以確認。
7. 選擇*是、取消管理應用程式*。

系統應用程式呢？

Astra Control也會探索Kubernetes叢集上執行的系統應用程式。我們預設不會顯示這些系統應用程式、因為您很少需要備份這些應用程式。

您可以從「應用程式」頁面顯示系統應用程式、方法是選取工具列「叢集」篩選器下方的*「顯示系統應用程式*」核取方塊。



Astra Control本身並非標準應用程式、而是「系統應用程式」。 您不應嘗試自行管理Astra Control。依預設、Astra Control本身不會顯示用於管理。

如需詳細資訊、請參閱

- "使用Astra Control API"

定義自訂應用程式範例

建立自訂應用程式可讓您將Kubernetes叢集的元素群組成單一應用程式。這組Kubernetes資源是以命名空間和標籤為基礎。

自訂應用程式可讓您更精細地控制要納入Astra Control作業的內容、包括：

- 複製
- Snapshot
- 備份
- 保護原則

在大多數情況下、您會想要在整個應用程式上使用Astra Control的功能。不過、您也可以建立自訂應用程式、透過指派給命名空間中Kubernetes物件的標籤來使用這些功能。

自訂應用程式只能在單一叢集的指定命名空間內建立。Astra Control不支援自訂應用程式跨越多個命名空間或叢集的功能。

標籤是可指派給Kubernetes物件以供識別的金鑰/值配對。標籤可讓您更輕鬆地排序、組織及尋找Kubernetes物件。若要深入瞭解Kubernetes標籤、 "請參閱Kubernetes官方文件"。

相同資源的原則重疊、名稱不同、可能會造成資料衝突。如果您為資源建立自訂應用程式、請確定該應用程式並未複製或備份到任何其他原則之下。

您需要的產品

- 新增至Astra Control的叢集

步驟

1. 從「應用程式」頁面、選取「**+定義」。

   「自訂應用程式」視窗會顯示哪些資源將納入或排除在自訂應用程式之外。這有助於確保您選擇正確的條件來定義自訂應用程式。

2. 在快顯視窗中、輸入應用程式名稱、在「叢集*」下拉式清單中選擇叢集、然後從「命名空間*」下拉式清單中選擇應用程式的命名空間。
3. 從下拉式* Label *清單中、選取應用程式和命名空間的標籤。
4. 在定義一個部署的自訂應用程式之後、視需要重複其他部署的程序。

當您完成建立這兩個自訂應用程式時、您可以將這些資源視為任何其他Astra Control應用程式。他們可以複製這些資源、建立備份與快照、並根據Kubernetes標籤為每個資源群組建立自訂保護原則。

範例：不同版本的個別保護原則

在此範例中、DevOps團隊正在管理一次一次性發行部署。他們的叢集有三個執行Ngin像 的Pod。其中兩個Pod專用於穩定版本。第三個pod是用於金箱版本。

DevOps團隊的Kubernetes管理員將標籤「部署=穩定」新增至穩定的發行Pod。該團隊將標籤「Deployment = Canary」新增至金級發行Pod。

該團隊的穩定版本包括每小時快照和每日備份的需求。這種精簡版更為短暫、因此他們想要針對任何標示為「部署=資料」的項目、建立更具競爭力的短期保護政策。

為了避免可能的資料衝突、管理員將建立兩個自訂應用程式：一個用於「資料」版本、另一個用於「穩定」版本。如此可將兩個Kubernetes物件群組的備份、快照和複製作業分開進行。

保護應用程式

保護總覽

您可以使用Astra Control Center為應用程式建立備份、複製、快照及保護原則。備份應用程式有助於您的服務和相關資料盡可能可用；在災難案例中、從備份還原可確保應用程式及其相關資料的完整還原、並將中斷時間降至最低。備份、複製和快照有助於防範勒索軟體、意外資料遺失和環境災難等常見威脅。 "瞭解Astra Control Center中可用的資料保護類型、以及使用時間"。

應用程式保護工作流程

您可以使用下列範例工作流程、開始保護應用程式。

[一] 備份所有應用程式

為了確保應用程式立即受到保護、 "建立所有應用程式的手動備份"。

[二] 為每個應用程式設定保護原則

若要自動化未來的備份與快照、 "為每個應用程式設定保護原則"。舉例來說、您可以從每週備份和每日快照開始著手、兩個快照均保留一個月。強烈建議使用保護原則來自動化備份與快照、而不要手動備份與快照。

[三] 選用：調整保護原則

隨著應用程式及其使用模式的改變、請視需要調整保護原則、以提供最佳保護。

[四] 發生災難時、請還原您的應用程式

如果發生資料遺失、您可以透過進行恢復 "還原最新的備份" 每個應用程式的第一名。然後您可以還原最新的快照（如果有）。

利用快照與備份來保護應用程式

使用自動保護原則或臨機操作、拍攝快照和備份資料、保護您的應用程式。您可以使用Astra UI或 "Astra Control API" 保護應用程式。

> 如果您使用Helm來部署應用程式、Astra Control Center需要Helm版本3。完全支援使用Helm 3部署的應用程式管理與複製（或從Helm 2升級至Helm 3）。不支援以Helm 2部署的應用程式。

> 當您在OpenShift叢集上建立裝載應用程式的專案時、專案（或Kubernetes命名空間）會被指派安全性轉換唯一碼。若要啟用Astra Control Center來保護應用程式、並將應用程式移至OpenShift中的其他叢集或專案、您必須新增原則、讓應用程式以任何唯一識別碼的形式執行。例如、下列OpenShift CLI命令會將適當的原則授予WordPress應用程式。
>
> 「occ new－project worm新聞」（occ new－project wormPress）「ocadm policy add－scc to－group anyuid system:serviceaccounts：wormPress」（ocadm policy add－scc對使用者權限-z預設值-n wormPress）

設定保護原則

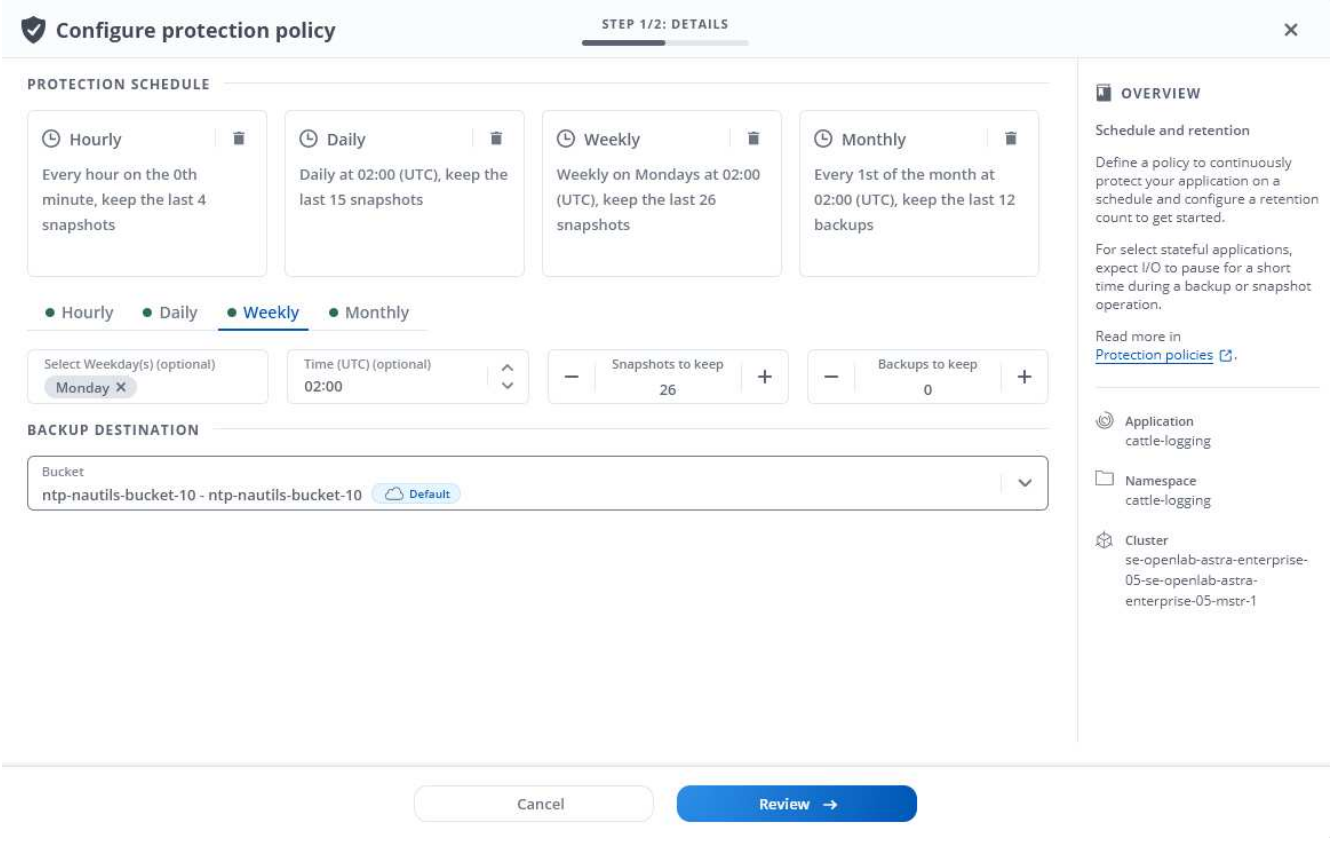
保護原則可在已定義的排程中建立快照、備份或兩者、以保護應用程式。您可以選擇每小時、每天、每週和每月建立快照和備份、也可以指定要保留的複本數量。例如、保護原則可能會建立每週備份和每日快照、並將備份和快照保留一個月。建立快照和備份的頻率、以及保留快照的時間長短、取決於組織的需求。

步驟

1. 選取*應用程式*、然後選取應用程式名稱。
2. 選擇*資料保護*。
3. 選取*設定保護原則*。
4. 選擇每小時、每天、每週和每月保留的快照和備份數量、以定義保護排程。

   您可以同時定義每小時、每日、每週及每月排程。在您設定保留層級之前、排程不會變成作用中。

   下列範例設定四種保護排程：每小時、每日、每週及每月提供快照與備份。



5. 選擇* Review *。
6. 選取*設定保護原則。*

結果

Astra Control Center使用您定義的排程和保留原則、建立並保留快照和備份、以實作資料保護原則。

建立快照

您可以隨時建立隨需快照。

步驟

1. 選擇*應用程式*。
2. 在所需應用程式*「Actions」（動作）欄的「Options」（選項）功能表中、選取*「Snapshot」（快照）*。
3. 自訂快照名稱、然後選取* Review *。
4. 檢閱快照摘要、然後選取* Snapshot *。

結果

快照程序隨即開始。當「資料保護>*快照*」頁面*「動作*」欄中的狀態為*可用*時、快照就會成功。

建立備份

您也可以隨時備份應用程式。

> Astra Control Center中的S3鏟斗未報告可用容量。在備份或複製由Astra Control Center管理的應用程式之前、請先查看ONTAP 資訊庫（英文）或StorageGRID 資訊庫（英文）管理系統中的庫位資訊。

步驟

1. 選擇*應用程式*。
2. 在所需應用程式*「Actions」（動作）欄的「Options」（選項）功能表中、選取「* Backup *」。
3. 自訂備份名稱。
4. 選擇是否要從現有的快照備份應用程式。如果選取此選項、您可以從現有快照清單中進行選擇。
5. 從儲存貯體區清單中選取、以選擇備份目的地。
6. 選擇* Review *。
7. 檢閱備份摘要、然後選取*備份*。

結果

Astra Control Center會建立應用程式的備份。

> 如果您的網路中斷或異常緩慢、備份作業可能會逾時。這會導致備份失敗。

> 無法停止執行中的備份。如果您需要刪除備份、請等到備份完成後再使用中的指示 [刪除備份]。若要刪除失敗的備份、 "使用Astra Control API"。

> 資料保護作業（複製、備份、還原）及後續持續調整磁碟區大小之後、UI中會顯示新的磁碟區大小、延遲最多20分鐘。資料保護作業只需幾分鐘就能成功完成、您可以使用儲存後端的管理軟體來確認磁碟區大小的變更。

檢視快照與備份

您可以從「資料保護」索引標籤檢視應用程式的快照與備份。

步驟

1. 選取*應用程式*、然後選取應用程式名稱。
2. 選擇*資料保護*。

   快照預設會顯示。

3. 選取*備份*以查看備份清單。

刪除快照

刪除不再需要的排程或隨需快照。

步驟

1. 選取*應用程式*、然後選取應用程式名稱。
2. 選擇*資料保護*。

3. 在所需快照*「Actions」（動作）欄的「Options」（選項）功能表中、選取*「Delete snapshot」（刪除快照）*。
4. 輸入「DELETE」一詞以確認刪除、然後選取*「Yes、Delete snapshot （是、刪除快照）」。

