



# 管理工作流程

## Astra Automation

NetApp  
December 01, 2023

# 目錄

管理工作流程 .....	1
開始之前 .....	1
應用程式控制 .....	2
應用程式保護 .....	5
複製及還原應用程式 .....	13
命名空間 .....	18
支援 .....	20

# 管理工作流程

## 開始之前

您可以使用這些工作流程來管理Astra託管叢集中的應用程式。



NetApp可隨時擴充及強化這些工作流程、因此您應定期檢閱。

### 一般準備

使用任何Astra工作流程之前、請務必先檢閱 "[準備好使用工作流程](#)"。

### 工作流程類別

管理工作流程會依不同類別進行組織、以便更容易找到您想要的工作流程。

類別	說明
應用程式控制	這些工作流程可讓您控制託管及未受管理的應用程式。您可以列出應用程式、以及建立和移除託管應用程式。
應用程式保護	您可以使用這些工作流程、透過快照和備份來保護託管應用程式。
複製及還原應用程式	這些工作流程說明如何複製及還原託管應用程式。
支援	有幾種工作流程可用來偵錯及支援您的應用程式、以及一般Kubernetes環境。

### 其他考量

使用管理工作流程時、還有幾項額外考量。

#### 複製應用程式

複製應用程式時、需要考量一些事項。下列參數是Json輸入的一部分。

#### 來源叢集識別碼

的價值 `sourceClusterID` 永遠識別安裝原始應用程式的叢集。

#### 叢集識別碼

的價值 `clusterID` 識別要安裝新應用程式的叢集。

- 在同一個叢集內複製時、`clusterID` 和 `sourceClusterID` 具有相同的值。
- 在叢集之間進行複製時、這兩個值會有所不同 `clusterID` 應該是目標叢集ID。

#### 命名空間

◦ `namespace` 值必須與原始來源應用程式不同。此外、複本的命名空間也不存在、Astra也會建立該名稱空間。

## 備份與快照

您可以選擇使用從現有備份或快照複製應用程式 `backupID` 或 `snapshotID` 參數。如果您未提供備份或快照、Astra會先建立應用程式的備份、然後從備份複製。

## 還原應用程式

以下是還原應用程式時的幾項考量事項。

- 還原應用程式與複製作業非常類似。
- 還原應用程式時、您必須提供備份或快照。

# 應用程式控制

## 列出應用程式

您可以列出目前由Astra管理的應用程式。您可以在尋找特定應用程式的快照或備份時執行此動作。

### 1.列出應用程式

執行下列REST API呼叫。

HTTP方法	路徑
取得	<code>/account/ {account_id} /k8s/v2/apps</code>

## 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
包括	查詢	否	選擇性地選取您要傳回回應中的值。

### Curl範例：傳回所有應用程式的所有資料

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

### Curl範例：傳回所有應用程式的名稱、ID和狀態

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps?include=name,id
,state' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

## Json輸出範例

```
{
  "items": [
    [
      "mysql",
      "4ee2b8fa-3696-4f32-8879-399792f477c3",
      "ready"
    ],
    [
      "postgresql",
      "3b984474-e5c9-4b64-97ee-cdeb9bcd212e",
      "ready"
    ],
  ],
  "metadata": {}
}
```

## 取得應用程式

您可以擷取所有描述單一應用程式的資源變數。

開始之前

您必須擁有想要擷取的應用程式ID。如有需要、您可以使用工作流程 ["列出應用程式"](#) 以找出應用程式。

### 1.取得應用程式

執行下列REST API呼叫。

HTTP方法	路徑
取得	/Accounts / {account_id} /k8s/v2/apps / {app_id}

### 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
應用程式ID	路徑	是的	要擷取的應用程式ID值。

**Curl範例：**傳回應用程式的所有資料

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps/<APP_ID>'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

## 管理應用程式

您可以在特定命名空間中、根據Astra已知的應用程式來建立託管應用程式。當Astra管理或定義應用程式時、您可以透過備份和快照來保護應用程式。

### 1. 選取命名空間

執行工作流程 ["列出命名空間"](#) 然後選取命名空間。

### 2. 選取叢集

執行工作流程 ["列出叢集"](#) 然後選取叢集。

### 3. 管理應用程式

執行下列REST API呼叫來管理應用程式。

HTTP方法	路徑
貼文	/Accounts / {account_id} /k8s/v2/apps

### 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
JSON	本文	是的	提供識別要管理之應用程式所需的參數。請參閱以下範例。

### JSON輸入範例

```
{
  "clusterID": "7ce83fba-6aa1-4e0c-a194-26e714f5eb46",
  "name": "subtext",
  "namespaceScopedResources": [{"namespace": "kube-matrix"}],
  "type": "application/astra-app",
  "version": "2.0"
}
```

### Curl範例：管理應用程式

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Content-Type: application/astra-app+json' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>' --data @JSONinput
```

## 取消管理應用程式

您可以在不再需要的情況下移除託管應用程式。移除託管應用程式也會刪除相關的排程。

開始之前

您必須擁有想要取消管理的應用程式ID。如有需要、您可以使用工作流程 ["列出應用程式"](#) 以找出應用程式。

應用程式的備份與快照不會在刪除時自動移除。如果您不再需要備份和快照、則應在移除應用程式之前先將其刪除。

### 1. 不受管理的應用程式

執行下列REST API呼叫以移除應用程式。

HTTP方法	路徑
刪除	/Accounts / {account_id} /k8s/v2/apps / {app_id}

### 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
應用程式ID	路徑	是的	識別要移除的應用程式。

### Curl範例：移除託管應用程式

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps/<APP_ID>'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

## 應用程式保護

### 列出快照

您可以列出針對特定應用程式所拍攝的快照。

開始之前

您必須擁有要列出快照的應用程式ID。如有需要、您可以使用工作流程 ["列出應用程式"](#) 以找出應用程式。

### 1. 列出快照

執行下列REST API呼叫、以列出快照。

HTTP方法	路徑
取得	/Accounts / {account_id} /k8s/v1/apps/ {app_id} /應用程式快照

### 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
應用程式ID	路徑	是的	識別擁有所列快照的應用程式。
數	查詢	否	如果 count=true 快照數量會包含在回應的中繼資料區段中。

### Curl範例：傳回應用程式的所有快照

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appSnaps' --header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

### Curl範例：傳回應用程式和計數的所有快照

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appSnaps?count=true' --header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

### Json輸出範例



```

{
  "items": [
    {
      "type": "application/astra-appSnap",
      "version": "1.1",
      "id": "1ce34da4-bb0a-4926-b925-4a5d85dda8c2",
      "hookState": "success",
      "metadata": {
        "createdBy": "a530e865-23e8-4e2e-8020-e92c419a3867",
        "creationTimestamp": "2022-10-30T22:44:20Z",
        "modificationTimestamp": "2022-10-30T22:44:20Z",
        "labels": []
      },
      "snapshotAppAsset": "0ebfe3f8-40ed-4bdc-88c4-2144fbda85a0",
      "snapshotCreationTimestamp": "2022-10-30T22:44:33Z",
      "name": "snapshot-david-1",
      "state": "completed",
      "stateUnready": []
    }
  ],
  "metadata": {}
}

```

## 列出備份

您可以列出已為特定應用程式建立的備份。

開始之前

您必須擁有想要列出備份的應用程式ID。如有需要、您可以使用工作流程 ["列出應用程