結果

Astra Control Center會刪除快照。

刪除備份

刪除不再需要的排程或隨需備份。

> 無法停止執行中的備份。如果您需要刪除備份、請等到備份完成後再使用這些指示。若要刪除失敗的備份、 "使用Astra Control API"。

1. 選取*應用程式*、然後選取應用程式名稱。
2. 選擇*資料保護*。
3. 選擇*備份*。
4. 在所需備份*「Actions」（動作）欄的「**Options**」（選項）功能表中、選取「Delete backup*」（刪除備份*）。
5. 輸入「DELETE」一詞以確認刪除、然後選取*「Yes、Delete backup*（是、刪除備份*）」。

結果

Astra Control Center會刪除備份。

還原應用程式

Astra Control可以從快照或備份還原應用程式。將應用程式還原至同一個叢集時、從現有的快照還原速度會更快。您可以使用Astra Control UI或 "Astra Control API" 以還原應用程式。

關於這項工作

- 強烈建議您在還原應用程式之前、先擷取應用程式的快照或備份應用程式。這可讓您在還原失敗時、從快照或備份進行複製。
- 如果您使用Helm來部署應用程式、Astra Control Center需要Helm版本3。完全支援使用Helm 3部署的應用程式管理與複製（或從Helm 2升級至Helm 3）。不支援以Helm 2部署的應用程式。
- 如果還原至不同的叢集、請確定叢集使用相同的持續磁碟區存取模式（例如ReadWriteMany）。如果目的地持續磁碟區存取模式不同、還原作業將會失敗。
- 任何具有命名空間名稱/ ID或命名空間標籤限制的成員使用者、都可以將應用程式複製或還原到同一個叢集上的新命名空間、或是組織帳戶中的任何其他叢集。不過、相同的使用者無法存取新命名空間中的複製或還原應用程式。在複製或還原作業建立新命名空間之後、帳戶管理員/擁有者可以編輯成員使用者帳戶、並更新受影響使用者的角色限制、以便授予新命名空間的存取權。
- 當您在OpenShift叢集上建立裝載應用程式的專案時、專案（或Kubernetes命名空間）會被指派安全性轉換唯一碼。若要啟用Astra Control Center來保護應用程式、並將應用程式移至OpenShift中的其他叢集或專案、您必須新增原則、讓應用程式以任何唯一識別碼的形式執行。例如、下列OpenShift CLI命令會將適當的原則授予WordPress應用程式。

「occ new－project worm新聞」（occ new－project wormPress）「ocadm policy add－scc to－group anyuid system:serviceaccounts：wormPress」（ocadm policy add－scc對使用者權限-z預設值-n wormPress）

步驟

1. 選取*應用程式*、然後選取應用程式名稱。
2. 選擇*資料保護*。
3. 如果要從快照還原、請選取* Snapshot*圖示。否則、請選取*備份*圖示以從備份還原。
4. 從您要還原之快照或備份的「動作」欄中的「選項」功能表中、選取「還原應用程式」。
5. 還原詳細資料：指定還原應用程式的詳細資料。預設會顯示目前的叢集和命名空間。保留這些值不變、即可將應用程式還原至舊版。如果您要還原至不同的叢集或命名空間、請變更這些值。
   - 輸入應用程式的名稱和命名空間。
   - 選擇應用程式的目的地叢集。
   - 選擇* Review *。

> 如果還原至先前刪除的命名空間、則會在還原程序中建立名稱相同的新命名空間。任何在先前刪除命名空間中擁有管理應用程式權限的使用者、都必須手動還原新重新建立命名空間的權限。

6. 還原摘要：檢閱還原動作的詳細資料、輸入「還原」、然後選取*還原*。

結果

Astra Control Center會根據您提供的資訊來還原應用程式。如果您就地還原應用程式、則任何現有持續磁碟區的內容都會由還原應用程式的持續磁碟區內容取代。

> 在執行資料保護作業（複製、備份、還原）及後續持續調整磁碟區大小之後、新的磁碟區大小會在網路UI中顯示、延遲最多20分鐘。資料保護作業只需幾分鐘就能成功完成、您可以使用儲存後端的管理軟體來確認磁碟區大小的變更。

複製及移轉應用程式

複製現有的應用程式、在相同的Kubernetes叢集或其他叢集上建立複製的應用程式。當Astra Control Center複製應用程式時、會建立應用程式組態和持續儲存的複本。

如果您需要將應用程式和儲存設備從一個Kubernetes叢集移至另一個叢集、複製作業將有助於您。例如、您可能想要透過CI/CD傳輸途徑和Kubernetes命名空間來移動工作負載。您可以使用Astra UI或 "Astra Control API" 複製及移轉應用程式。

您需要的產品

若要將應用程式複製到不同的叢集、您需要預設的儲存區。當您新增第一個儲存區時、它會成為預設儲存區。

關於這項工作

- 如果您部署的應用程式已明確設定StorageClass、且需要複製應用程式、則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。

- 如果您複製由操作人員部署的Jenkins CI執行個體、則需要手動還原持續性資料。這是應用程式部署模式的限制。
- Astra Control Center中的S3鏟斗未報告可用容量。在備份或複製由Astra Control Center管理的應用程式之前、請先查看ONTAP 資訊庫（英文）或StorageGRID 資訊庫（英文）管理系統中的庫位資訊。
- 在應用程式備份或應用程式還原期間、您可以選擇性地指定庫位ID。不過、應用程式複製作業一律會使用已定義的預設儲存區。沒有選項可變更實體複本的儲存區。如果您想要控制所使用的儲存桶、您也可以選擇 "變更庫位預設值" 或執行 "備份" 接著是A "還原" 獨立提供。
- 任何具有命名空間名稱/ ID或命名空間標籤限制的成員使用者、都可以將應用程式複製或還原到同一個叢集上的新命名空間、或是組織帳戶中的任何其他叢集。不過、相同的使用者無法存取新命名空間中的複製或還原應用程式。在複製或還原作業建立新命名空間之後、帳戶管理員/擁有者可以編輯成員使用者帳戶、並更新受影響使用者的角色限制、以便授予新命名空間的存取權。

OpenShift考量

- 如果您在叢集之間複製應用程式、來源叢集和目的地叢集必須是OpenShift的相同發佈版本。例如、如果您從OpenShift 4.7叢集複製應用程式、請使用同樣為OpenShift 4.7的目的地叢集。
- 當您在OpenShift叢集上建立裝載應用程式的專案時、專案（或Kubernetes命名空間）會被指派安全性轉換唯一碼。若要啟用Astra Control Center來保護應用程式、並將應用程式移至OpenShift中的其他叢集或專案、您必須新增原則、讓應用程式以任何唯一識別碼的形式執行。例如、下列OpenShift CLI命令會將適當的原則授予WordPress應用程式。

  「occ new－project worm新聞」（occ new－project wormPress）「ocadm policy add－scc to－group anyuid system:serviceaccounts：wormPress」（ocadm policy add－scc對使用者權限-z預設值-n wormPress）

步驟

1. 選擇*應用程式*。
2. 執行下列其中一項：
   - 在所需應用程式的*「Actions」（動作）欄中、選取「Options」（選項）功能表。
   - 選取所需應用程式的名稱、然後選取頁面右上角的狀態下拉式清單。
3. 選擇* Clone（克隆）*。
4. 複製詳細資料：指定複製的詳細資料：
   - 輸入名稱。
   - 輸入複本的命名空間。
   - 選擇要複製的目的地叢集。
   - 選擇是要從現有的快照或備份建立複本。如果您未選取此選項、Astra Control Center會從應用程式的目前狀態建立複本。
5. 資料來源：如果您選擇從現有的快照或備份中複製、請選擇您要使用的快照或備份。
6. 選擇* Review *。
7. * Clone Summary（複製摘要）：檢閱有關複製的詳細資料、然後選取 Clone（複製）*。

結果

Astra Control Center會根據您提供的資訊來複製該應用程式。當新的應用程式實體複本在「應用程式」頁面上處於「可用」狀態時、即表示該實體複本作業成功。

資料保護作業（複製、備份、還原）及後續持續調整磁碟區大小之後、UI中會顯示新的磁碟區大小、延遲最多20分鐘。資料保護作業只需幾分鐘就能成功完成、您可以使用儲存後端的管理軟體來確認磁碟區大小的變更。

管理應用程式執行掛勾

執行攔截是自訂指令碼、您可以在託管應用程式的快照之前或之後執行。例如、如果您有資料庫應用程式、您可以使用執行掛勾來暫停快照之前的所有資料庫交易、並在快照完成後繼續交易。如此可確保應用程式一致的快照。

預設執行掛勾和規則運算式

對於某些應用程式、Astra Control隨附NetApp提供的預設執行掛勾、可處理快照前後的凍結和解凍作業。Astra Control使用規則運算式、將應用程式的容器映像與下列應用程式配對：

- MariaDB
  - 符合規則運算式：\bmariadb\b
- MySQL
  - 相符的規則運算式：\bmysql\b
- PostgreSQL
  - 相符的規則運算式：\bpostgresql\b

如果有相符項目、則NetApp提供的該應用程式預設執行掛勾會顯示在應用程式的作用中執行掛勾清單中、而這些掛勾會在擷取該應用程式的快照時自動執行。如果其中一個自訂應用程式有類似的映像名稱、正好符合其中一個規則運算式（而且您不想使用預設的執行掛勾）、您可以變更映像名稱、 或停用該應用程式的預設執行掛勾、改用自訂掛勾。

您無法刪除或修改預設的執行掛勾。

關於自訂執行掛勾的重要注意事項

規劃應用程式的執行掛勾時、請考量下列事項。

- Astra Control需要以執行Shell指令碼的格式寫入執行掛勾。
- 指令碼大小限制為128 KB。
- Astra Control會使用執行掛勾設定和任何符合條件來判斷哪些掛勾適用於快照。
- 所有執行掛機故障都是軟性故障、即使掛機故障、仍會嘗試其他掛機和快照。但是、當掛機失敗時、會在*活動*頁面事件記錄中記錄警告事件。
- 若要建立、編輯或刪除執行掛勾、您必須是擁有擁有者、管理員或成員權限的使用者。
- 如果執行掛機執行時間超過25分鐘、掛機將會失敗、並建立傳回代碼為「N/A」的事件記錄項目。任何受影響的快照都會逾時並標示為故障、並會出現一個事件記錄項目、指出逾時時間。

由於執行掛勾通常會減少或完全停用執行中應用程式的功能、因此您應該一律盡量縮短自訂執行掛勾執行所需的時間。

執行快照時、執行掛機事件會依照下列順序進行：

1. 任何適用的NetApp提供的預設快照前執行掛勾、都會在適當的容器上執行。
2. 任何適用的自訂快照前執行掛勾都會在適當的容器上執行。您可以視需要建立及執行任意數量的自訂快照前置掛勾、但在快照之前執行這些掛勾的順序並不保證也無法設定。
3. 快照即會執行。
4. 任何適用的自訂快照後執行掛勾都會在適當的容器上執行。您可以視需要建立及執行任意數量的自訂快照後掛勾、但快照後執行這些掛勾的順序並不保證也無法設定。
5. 任何適用的NetApp提供的預設快照後執行掛勾、都會在適當的容器上執行。

> 在正式作業環境中啟用執行攔截指令碼之前、請務必先進行測試。您可以使用'kubecl exec'命令來方便地測試指令碼。在正式作業環境中啟用執行掛勾之後、請測試所產生的快照、以確保它們一致。您可以將應用程式複製到暫用命名空間、還原快照、然後測試應用程式、藉此完成此作業。

檢視現有的執行掛勾

您可以檢視現有的自訂或NetApp提供的應用程式預設執行勾點。

步驟

1. 移至*應用程式*、然後選取託管應用程式的名稱。
2. 選取*執行掛勾*索引標籤。

   您可以在結果清單中檢視所有已啟用或已停用的執行掛勾。您可以查看Hook的狀態、來源、以及何時執行（快照前或快照後）。若要檢視執行掛起的相關事件記錄、請前往左側導覽區域的*活動*頁面。

建立自訂執行掛勾

您可以為應用程式建立自訂執行掛勾。請參閱 "執行攔截範例" 如需攔截範例、您需要擁有擁有擁有者、管理員或成員權限、才能建立執行掛勾。

> 當您建立自訂Shell指令碼作為執行掛勾時、請記得在檔案開頭指定適當的Shell、除非您執行Linux命令或提供執行檔的完整路徑。

步驟

1. 選取*應用程式*、然後選取託管應用程式的名稱。
2. 選取*執行掛勾*索引標籤。
3. 選取*新增連結*。
4. 在* HookDetails（掛機詳細資料）區域中、視掛機的執行時間而定、選擇*預先快照*或*快照後。
5. 輸入掛機的唯一名稱。
6. （選用）輸入執行期間要傳遞至掛機的任何引數、並在您輸入的每個引數之後按Enter鍵以記錄每個引數。
7. 在「* Container images"（* Container映像*）區域中、如果掛勾應針對應用程式中包含的所有容器映像執行、請啟用「* Apply to all Container images"（套用至所有容器映像）核取方塊。如果掛機只能對一個或多個指定的容器映像起作用、請在「要比對的容器映像名稱」欄位中輸入容器映像名稱。

8. 在*指令碼*區域中、執行下列其中一項：
    - 上傳自訂指令碼。
        i. 選取*上傳檔案*選項。
        ii. 瀏覽至檔案並上傳。
        iii. 為指令碼指定唯一名稱。
        iv. （選用）輸入其他系統管理員應該知道的任何指令碼附註。
    - 從剪貼簿貼入自訂指令碼。
        i. 選擇*從剪貼簿貼上*選項。
        ii. 選取文字欄位、然後將指令碼文字貼到欄位中。
        iii. 為指令碼指定唯一名稱。
        iv. （選用）輸入其他系統管理員應該知道的任何指令碼附註。
9. 選取*新增攔截*。

停用執行掛勾

如果您想要暫時避免在應用程式快照之前或之後執行、可以停用執行掛勾。您需要擁有擁有擁有者、管理員或成員權限、才能停用執行掛勾。

步驟

1. 選取*應用程式*、然後選取託管應用程式的名稱。
2. 選取*執行掛勾*索引標籤。
3. 在「動作」欄中選取「選項」功能表、以顯示您要停用的掛勾。
4. 選擇*停用*。

刪除執行掛勾

如果不再需要執行掛勾、您可以完全移除該掛勾。您需要擁有擁有擁有者、管理員或成員權限、才能刪除執行掛勾。

步驟

1. 選取*應用程式*、然後選取託管應用程式的名稱。
2. 選取*執行掛勾*索引標籤。
3. 在「動作」欄中選取「選項」功能表、以選取您要刪除的掛勾。
4. 選擇*刪除*。

執行攔截範例

請使用下列範例、瞭解如何建立執行掛勾的架構。您可以使用這些hooks做為範本、或做為測試指令碼。

簡單的成功範例

這是一個簡單的勾點、可成功將訊息寫入標準輸出和標準錯誤。

```
#!/bin/sh

# success_sample.sh
#
# A simple noop success hook script for testing purposes.
#
# args: None
#


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#
```

```
# log something to stdout
info "running success_sample.sh"

# exit with 0 to indicate success
info "exit 0"
exit 0
```

簡單的成功範例（**Bash**版本）

這是一個簡單的攔截、成功將訊息寫入標準輸出和標準錯誤（寫入Bash）。

```
#!/bin/bash

# success_sample.bash
#
# A simple noop success hook script for testing purposes.
#
# args: None


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
```

```
}


#
# main
#

# log something to stdout
info "running success_sample.bash"

# exit with 0 to indicate success
info "exit 0"
exit 0
```

簡單的成功範例（**zsh**版本）

這是一個簡單的勾點範例、可成功將訊息寫入標準輸出和標準錯誤、並寫入Z Shell。

```
#!/bin/zsh

# success_sample.zsh
#
# A simple noop success hook script for testing purposes.
#
# args: None
#


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}
```

```
#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample.zsh"

# exit with 0 to indicate success
info "exit 0"
exit 0
```

成功的引數範例

下列範例示範如何在攔截中使用args。

```
#!/bin/sh

# success_sample_args.sh
#
# A simple success hook script with args for testing purposes.
#
# args: Up to two optional args that are echoed to stdout

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
```

```
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running success_sample_args.sh"


# collect args
arg1=$1
arg2=$2

# output args and arg count to stdout
info "number of args: $#"
info "arg1 ${arg1}"
info "arg2 ${arg2}"

# exit with 0 to indicate success
info "exit 0"
exit 0
```

快照前/快照後掛機範例

以下範例說明如何將相同的指令碼同時用於快照前和快照後掛勾。

```
#!/bin/sh

# success_sample_pre_post.sh
#
# A simple success hook script example with an arg for testing purposes
```

```
# to demonstrate how the same script can be used for both a prehook and
posthook
#
# args: [pre|post]

# unique error codes for every error case
ebase=100
eusage=$((ebase+1))
ebadstage=$((ebase+2))
epre=$((ebase+3))
epost=$((ebase+4))


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# Would run prehook steps here
#
prehook() {
```

```
    info "Running noop prehook"
    return 0
}

#
# Would run posthook steps here
#
posthook() {
    info "Running noop posthook"
    return 0
}


#
# main
#

# check arg
stage=$1
if [ -z "${stage}" ]; then
    echo "Usage: $0 <pre|post>"
    exit ${eusage}
fi

if [ "${stage}" != "pre" ] && [ "${stage}" != "post" ]; then
    echo "Invalid arg: ${stage}"
    exit ${ebadstage}
fi

# log something to stdout
info "running success_sample_pre_post.sh"

if [ "${stage}" = "pre" ]; then
    prehook
    rc=$?
    if [ ${rc} -ne 0 ]; then
        error "Error during prehook"
    fi
fi

if [ "${stage}" = "post" ]; then
    posthook
    rc=$?
    if [ ${rc} -ne 0 ]; then
        error "Error during posthook"
    fi
```

```
fi

exit ${rc}
```

故障範例

下列範例示範如何處理攔截式故障。

```
#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#



#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}
```

```
#
# main
#

# log something to stdout
info "running failure_sample_arg_exit_code.sh"

argexitcode=$1

# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}
```

詳細故障範例

下列範例示範如何以更詳細的記錄功能來處理掛機中的失敗。

```
#!/bin/sh

# failure_sample_verbose.sh
#
# A simple failure hook script with args for testing purposes.
#
# args: [The number of lines to output to stdout]


#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
```

```
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running failure_sample_verbose.sh"


# output arg value to stdout
linecount=$1
info "line count ${linecount}"

# write out a line to stdout based on line count arg
i=1
while [ "$i" -le ${linecount} ]; do
    info "This is line ${i} from failure_sample_verbose.sh"
    i=$(( i + 1 ))
done

error "exiting with error code 8"
exit 8
```

退出程式碼範例失敗

下列範例示範攔截失敗、並顯示結束程式碼。

```
#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#
```

```
#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
# main
#

# log something to stdout
info "running failure_sample_arg_exit_code.sh"

argexitcode=$1

# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}
```

失敗後成功範例

下列範例示範第一次執行時發生掛機故障、但在第二次執行後成功。

```
#!/bin/sh

# failure_then_success_sample.sh
#
# A hook script that fails on initial run but succeeds on second run for
testing purposes.
#
# Helpful for testing retry logic for post hooks.
#
# args: None
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}


#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}


#
```

```
# main
#

# log something to stdout
info "running failure_success sample.sh"


if [ -e /tmp/hook-test.junk ] ; then
    info "File does exist.  Removing /tmp/hook-test.junk"
    rm /tmp/hook-test.junk
    info "Second run so returning exit code 0"
    exit 0
else
    info "File does not exist.  Creating /tmp/hook-test.junk"
    echo "test" > /tmp/hook-test.junk
    error "Failed first run, returning exit code 5"
    exit 5
fi
```

檢視應用程式和叢集健全狀況

檢視應用程式與叢集健全狀況的摘要

選取*儀表板*以查看應用程式、叢集、儲存後端及其健全狀況的高層級檢視。

這些不只是靜態數字或狀態、您可以逐一深入瞭解。例如、如果應用程式未受到完整保護、您可以將游標停留在圖示上、以識別哪些應用程式未受到完整保護、這也是原因之一。

應用程式並排顯示

「應用程式」方塊可協助您識別下列項目：

- 您目前使用Astra管理的應用程式數量。
- 這些託管應用程式是否健全。
- 應用程式是否受到完整保護（如果有最近的備份可用、則會受到保護）。
- 已探索但尚未管理的應用程式數量。

  理想情況下、這個數字會是零、因為您會在發現應用程式之後管理或忽略這些應用程式。然後、您可以監控儀表板上探索到的應用程式數量、以識別開發人員何時將新應用程式新增至叢集。

叢集並排顯示

「叢集」方塊提供類似的詳細資料、說明您使用Astra Control Center管理的叢集健全狀況、您也可以深入瞭解更多詳細資料、就像使用應用程式一樣。

儲存後端並排顯示

「儲存後端」方塊提供資訊、協助您識別儲存後端的健全狀況、包括：

- 管理多少個儲存後端
- 這些託管後端是否健全
- 後端是否受到完整保護
- 已探索但尚未管理的後端數目。

檢視叢集的健全狀況和詳細資料

新增要由Astra Control Center管理的叢集之後、您可以檢視叢集的詳細資料、例如其位置、工作節點、持續磁碟區和儲存類別。

步驟

1. 在Astra Control Center UI中、選取* Clusters*。
2. 在「叢集」頁面上、選取您要檢視其詳細資料的叢集。

> 如果叢集位於 `removed` 狀態但叢集和網路連線似乎正常（外部使用Kubernetes API存取叢集的嘗試成功）、您提供給Astra Control的Kubeconfig可能不再有效。這可能是因為叢集上的憑證輪替或過期。若要修正此問題、請使用更新Astra Control中與叢集相關的認證資料 "Astra Control API"。

3. 查看*概述*、*儲存設備*和*活動*索引標籤上的資訊、以尋找您要尋找的資訊。
   - 總覽：工作節點的詳細資料、包括其狀態。
   - * Storage *：與運算相關的持續磁碟區、包括儲存類別和狀態。
   - 活動：顯示與叢集相關的活動。

> 您也可以從Astra控制中心*儀表板*開始檢視叢集資訊。在*叢集*索引標籤的*資源摘要*下、您可以選取受管理的叢集、然後前往*叢集*頁面。進入「叢集」頁面之後、請依照上述步驟操作。

檢視應用程式的健全狀況和詳細資料

在您開始管理應用程式之後、Astra會提供應用程式的詳細資料、讓您識別應用程式的狀態（是否健全）、保護狀態（是否在故障時受到完整保護）、Pod、持續儲存設備等。

步驟

1. 在Astra Control Center UI中、選取* Applications*、然後選取應用程式名稱。
2. 尋找您要尋找的資訊：

   應用程式狀態

   提供反映Kubernetes應用程式狀態的狀態。例如、Pod和持續磁碟區是否在線上？如果某個應用程式不健全、您必須查看Kubernetes記錄檔、在叢集上進行疑難排解。Astra並未提供資訊來協助您修正毀損的應用程式。

應用程式保護狀態

提供應用程式受到保護的程度狀態：

- 完全保護：應用程式有作用中的備份排程、而且備份成功的時間不到一週
- 部分保護：應用程式有作用中的備份排程、作用中的快照排程、或成功的備份或快照
- 未受保護：未受到完整保護或部分保護的應用程式。

  您必須等到最近進行備份之後、才能獲得完整保護。這很重要、因為備份儲存在遠離持續磁碟區的物件存放區中。如果故障或意外將叢集及其持續儲存設備擦除、則需要備份才能恢復。快照無法讓您恢復。

總覽

與應用程式相關聯的Pod狀態資訊。

資料保護

可讓您設定資料保護原則、並檢視現有的快照與備份。

儲存設備

顯示應用程式層級的持續磁碟區。持續磁碟區的狀態是從Kubernetes叢集的觀點來看。

資源

可讓您驗證要備份和管理的資源。

活動

顯示與應用程式相關的活動。

> 您也可以從Astra Control Center * Dashboard 開始檢視應用程式資訊。在*應用程式*索引標籤的*資源摘要*下、您可以選取託管應用程式、以前往*應用程式*頁面。進入「*應用程式」頁面之後、請依照上述步驟操作。

管理您的帳戶

管理使用者

您可以使用Astra Control UI來邀請、新增、移除及編輯Astra Control Center安裝的使用者。您可以使用Astra Control UI或 "Astra Control API" 管理使用者：

邀請使用者

帳戶擁有者和管理員可以邀請新使用者前往Astra Control Center。

步驟

1. 在*管理您的帳戶*導覽區域中、選取*帳戶*。
2. 選取*使用者*索引標籤。
3. 選擇*邀請使用者*。

4. 輸入使用者名稱和電子郵件地址。
5. 選取具有適當系統權限的使用者角色。

   每個角色都提供下列權限：

   - *檢視器*可以檢視資源。
   - *成員*具有檢視者角色權限、可管理應用程式與叢集、取消管理應用程式、以及刪除快照與備份。
   - 「管理」具有「成員角色」權限、可新增及移除擁有者以外的任何其他使用者。
   - *擁有者*具有管理員角色權限、可新增及移除任何使用者帳戶。
6. 若要新增限制給具有成員或檢視者角色的使用者、請啟用*限制角色限制*核取方塊。

   如需新增限制的詳細資訊、請參閱 "管理角色"。

7. 選擇*邀請使用者*。

   使用者會收到一封電子郵件、通知他們已被邀請前往Astra Control Center。電子郵件包含暫時性密碼、首次登入時必須變更。

新增使用者

帳戶擁有者和系統管理員可以新增更多使用者至Astra Control Center安裝。

步驟

1. 在*管理您的帳戶*導覽區域中、選取*帳戶*。
2. 選取*使用者*索引標籤。
3. 選取*新增使用者*。
4. 輸入使用者的名稱、電子郵件地址和暫用密碼。

   使用者必須在第一次登入時變更密碼。

5. 選取具有適當系統權限的使用者角色。

   每個角色都提供下列權限：

   - *檢視器*可以檢視資源。
   - *成員*具有檢視者角色權限、可管理應用程式與叢集、取消管理應用程式、以及刪除快照與備份。
   - 「管理」具有「成員角色」權限、可新增及移除擁有者以外的任何其他使用者。
   - *擁有者*具有管理員角色權限、可新增及移除任何使用者帳戶。
6. 若要新增限制給具有成員或檢視者角色的使用者、請啟用*限制角色限制*核取方塊。

   如需新增限制的詳細資訊、請參閱 "管理角色"。

7. 選取*「Add*」。

管理密碼

您可以在Astra Control Center中管理使用者帳戶的密碼。

變更您的密碼

您可以隨時變更使用者帳戶的密碼。

步驟

1. 選取畫面右上角的使用者圖示。
2. 選擇*設定檔*。
3. 從「動作」欄的「選項」功能表中選取「變更密碼」。
4. 輸入符合密碼需求的密碼。
5. 再次輸入密碼進行確認。
6. 選擇*變更密碼*。

重設其他使用者的密碼

如果您的帳戶具有「管理員」或「擁有者」角色權限、您可以重設其他使用者帳戶和您自己的密碼。當您重設密碼時、您會設定使用者登入時必須變更的暫用密碼。

步驟

1. 在*管理您的帳戶*導覽區域中、選取*帳戶*。
2. 選取「動作」下拉式清單。
3. 選擇*重設密碼*。
4. 輸入符合密碼需求的暫用密碼。
5. 再次輸入密碼進行確認。

> 下次使用者登入時、系統會提示使用者變更密碼。

6. 選擇*重設密碼*。

變更使用者角色

擁有擁有者角色的使用者可以變更所有使用者的角色、而擁有管理員角色的使用者則可以變更擁有管理員、成員或檢視者角色的使用者角色。

步驟

1. 在*管理您的帳戶*導覽區域中、選取*帳戶*。
2. 選取「動作」下拉式清單。
3. 選擇*編輯角色*。
4. 選取新角色。
5. 若要將限制套用至角色、請啟用*限制角色至限制*核取方塊、然後從清單中選取限制。

如果沒有限制、您可以新增限制。如需詳細資訊、請參閱 "管理角色"。

6. 選擇* Confirm（確認）*。

結果

Astra Control Center會根據您選取的新角色來更新使用者權限。

移除使用者

擁有擁有者或管理員角色的使用者可以隨時從帳戶中移除其他使用者。

步驟

1. 在*管理您的帳戶*導覽區域中、選取*帳戶*。
2. 在「使用者」索引標籤中、選取您要移除之每個使用者列中的核取方塊。
3. 從「動作」欄的「選項」功能表中、選取「移除使用者」。
4. 出現提示時、請輸入「移除」一詞、然後選取「是、移除使用者*」、確認刪除。

結果

Astra Control Center會將使用者從帳戶中移除。

管理角色

您可以新增命名空間限制、並將使用者角色限制在這些限制中、藉此管理角色。這可讓您控制組織內資源的存取。您可以使用Astra Control UI或 "Astra Control API" 以管理角色。

將命名空間限制新增至角色

管理員或擁有者使用者可以新增命名空間限制。

步驟

1. 在*管理您的帳戶*導覽區域中、選取*帳戶*。
2. 選取*使用者*索引標籤。
3. 在「動作」欄中、選取具有「成員」或「檢視者」角色的使用者功能表按鈕。
4. 選擇*編輯角色*。
5. 啟用「限制角色*」核取方塊。

   此核取方塊僅適用於「成員」或「檢視者」角色。您可以從*角色*下拉式清單中選取不同的角色。

6. 選取*新增限制*。

   您可以依命名空間或命名空間標籤檢視可用限制清單。

7. 在*限制類型*下拉式清單中、視命名空間的設定方式而定、選取* Kubernetes命名空間*或* Kubernetes命名空間標籤*。
8. 從清單中選取一或多個命名空間或標籤、以構成限制、限制角色只能使用這些命名空間。
9. 選擇* Confirm（確認）*。

「編輯角色」頁面會顯示您為此角色選擇的限制清單。

10. 選擇* Confirm（確認）*。

    在「帳戶」頁面上、您可以在「角色」欄中檢視任何成員或檢視者角色的限制條件。

> 如果您啟用角色的限制、並選取* Confirm（確認）*而不新增任何限制、則該角色會被視為具有完整限制（該角色無法存取指派給命名空間的任何資源）。

從角色移除命名空間限制

管理員或擁有者使用者可以從角色移除命名空間限制。

步驟

1. 在*管理您的帳戶*導覽區域中、選取*帳戶*。
2. 選取*使用者*索引標籤。
3. 在「動作」欄中、選取具有作用中限制之「成員」或「檢視者」角色的使用者功能表按鈕。
4. 選擇*編輯角色*。

   「編輯角色」對話方塊會顯示角色的作用中限制。

5. 選取您需要移除之限制右側的* X*。
6. 選擇* Confirm（確認）*。

以取得更多資訊

- "使用者角色和命名空間"

檢視及管理通知

Astra會在行動完成或失敗時通知您。例如、如果成功完成應用程式的備份、您會看到通知。

您可以從介面右上角管理這些通知：

步驟

1. 選取右上角的未讀取通知數。
2. 檢閱通知、然後選取*標示為已讀取*或*顯示所有通知*。

   如果您選取*顯示所有通知*、則會載入「通知」頁面。

3. 在*通知*頁面上、檢視通知、選取您要標示為已讀的通知、選取*行動*、然後選取*標示為已讀*。

新增及移除認證資料

隨時從ONTAP 您的帳戶新增及移除本地私有雲端供應商的認證資料、例如用OpenShift管理的Kubernetes叢集、或Unmanaged Kubernetes叢集。Astra Control Center會使用這些認證資料來探索叢集和叢集上的應用程式、並代表您配置資源。

請注意、Astra Control Center中的所有使用者都共用相同的認證資料集。

新增認證資料

您可以在管理叢集時、將認證新增至Astra Control Center。若要新增叢集以新增認證、請參閱 "新增Kubernetes叢集"。

> 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。

移除認證資料

隨時從帳戶移除認證資料。您只能在之後移除認證 "取消管理所有相關的叢集"。

> 您新增至Astra Control Center的第一組認證資料一律使用中、因為Astra Control Center使用認證資料來驗證備份儲存區。最好不要移除這些認證資料。

步驟

1. 選擇*帳戶*。
2. 選取*認證*索引標籤。
3. 在*狀態*欄中選取您要移除之認證的「選項」功能表。
4. 選擇*移除*。
5. 輸入「移除」一詞以確認刪除、然後選取*是、移除認證*。

結果

Astra Control Center會從帳戶移除認證資料。

監控帳戶活動

您可以檢視Astra Control帳戶中活動的詳細資料。例如、當邀請新使用者、新增叢集或擷取快照時。您也可以將帳戶活動匯出至CSV檔案。

檢視**Astra Control**中的所有帳戶活動

1. 選擇*活動*。
2. 使用篩選器縮小活動清單範圍、或使用搜尋方塊找到您想要的確切內容。
3. 選取*匯出至CSV*、將您的帳戶活動下載至CSV檔案。

檢視特定應用程式的帳戶活動

1. 選取*應用程式*、然後選取應用程式名稱。

2. 選擇*活動*。

檢視叢集的帳戶活動

1. 選取*叢集*、然後選取叢集名稱。
2. 選擇*活動*。

採取行動以解決需要注意的事件

1. 選擇*活動*。
2. 選取需要注意的事件。
3. 選取*「採取行動」*下拉式選項。

   您可在此清單中檢視可能採取的修正行動、檢視與問題相關的文件、並取得協助解決問題的支援。

更新現有授權

您可以將試用版授權轉換為完整授權、也可以使用新授權來更新現有的試用版或完整授權。如果您沒有完整授權、請與NetApp銷售聯絡人聯絡、以取得完整授權與序號。您可以使用Astra UI或 "Astra Control API" 以更新現有授權。

步驟

1. 登入 "NetApp 支援網站"。
2. 存取Astra Control Center下載頁面、輸入序號、然後下載完整的NetApp授權檔案（NLF）。
3. 登入Astra Control Center UI。
4. 從左側導覽中、選取*帳戶*>*授權*。
5. 在「帳戶>*授權*」頁面中、選取現有授權的狀態下拉式功能表、然後選取「取代」。
6. 瀏覽至您下載的授權檔案。
7. 選取*「Add*」。

「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。

以取得更多資訊

- "Astra Control Center授權"

管理儲存庫連線

您可以將儲存庫連線至Astra Control、作為軟體套件安裝映像和成品的參考。當您匯入軟體套件時、Astra Control會參考映像儲存庫中的安裝映像、二進位檔及成品儲存庫中的其他成品。

您需要的產品

- Kubernetes叢集已安裝Astra Control Center
- 可存取的執行中Docker儲存庫
- 可存取的執行中成品儲存庫（例如Artifactory）

連接Docker映像儲存庫

您可以連接Docker映像儲存庫來保存套件安裝映像、例如Astra Data Store的安裝映像。安裝套件時、Astra Control會從映像儲存庫匯入套件映像檔。

步驟

1. 在*管理您的帳戶*導覽區域中、選取*帳戶*。
2. 選取*連線*索引標籤。
3. 在「* Docker Image Repository 」（ Docker影像儲存庫*）區段中、選取右上角的功能表。
4. 選擇*連接*。
5. 新增儲存庫的URL和連接埠。
6. 輸入儲存庫的認證資料。
7. 選擇*連接*。

結果

儲存庫已連線。在「* Docker Image Repository 」（ Docker映像儲存庫*）區段中、儲存庫應顯示連線狀態。

中斷Docker映像儲存庫的連線

如果不再需要、您可以移除與Docker映像儲存庫的連線。

步驟

1. 在*管理您的帳戶*導覽區域中、選取*帳戶*。
2. 選取*連線*索引標籤。
3. 在「* Docker Image Repository 」（ Docker影像儲存庫*）區段中、選取右上角的功能表。
4. 選擇*中斷連線*。
5. 選取*是、中斷Docker映像儲存庫的連線*。

結果

儲存庫已中斷連線。在「* Docker Image Repository 」（ Docker映像儲存庫*）區段中、儲存庫應顯示中斷連線狀態。

連接成品儲存庫

您可以將成品儲存庫連線至主機成品、例如軟體套件二進位檔。安裝套件時、Astra Control會從映像儲存庫匯入軟體套件的成品。

步驟

1. 在*管理您的帳戶*導覽區域中、選取*帳戶*。
2. 選取*連線*索引標籤。
3. 在「雜訊儲存庫」區段中、選取右上角的功能表。
4. 選擇*連接*。
5. 新增儲存庫的URL和連接埠。

6. 如果需要驗證、請啟用「使用驗證」核取方塊、然後輸入儲存庫的認證資料。
7. 選擇*連接*。

結果

儲存庫已連線。在「雜訊儲存庫」區段中、儲存庫應顯示連線狀態。

中斷成品儲存庫的連線

如果不再需要、您可以移除與成品儲存庫的連線。

步驟

1. 在*管理您的帳戶*導覽區域中、選取*帳戶*。
2. 選取*連線*索引標籤。
3. 在「雜訊儲存庫」區段中、選取右上角的功能表。
4. 選擇*中斷連線*。
5. 選取*是、中斷產出工件儲存庫的連線*。

結果

儲存庫已中斷連線。在「雜訊儲存庫」區段中、儲存庫應顯示連線狀態。

如需詳細資訊、請參閱

- "管理軟體套件"

管理軟體套件

NetApp為Astra Control Center提供額外功能、提供您可從NetApp支援網站下載的軟體套件。連接Docker和成品儲存庫之後、您可以上傳和匯入套件、將此功能新增至Astra Control Center。您可以使用CLI或Astra Control Center Web UI來管理軟體套件。

您需要的產品

- Kubernetes叢集已安裝Astra Control Center
- 連線的Docker映像儲存庫、可容納軟體套件映像。如需詳細資訊、請參閱 "管理儲存庫連線"。
- 連線的成品儲存庫、用於儲存軟體套件二進位檔和成品。如需詳細資訊、請參閱 "管理儲存庫連線"。
- NetApp支援網站提供的軟體套件

將軟體套件映像上傳至儲存庫

Astra Control Center會參考連線儲存庫中的套件映像和成品。您可以使用CLI將影像和成品上傳至儲存庫。

步驟

1. 請從NetApp支援網站下載軟體套件、並將其儲存在已安裝「kubecll」公用程式的機器上。
2. 擷取壓縮的套件檔案、並將目錄變更為Astra Control套裝組合檔案的位置（例如、「acs.manifest.bunder.yaml」）。
3. 將套件映像推送到Docker儲存庫。進行下列替代：

- 以Astra Control套裝組合檔案的名稱取代bunder_file。
- 將my_registry取代為Docker儲存庫的URL。
- 以儲存庫的認證資料取代my_register_user和my_register_password。

```
kubectl astra packages push-images -m BUNDLE_FILE -r MY_REGISTRY -u
MY_REGISTRY_USER -p MY_REGISTRY_PASSWORD
```

4. 如果套件含有成品、請將成品複製到成品儲存庫。以Astra Control套件檔案的名稱取代bunder_file、並以網路位置取代network_location、將成品檔案複製到：

```
kubectl astra packages copy-artifacts -m BUNDLE_FILE -n NETWORK_LOCATION
```

新增軟體套件

您可以使用Astra Control Center套裝組合檔案匯入軟體套件。這樣做會安裝套件、讓Astra Control Center能夠使用該軟體。

使用**Astra Control**網路**UI**新增軟體套件

您可以使用Astra Control Center網路UI來新增已上傳至連線儲存庫的軟體套件。

步驟

1. 在*管理您的帳戶*導覽區域中、選取*帳戶*。
2. 選取*套件*索引標籤。
3. 選取*「Add*（新增*）」按鈕。
4. 在檔案選擇對話方塊中、選取上傳圖示。
5. 選擇要上傳的Astra Control套裝組合檔案、格式為「.yaml」。
6. 選取*「Add*」。

結果

如果套件檔案有效、且套件映像和成品位於連線的儲存庫中、則套件會新增至Astra Control Center。當「狀態」欄中的狀態變更為*可用*時、您可以使用套件。您可以將游標暫留在套件的狀態上、以取得更多資訊。

> 如果您的儲存庫中找不到套件的一或多個影像或成品、則會出現該套件的錯誤訊息。

使用**CLI**新增軟體套件

您可以使用CLI匯入已上傳至連線儲存庫的軟體套件。為達成此目的、您必須先記錄Astra Control Center帳戶ID和API權杖。

步驟

1. 使用網頁瀏覽器登入Astra Control Center網頁UI。
2. 從儀表板中、選取右上角的使用者圖示。

3. 選擇* API存取*。
4. 請記下畫面頂端附近的帳戶ID。
5. 選取*產生API權杖*。
6. 在產生的對話方塊中、選取*產生API權杖*。
7. 記下產生的權杖、然後選取*關閉*。在CLI中、將目錄變更為擷取套件內容中的「.yaml」套件檔案位置。
8. 使用套件檔案匯入套件、並進行下列替代：
    - 以Astra Control套裝組合檔案的名稱取代bunder_file。
    - 以Astra Control執行個體的DNS名稱取代伺服器。
    - 使用您先前記錄的帳戶ID和API Token來取代帳戶ID和Token。

```
kubectl astra packages import -m BUNDLE_FILE -u SERVER -a ACCOUNT_ID
-k TOKEN
```

結果

如果套件檔案有效、且套件映像和成品位於連線的儲存庫中、則套件會新增至Astra Control Center。

> 如果您的儲存庫中找不到套件的一或多個影像或成品、則會出現該套件的錯誤訊息。

移除軟體套件

您可以使用Astra Control Center網路UI移除先前匯入Astra Control Center的軟體套件。

步驟

1. 在*管理您的帳戶*導覽區域中、選取*帳戶*。
2. 選取*套件*索引標籤。

    您可以在此頁面上查看已安裝套件的清單及其狀態。

3. 在套件的「動作」欄中、開啟「動作」功能表。
4. 選擇*刪除*。

結果

套件會從Astra Control Center刪除、但套件的映像和成品仍會保留在儲存庫中。

如需詳細資訊、請參閱

- "管理儲存庫連線"

管理儲存庫

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、物件存放區供應商是不可或缺的。使用Astra Control Center、新增物件存放區供應商做為您的應用程式離叢集備份目的地。

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。

請使用下列其中一家Amazon Simple Storage Service（S3）資源庫供應商：

- NetApp ONTAP 產品S3
- NetApp StorageGRID 產品S3
- 一般S3
- Microsoft Azure

> 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。

儲存庫可以位於下列其中一種狀態：

- 擱置中：已排定要探索的儲存區。
- 可用：鏟斗可供使用。
- 已移除：目前無法存取儲存貯體。

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

您可以執行與管理儲存庫相關的工作：

- "新增儲存庫"
- [編輯儲存庫]
- [旋轉或移除庫位認證資料]
- [移除貯體]

> Astra Control Center中的S3鏟斗未報告可用容量。在備份或複製由Astra Control Center管理的應用程式之前、請先查看ONTAP 資訊庫（英文）或StorageGRID 資訊庫（英文）管理系統中的庫位資訊。

編輯儲存庫

您可以變更儲存區的存取認證資訊、並變更所選儲存區是否為預設儲存區。

> 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。請參閱 "版本資訊"。

步驟

1. 從左側導覽中、選取*鏟斗*。
2. 從「動作」欄的「選項」功能表中、選取「編輯」。
3. 變更儲存桶類型以外的任何資訊。

您無法修改貯體類型。

4. 選擇*更新*。

旋轉或移除庫位認證資料

Astra Control使用儲存區認證來取得S3儲存區的存取權、並提供密碼金鑰、以便Astra Control Center能夠與儲存區通訊。

旋轉儲存庫認證資料

如果您旋轉認證資料、請在維護期間（排程或隨需）無備份進行時、於維護期間旋轉認證資料。

編輯及旋轉認證的步驟

1. 從左側導覽中、選取*鏟斗*。
2. 從「動作」欄的「選項」功能表中、選取「編輯」。
3. 建立新認證資料。
4. 選擇*更新*。

移除庫位認證資料

只有在新認證已套用至庫位、或庫位已不再有效使用時、才應移除庫位認證。

您新增至Astra Control的第一組認證資料一律使用中、因為Astra Control使用認證資料來驗證備份儲存區。如果儲存區正在使用中、請勿移除這些認證資料、因為這會導致備份失敗和備份不可用。

如果您確實移除作用中的儲存區認證、請參閱 "移除庫位認證疑難排解"。

如需如何使用Astra Control API移除S3認證的指示、請參閱 "Astra Automation和API資訊"。

移除貯體

您可以移除不再使用或不健全的庫位。您可能會想要這麼做、讓物件存放區組態保持簡單且最新狀態。

您無法移除預設的儲存區。如果您要移除該儲存區、請先選取另一個儲存區做為預設值。

您需要的產品

- 開始之前、您應檢查以確保此儲存區沒有執行中或已完成的備份。
- 您應檢查以確保儲存庫未用於任何作用中的保護原則。

如果有、您將無法繼續。

步驟

1. 從左側導覽中選取*鏟斗*。
2. 從* Actions（操作）功能表中、選取*移除。

> Astra Control會先確保不會有使用儲存庫進行備份的排程原則、而且您要移除的儲存庫中沒有作用中的備份。

3. 輸入「移除」以確認動作。
4. 選擇*是、移除桶*。

如需詳細資訊、請參閱

- "使用Astra Control API"

管理儲存後端

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。如果Astra Control Center連接Cloud Insights 到VMware、您可以監控儲存容量和健全狀況詳細資料、包括效能。

如需如何使用Astra Control API管理儲存後端的指示、請參閱 "Astra Automation和API資訊"。

您可以完成下列與管理儲存後端相關的工作：

- "新增儲存後端"
- [檢視儲存後端詳細資料]
- [取消管理儲存後端]
- [更新儲存後端授權]
- [將節點新增至儲存後端叢集]
- [移除儲存後端]

檢視儲存後端詳細資料

您可以從儀表板或後端選項檢視儲存後端資訊。

在「儲存後端詳細資料」頁面的Astra Data Store中、您可以看到下列資訊：

- Astra Data Store叢集
  - 處理量、IOPS及延遲
  - 已用容量與總容量比較
- 針對每個Astra Data Store叢集Volume
  - 已用容量與總容量比較
  - 處理量

從儀表板檢視儲存後端詳細資料

步驟

1. 從左側導覽中選取*儀表板*。

2. 請檢閱顯示狀態的儲存後端區段：
    - 不健全：儲存設備未處於最佳狀態。這可能是因為延遲問題、或是應用程式因為容器問題而降級。
    - 一切正常：儲存設備已經過管理、並處於最佳狀態。
    - 探索：儲存設備已被探索、但未由Astra Control管理。

從後端選項檢視儲存後端詳細資料

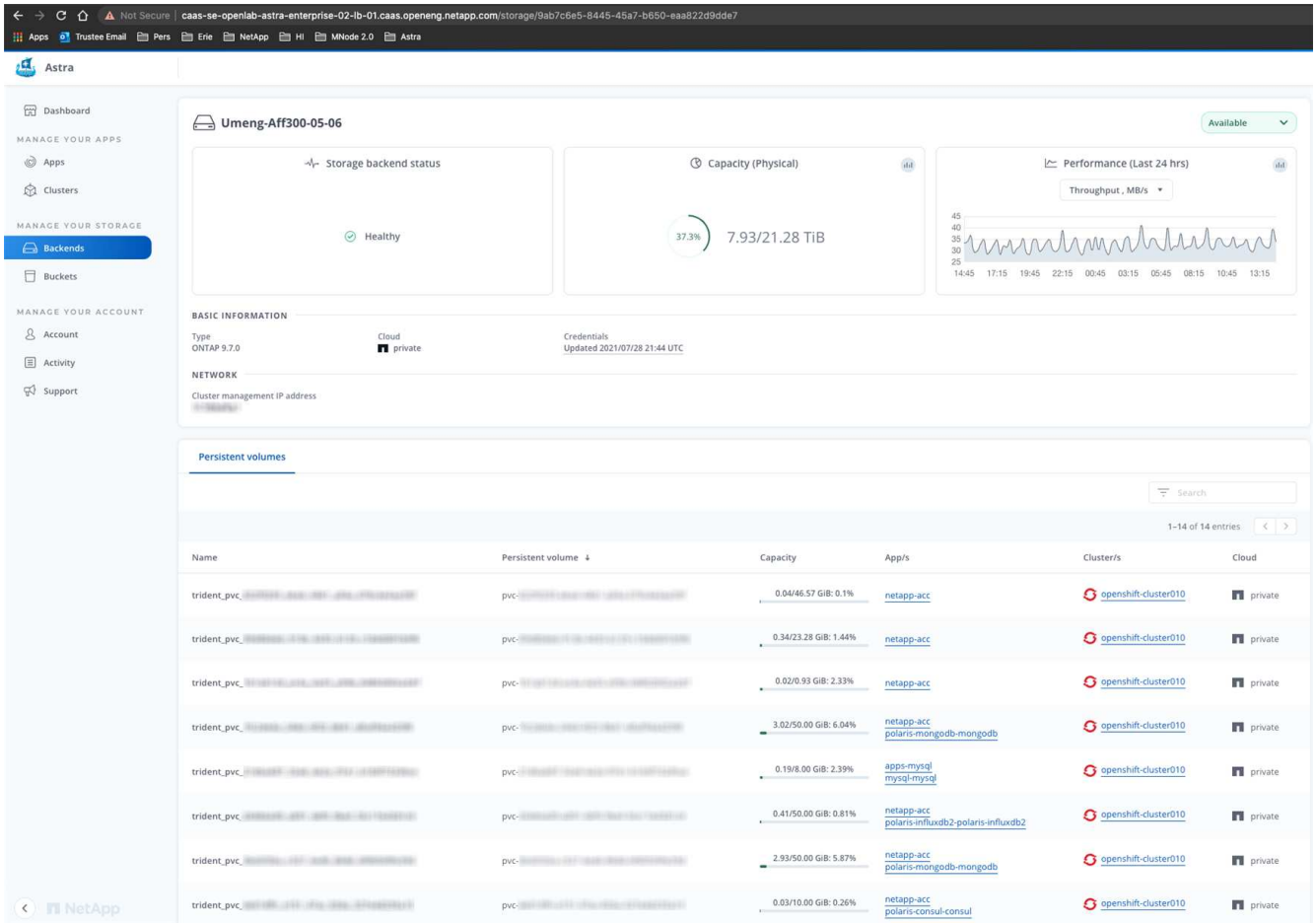
檢視後端健全狀況、容量和效能（IOPS處理量和/或延遲）的相關資訊。

透過Cloud Insights 連線至功能區、您可以看到Kubernetes應用程式所使用的磁碟區、這些磁碟區儲存在選定的儲存後端上。

步驟

1. 在左側導覽區域中、選取*後端*。
2. 選取儲存後端。

> 如果您連線至NetApp Cloud Insights 解決方案、Cloud Insights 則會在「後端」頁面上顯示來自於《》的資料摘錄。



3. 若要直接移至Cloud Insights 「不顯示」、請選取Cloud Insights 「指標」影像旁的*「不顯示」圖示。

取消管理儲存後端

您可以取消管理後端。

步驟

1. 從左側導覽中選取*後端*。
2. 選取儲存後端。
3. 從「動作」欄的「選項」功能表中、選取「取消管理」。
4. 輸入「unManage（取消管理）」以確認此動作。
5. 選擇*是、取消管理儲存後端*。

移除儲存後端

您可以移除不再使用的儲存後端。您可能會想要這麼做、讓您的組態保持簡單且最新狀態。

> 如果您要移除Astra Data Store後端、則vCenter不得建立該後端。

您需要的產品

- 確保儲存後端未受管理。
- 確保儲存後端沒有任何與Astra Data Store叢集相關的磁碟區。

步驟

1. 從左側導覽中選取*後端*。
2. 如果管理後端、請取消管理。
   a. 選擇*託管*。
   b. 選取儲存後端。
   c. 從*「Actions」（動作）選項中、選取「UnManage」（取消管理）*。
   d. 輸入「unManage（取消管理）」以確認此動作。
   e. 選擇*是、取消管理儲存後端*。
3. 選擇*已探索*。
   a. 選取儲存後端。
   b. 從* Actions （操作）選項中選擇*移除*。
   c. 輸入「移除」以確認動作。
   d. 選擇*是、移除儲存後端*。

更新儲存後端授權

您可以更新Astra Data Store儲存後端的授權、以支援更大的部署或增強功能。

您需要的產品

- 已部署並管理的Astra Data Store儲存後端

- Astra Data Store授權檔案（請聯絡您的NetApp銷售代表以購買Astra Data Store授權）

步驟

1. 從左側導覽中選取*後端*。
2. 選取儲存後端的名稱。
3. 在*基本資訊*下、您可以看到安裝的授權類型。

   如果您將游標暫留在授權資訊上、則會出現一個快顯視窗、內含更多資訊、例如過期和權利資訊。

4. 在「授權」下、選取授權名稱旁的編輯圖示。
5. 在「更新授權」頁面中、執行下列其中一項：

授權狀態	行動
Astra Data Store至少新增一項授權。	從清單中選取授權。
Astra Data Store未新增授權。	a. 選取*「Add*（新增*）」按鈕。 b. 選取要上傳的授權檔案。 c. 選擇*「Add*」（新增*）上傳授權檔案。

6. 選擇*更新*。

將節點新增至儲存後端叢集

您可以將節點新增至Astra Data Store叢集、最多可新增至Astra Data Store安裝的授權類型所支援的節點數。

您需要的產品

- 已部署且獲得授權的Astra Data Store儲存後端
- 您已在Astra控制中心新增Astra Data Store軟體套件
- 一或多個新節點、以新增至叢集

步驟

1. 從左側導覽中選取*後端*。
2. 選取儲存後端的名稱。
3. 在「基本資訊」下、您可以看到此儲存後端叢集中的節點數目。
4. 在*節點*下、選取節點數旁的編輯圖示。
5. 在「新增節點」頁面中、輸入新節點的相關資訊：
   a. 為每個節點指派節點標籤。
   b. 執行下列其中一項：
      - 如果您想要Astra Data Store根據授權一律使用最大可用節點數、請啟用「永遠使用最多允許的節點數」核取方塊。
      - 如果您不希望Astra Data Store永遠使用最大可用節點數、請選取所需使用的總節點數。

c. 如果您部署Astra Data Store並啟用Protection Domain、請將新節點指派給Protection Domain。

6. 選擇*下一步*。
7. 輸入每個新節點的IP位址和網路資訊。為單一新節點輸入單一IP位址、或為多個新節點輸入IP位址集區。

   如果Astra Data Store可以使用部署期間所設定的IP位址、您就不需要輸入任何IP位址資訊。

8. 選擇*下一步*。
9. 檢閱新節點的組態。
10. 選取*新增節點*。

如需詳細資訊、請參閱

- "使用Astra Control API"

監控及保護基礎架構

您可以設定多項選用設定、以增強Astra Control Center體驗。如果您執行Astra Control Center的網路需要Proxy才能連線至網際網路（將支援套件上傳至NetApp支援網站或建立Cloud Insights 連線至鏈接）、您應該在Astra Control Center中設定Proxy伺服器。若要監控並深入瞭解您的完整基礎架構、請建立與NetApp Cloud Insights的連結。若要從Astra Control Center監控的系統收集Kubernetes事件、請新增Fludd連線。

新增Proxy伺服器

如果您執行Astra Control Center的網路需要Proxy才能連線至網際網路（將支援套件上傳至NetApp支援網站或建立Cloud Insights 連線至鏈接）、您應該在Astra Control Center中設定Proxy伺服器。

> Astra Control Center不會驗證您為Proxy伺服器輸入的詳細資料。請確認輸入正確的值。

步驟

1. 使用具有*管理*/*擁有者*權限的帳戶登入Astra Control Center。
2. 選擇*帳戶*>*連線*。
3. 從下拉式清單中選取*「Connect*」以新增Proxy伺服器。

4. 輸入Proxy伺服器名稱或IP位址及Proxy連接埠號碼。
5. 如果您的Proxy伺服器需要驗證、請選取核取方塊、然後輸入使用者名稱和密碼。
6. 選擇*連接*。

結果

如果您輸入的代理資訊已儲存、則「帳戶>*連線*」頁面的「* HTTP Proxy*」區段會指出其已連線、並顯示伺服器名稱。

編輯**Proxy**伺服器設定

您可以編輯Proxy伺服器設定。

步驟

1. 使用具有*管理*/*擁有者*權限的帳戶登入Astra Control Center。
2. 選擇*帳戶*>*連線*。
3. 從下拉式清單中選取*編輯*以編輯連線。
4. 編輯伺服器詳細資料和驗證資訊。
5. 選擇*保存*。

停用**Proxy**伺服器連線

您可以停用Proxy伺服器連線。在停用之前、系統會先警告您、否則可能會對其他連線造成潛在的中斷。

步驟

1. 使用具有*管理*/*擁有者*權限的帳戶登入Astra Control Center。
2. 選擇*帳戶*>*連線*。
3. 從下拉式清單中選取*「Disconnect*（中斷連線）」以停用連線。
4. 在開啟的對話方塊中、確認作業。

連線**Cloud Insights** 至

若要監控並深入瞭解完整的基礎架構、請將NetApp Cloud Insights 知識與Astra Control Center執行個體連結起來。包含在您的Astra Control Center授權中。Cloud Insights

應可從Astra Control Center使用的網路存取、或透過Proxy伺服器間接存取。Cloud Insights

當Astra Control Center連線Cloud Insights 至不實時、就會建立一個擷取單元Pod。此Pod可從Astra Control Center管理的儲存後端收集資料、並將資料推送到Cloud Insights此Pod需要8 GB RAM和2個CPU核心。

> 啟用Cloud Insights 完「支援不中斷連線」後、您可以在*後端*頁面上檢視處理量資訊、Cloud Insights 並在選取儲存後端後端後、從此處連線至「支援不中斷連線」。您也可以在「叢集」區段的*儀表板*上找到相關資訊、也可以從Cloud Insights 這裡連線至。

您需要的產品

- 具有*管理*/*擁有者*權限的Astra Control Center帳戶。
- 有效的Astra Control Center授權。
- 如果您執行Astra Control Center的網路需要Proxy才能連線至網際網路、則為Proxy伺服器。

> 如果您是Cloud Insights 不熟悉的人、請熟悉這些功能。請參閱 "本文檔 Cloud Insights"。

步驟

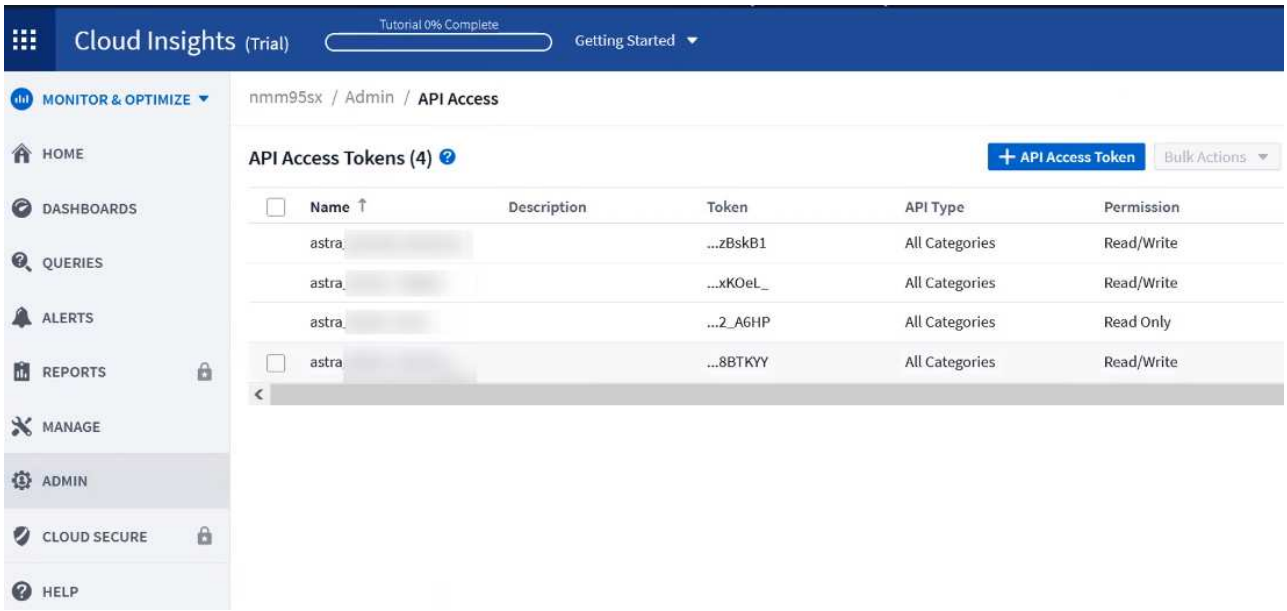
1. 使用具有*管理*/*擁有者*權限的帳戶登入Astra Control Center。
2. 選擇*帳戶*>*連線*。
3. 在下拉式清單中選擇*「Connect*（連線*）」顯示*「Disconnected（中斷連線）」的位置、以新增連線。

4. 輸入Cloud Insights 「不再使用API」權杖和租戶URL。租戶URL的格式如下：

```
https://<environment-name>.c01.cloudinsights.netapp.com/
```

當您取得Cloud Insights 不含功能的授權時、就會收到租戶URL。如果您沒有租戶URL、請參閱 "本文檔 Cloud Insights"。

a. 以取得 "API權杖"、登入Cloud Insights 您的URL。
b. 在支援區中、按一下「管理」>「* API存取*」、即可產生*讀取/寫入*和*唯讀* API存取權杖。Cloud Insights



c. 複製*唯讀*金鑰。您必須將其貼到Astra Control Center視窗中、才能啟用Cloud Insights 此功能的鏈路。如需讀取API存取權杖金鑰權限、請選取：資產、警示、擷取單位和資料收集。

d. 複製*讀取/寫入*金鑰。您需要將其貼到Astra Control Center * Connect Cloud Insights S還原*視窗中。如需讀取/寫入API存取權杖金鑰權限、請選取：資產、資料擷取、記錄擷取、擷取單位、 和資料收集：

> 我們建議您產生*唯讀*金鑰和*讀取/寫入*金鑰、而不要將相同的金鑰用於這兩種用途。根據預設、權杖過期期間設為一年。我們建議您保留預設選項、以便在權杖過期之前提供最長持續時間。如果您的權杖過期、遙測就會停止。

e. 將您從Cloud Insights 整個過程中複製的金鑰貼到Astra Control Center。

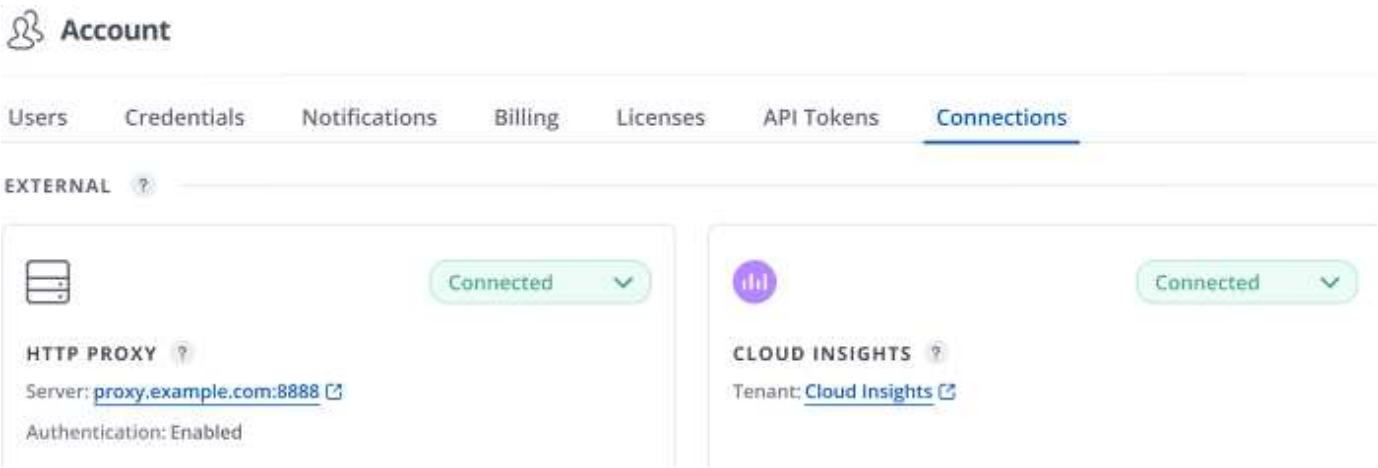
5. 選擇*連接*。

> 在您選取*連線之後、* Cloud Insights 帳戶*>*連線*頁面的*更新*區段中、連線狀態會變更為*擱置*。啟用連線並將狀態變更為「已連線」可能需要幾分鐘的時間。

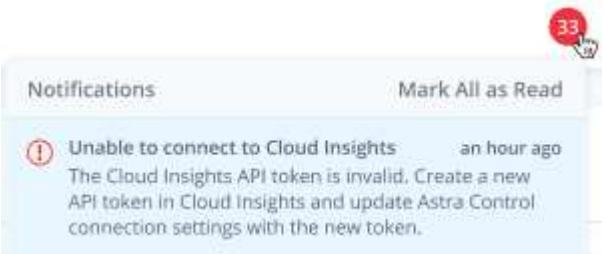
> 若要在Astra Control Center和Cloud Insights UI之間輕鬆來回、請確定您已登入這兩個項目。

檢視**Cloud Insights** 資料

如果連線成功、Cloud Insights 「帳戶>*連線*」頁面的* SURS*區段會指出連線狀態、並顯示租戶URL。您可以造訪Cloud Insights 景點、查看成功接收及顯示的資料。

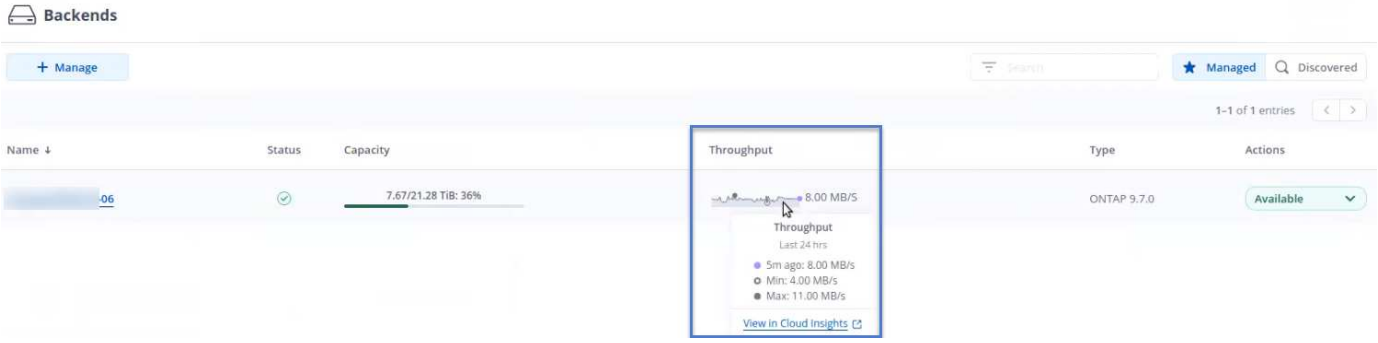


如果連線因為某種原因而失敗、狀態會顯示*失敗*。您可以在UI右上角的*通知*下找到失敗的原因。



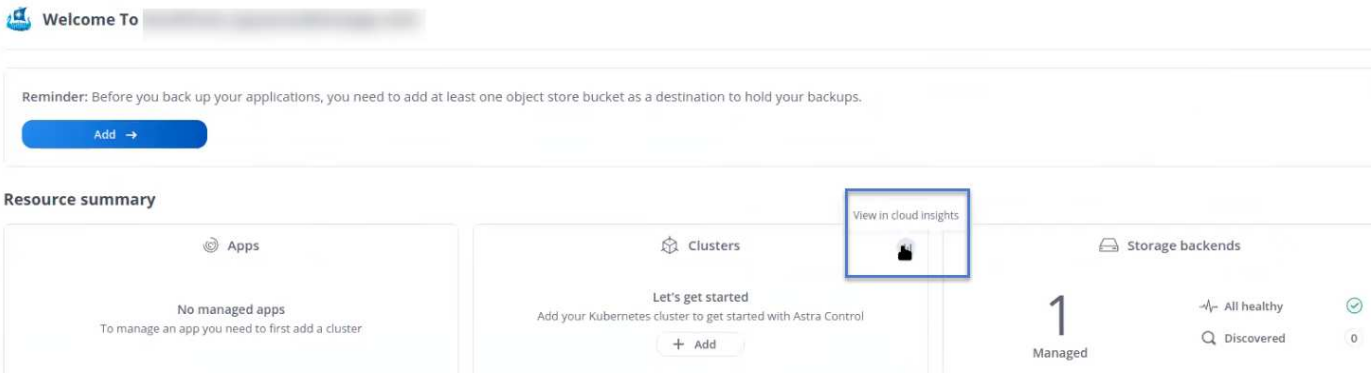
您也可以在*帳戶*>*通知*下找到相同的資訊。

從Astra Control Center、您可以在*後端*頁面上檢視處理量資訊、Cloud Insights 並在選擇儲存後端後端後、從此處連線
至



若要直接移至Cloud Insights 「不顯示」、請選取Cloud Insights 「指標」影像旁的*「不顯示」圖示。

您也可以在*儀表板*上找到相關資訊。



> 啟用Cloud Insights 完「支援不支援」連線後、如果您移除Astra Control Center中新增的後端、後端會停止向Cloud Insights 「支援不支援」回報。

編輯**Cloud Insights** 鏈接

您可以編輯Cloud Insights 此「不同步連線」。

> 您只能編輯API金鑰。若要變更Cloud Insights 此URL、我們建議您中斷Cloud Insights 連接此鏈接、並使用新的URL進行連線。

步驟

1. 使用具有*管理*/*擁有者*權限的帳戶登入Astra Control Center。
2. 選擇*帳戶*>*連線*。
3. 從下拉式清單中選取*編輯*以編輯連線。
4. 編輯Cloud Insights 「還原連線」設定。
5. 選擇*保存*。

停用**Cloud Insights** 鏈接

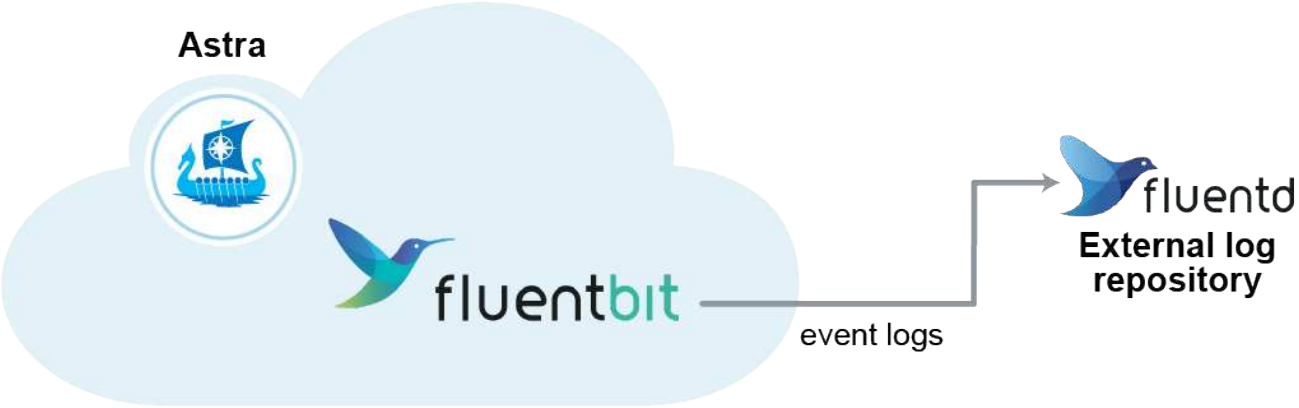
您可以停用Cloud Insights 由Astra Control Center管理的Kubernetes叢集的支援功能。停用Cloud Insights 此功能不會刪除已上傳至Cloud Insights 更新的遙測資料。

步驟

1. 使用具有*管理*/*擁有者*權限的帳戶登入Astra Control Center。
2. 選擇*帳戶*>*連線*。
3. 從下拉式清單中選取*「Disconnect*（中斷連線）」以停用連線。
4. 在開啟的對話方塊中、確認作業。在您確認操作之後、Cloud Insights 在*帳戶*>*連線*頁面上、顯示的「畫面」狀態會變更為*「待處理」*。狀態變更為*中斷連線*需要幾分鐘的時間。

連接至**Flud**

您可以將記錄（Kubernetes事件）從Astra Control Center傳送至您的Fluentd端點。Fluentd連線預設為停用。



只有來自託管叢集的事件記錄會轉送至Fluentd。

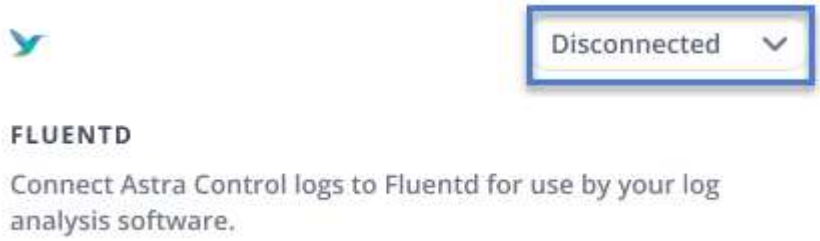
您需要的產品

- 具有*管理*/*擁有者*權限的Astra Control Center帳戶。
- Astra Control Center安裝並在Kubernetes叢集上執行。

Astra Control Center不會驗證您為Fluentd伺服器輸入的詳細資料。請確認輸入正確的值。

步驟

1. 使用具有*管理*/*擁有者*權限的帳戶登入Astra Control Center。
2. 選擇*帳戶*>*連線*。
3. 從顯示*中斷連線*的下拉式清單中選取*「Connect*（連線*）」以新增連線。



4. 輸入您的Fluentd伺服器的主機IP位址、連接埠號碼和共用金鑰。
5. 選擇*連接*。

結果

如果您為Fluentd伺服器輸入的詳細資料已儲存、則「帳戶>*連線*」頁面的「變動」區段會指出該資料已連線。現在您可以造訪您所連線的Fluentd伺服器、並檢視事件記錄。

如果連線因為某種原因而失敗、狀態會顯示*失敗*。您可以在UI右上角的*通知*下找到失敗的原因。

您也可以在*帳戶*>*通知*下找到相同的資訊。

如果您在收集記錄時遇到問題、請登入您的工作節點、並確保記錄可在「/var/log/contains/」中使用。

編輯**Fluentd**連線

您可以編輯Astra Control Center執行個體的Fluentd連線。

步驟

1. 使用具有*管理*/*擁有者*權限的帳戶登入Astra Control Center。
2. 選擇*帳戶*>*連線*。
3. 從下拉式清單中選取*編輯*以編輯連線。
4. 變更Fluentd端點設定。
5. 選擇*保存*。

停用**Fluentd**連線

您可以停用Astra Control Center執行個體的Fluentd連線。

步驟

1. 使用具有*管理*/*擁有者*權限的帳戶登入Astra Control Center。
2. 選擇*帳戶*>*連線*。
3. 從下拉式清單中選取*「Disconnect*（中斷連線）」以停用連線。
4. 在開啟的對話方塊中、確認作業。

取消管理應用程式和叢集

從Astra Control Center移除不再需要管理的任何應用程式或叢集。

取消管理應用程式

停止管理不再想從Astra Control Center備份、快照或複製的應用程式。

- 任何現有的備份與快照都會刪除。
- 應用程式與資料仍可繼續使用。

步驟

1. 從左側導覽列選取*應用程式*。
2. 選取您不想再管理的應用程式核取方塊。
3. 從*操作*功能表中、選取*取消管理*。
4. 輸入「unManage（取消管理）」以確認。
5. 確認您要取消管理應用程式、然後選取*是、取消管理應用程式*。

結果

Astra Control Center停止管理應用程式。

取消管理叢集

取消管理不再想從Astra Control Center管理的叢集。

- 此動作可防止您的叢集受到Astra Control Center的管理。它不會對叢集的組態進行任何變更、也不會刪除叢集。
- Trident不會從叢集解除安裝。 "瞭解如何解除安裝Trident"。

> 在取消管理叢集之前、您應該取消管理與叢集相關的應用程式。

步驟

1. 從左側導覽列選取*叢集*。
2. 選取您不再想在Astra Control Center中管理的叢集核取方塊。
3. 從「動作」欄的「選項」功能表中、選取「取消管理」。
4. 確認您要取消管理叢集、然後選取*是、取消管理叢集*。

結果

叢集的狀態會變更為*移除*、之後叢集會從*叢集*頁面移除、而且不再由Astra Control Center管理。

> *如果Astra Control Center和Cloud Insights Sfunk*未連線、取消叢集管理會移除所有安裝用於傳送遙測資料的資源。*如果Astra Control Center和Cloud Insights Sfunsare connected *、則取消管理叢集只會刪除「fluentbit"（變動位元）」和「EVENT-Outer（事件-輸出程式）」等Pod。

升級Astra Control Center

若要升級Astra Control Center、請從NetApp支援網站下載安裝套件、並完成以下指示、以升級您環境中的Astra Control Center元件。您可以使用此程序、在連線網際網路或無線環境中升級Astra Control Center。

您需要的產品

- "開始升級之前、請確保環境仍符合Astra Control Center部署的最低需求"。
- 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服務都處於健全狀態且可用。

  OpenShift範例：

  ```
  oc get apiservices
  ```

- 登出您的Astra Control Center。

關於這項工作

Astra Control Center升級程序會引導您完成下列高層級步驟：

- 下載Astra Control Center套裝組合
- [解壓縮套件並變更目錄]
- [將映像新增至本機登錄]
- 安裝更新的Astra Control Center操作員
- 升級Astra Control Center
- [升級協力廠商服務（選用）]
- [驗證系統狀態]
- [設定入口以進行負載平衡]

請勿在整個升級過程中執行下列命令、以免刪除所有Astra Control Center Pod：「kubecl DELETE -f Astra _control_center_oper_deploy。yaml」

當排程、備份和快照未執行時、請在維護期間執行升級。

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

下載Astra Control Center套裝組合

1. 請從下載Astra Control Center升級套裝組合（「Astra控制中心-[版本].tar.gz」） "NetApp 支援網站"。
2. （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
```

解壓縮套件並變更目錄

1. 擷取影像：

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

2. 變更至Astra目錄。

```
cd astra-control-center-[version]
```

將映像新增至本機登錄

1. 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Docker登錄：

   ```
   docker login [your_registry_path]
   ```

   b. 將影像載入Docker。

   c. 標記影像。

   d. [Subforte_image_local_register_push]將映像推送到本機登錄。

   ```
   export REGISTRY=[your_registry_path]
   for astraImageFile in $(ls images/*.tar)
     # Load to local cache. And store the name of the loaded image
   trimming the 'Loaded images: '
     do astraImage=$(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
   ```

安裝更新的**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. 將下列值新增至「env」區段：

```
- name: ACCOP_HELM_UPGRADETIMEOUT
  value: 300m
```

```
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"
        - name: ACCOP_HELM_UPGRADETIMEOUT
          value: 300m
        image: [your_registry_path]/acc-operator:[version x.y.z]
        imagePullPolicy: IfNotPresent
      imagePullSecrets: []
```

2. 安裝更新的Astra Control Center操作員：

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

回應範例：

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

升級Astra Control Center

1. 編輯Astra Control Center自訂資源（CR）（「Astra control_center_min.yaml」）、並將Astra版本（「'astraVersion」位於「Pec」內）編號變更為最新版本：

```
kubectl edit acc -n [netapp-acc or custom namespace]
```

> 您的登錄路徑必須符合您在中推送映像的登錄路徑 上一步。

2. 在Astra Control Center CR的「Pec」內的「additionalValues」中新增下列行：

```
additionalValues:
    nautilus:
      startupProbe:
        periodSeconds: 30
        failureThreshold: 600
```

3. 執行下列其中一項：

   a. 如果您沒有自己的IngressController或Ingressal、而且一直使用Astra Control Center搭配Traefik閘道做為負載平衡器類型服務、而且想要繼續進行該設定、請指定另一個欄位「ingressType」（如果尚未出現）、並將其設為「AccTraefik」。

   ```
   ingressType: AccTraefik
   ```

   b. 如果您想要切換至預設Astra控制中心一般入侵部署、請提供您自己的入侵控制器/入侵設定（TLS終止等）、開啟通往Astra控制中心的路由、並將「擷取類型」設為「一般」。

   ```
   ingressType: Generic
   ```

   > 如果您省略此欄位、程序就會變成一般部署。如果您不想要一般部署、請務必新增欄位。

4. （可選）驗證Pod是否終止並再次可用：

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

5. 等待Astra狀態條件指示升級已完成且準備就緒：

   ```
   kubectl get -o yaml -n [netapp-acc or custom namespace]
   astracontrolcenters.astra.netapp.io astra
   ```

   回應：

   ```
   conditions:
     - lastTransitionTime: "2021-10-25T18:49:26Z"
       message: Astra is deployed
       reason: Complete
       status: "True"
       type: Ready
     - lastTransitionTime: "2021-10-25T18:49:26Z"
       message: Upgrading succeeded.
       reason: Complete
       status: "False"
       type: Upgrading
   ```

6. 重新登入、確認所有託管叢集和應用程式仍存在且受到保護。
7. 如果營運者未更新Cert管理程式、請升級協力廠商服務、接著再升級。

升級協力廠商服務（選用）

在先前的升級步驟中、不會升級協力廠商服務Traefik和Cert Manager。您可以選擇使用本文所述的程序來升級、或是在系統需要時保留現有的服務版本。

- * Traefik*：依預設、Astra Control Center會管理Traefik部署的生命週期。將「externalTraefik」設為「假」（預設）表示系統中不存在外部Traefik、而Traefik則由Astra Control Center安裝及管理。在這種情況下、「externalTraefik」設定為「假」。

  另一方面、如果您有自己的Traefik部署、請將「externalTraefik」設為「true」。在這種情況下、除非將「shouldUpgrade」設為「true」、否則您將維持部署、Astra Control Center將不會升級客戶需求日。

- 認證管理程式：依預設、Astra Control Center會安裝認證管理程式（和CRD）、除非您將「externalCertManager」設為「true」。將「shouldUpgrade」設為「true」、讓Astra Control Center升級CRD。

如果符合下列任一條件、就會升級Traefik：

- 外部Traefik：假或
- externalTraefik：真實且應該升級：真。

步驟

1. 編輯「acc：

```
kubectl edit acc -n [netapp-acc or custom namespace]
```

2. 視需要將「externalTraefik」欄位和「shouldUpgrade」欄位變更為「true」或「假」。

```
crds:
    externalTraefik: false
    externalCertManager: false
    shouldUpgrade: false
```

驗證系統狀態

1. 登入Astra Control Center。
2. 確認您所有的託管叢集和應用程式仍存在且受到保護。

設定入口以進行負載平衡

您可以設定Kubernetes入口物件來管理外部服務存取、例如叢集中的負載平衡。

- 預設升級使用一般入口部署。在此情況下、您也需要設定入口控制器或入口資源。
- 如果您不想要入站控制器並想保留現有的內容、請將「擷取類型」設為「AccTraefik」。

如需有關「負載平衡器」和入口服務類型的其他詳細資料、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

- 在CR規格中、
  - 如果出現「crd.externalTraefik」、則應設定為「假」或
  - 如果“crd.externalTraefik”是真的，那麼“crd.doedUpgrade（升級）”也應該是真的。
- 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v1.21之間、甚至包括在內。

適用於**Nginvin**像 控制器的步驟

1. 使用現有的秘密「安全測試證書」、或建立類型的機密 "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. 在「NetApp-acc」（或自訂命名）命名空間中、針對已過時或新架構部署入口資源：
   a. 對於已過時的架構、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: nginx
spec:
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: [ACC address]
    http:
      paths:
      - backend:
        serviceName: traefik
        servicePort: 80
        pathType: ImplementationSpecific
```

b. 如需新架構、請遵循下列範例：

```
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 address>
    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
```

驗證入口設定

您可以在繼續之前驗證入口設定。

1. 確認Traefik已從負載平衡器變更為「clusterIP（叢集IP）」：

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

2. 驗證Traefik中的路由：

```
Kubectl get ingressroute ingressroutetls -n [netapp-acc or custom
namespace]
-o yaml | grep "Host("
```

結果應為空白。

解除安裝Astra Control Center

如果您要從試用版升級至完整版產品、可能需要移除Astra Control Center元件。若要移除Astra Control Center和Astra Control Center操作員、請依序執行本程序中所述的命令。

如果您對解除安裝有任何問題、請參閱 [疑難排解解除安裝問題]。

您需要的產品

- 使用Astra Control Center UI取消管理所有項目 "叢集"。

步驟

1. 刪除Astra Control Center。下列範例命令是根據預設安裝而來。如果您進行自訂組態、請修改命令。

```
kubectl delete -f astra_control_center_min.yaml -n netapp-acc
```

結果：

```
astracontrolcenter.astra.netapp.io "astra" deleted
```

2. 使用下列命令刪除「NetApp-acc」命名空間：

```
kubectl delete ns netapp-acc
```

結果：

```
namespace "netapp-acc" deleted
```

3. 使用下列命令刪除Astra Control Center作業系統元件：

```
kubectl delete -f astra_control_center_operator_deploy.yaml
```

結果：

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

疑難排解解除安裝問題

請使用下列因應措施來解決您在解除安裝Astra Control Center時遇到的任何問題。

解除安裝**Astra Control Center**無法清除受管理叢集上的監控操作員**Pod**

如果在卸載Astra Control Center之前未取消管理叢集、您可以使用下列命令手動刪除NetApp監控命名空間和命名空間中的Pod：

步驟

1. 刪除「acc監控」代理程式：

   ```
   kubectl delete agents acc-monitoring -n netapp-monitoring
   ```

   結果：

   ```
   agent.monitoring.netapp.com "acc-monitoring" deleted
   ```

2. 刪除命名空間：

   ```
   kubectl delete ns netapp-monitoring
   ```

結果：

```
namespace "netapp-monitoring" deleted
```

3. 確認移除的資源：

```
kubectl get pods -n netapp-monitoring
```

結果：

```
No resources found in netapp-monitoring namespace.
```

4. 確認監控代理程式已移除：

```
kubectl get crd|grep agent
```

結果範例：

```
agents.monitoring.netapp.com                    2021-07-21T06:08:13Z
```

5. 刪除自訂資源定義（CRD）資訊：

```
kubectl delete crds agents.monitoring.netapp.com
```

結果：

```
customresourcedefinition.apiextensions.k8s.io
"agents.monitoring.netapp.com" deleted
```

解除安裝**Astra Control Center**無法清除**Traefik CRD**

您可以手動刪除Traefik客戶需求日。客戶需求日是全域資源、刪除這些資源可能會影響叢集上的其他應用程式。

步驟

1. 列出叢集上安裝的Traefik客戶需求日：

```
kubectl get crds |grep -E 'traefik'
```

回應

```
ingressroutes.traefik.containo.us             2021-06-23T23:29:11Z
ingressroutetcps.traefik.containo.us          2021-06-23T23:29:11Z
ingressrouteudps.traefik.containo.us          2021-06-23T23:29:12Z
middlewares.traefik.containo.us               2021-06-23T23:29:12Z
middlewaretcps.traefik.containo.us            2021-06-23T23:29:12Z
serverstransports.traefik.containo.us         2021-06-23T23:29:13Z
tlsoptions.traefik.containo.us                2021-06-23T23:29:13Z
tlsstores.traefik.containo.us                 2021-06-23T23:29:14Z
traefikservices.traefik.containo.us           2021-06-23T23:29:15Z
```

2. 刪除客戶需求日：

```
kubectl delete crd ingressroutes.traefik.containo.us
ingressroutetcps.traefik.containo.us
ingressrouteudps.traefik.containo.us middlewares.traefik.containo.us
serverstransports.traefik.containo.us tlsoptions.traefik.containo.us
tlsstores.traefik.containo.us traefikservices.traefik.containo.us
middlewaretcps.traefik.containo.us
```

如需詳細資訊、請參閱

- "解除安裝的已知問題"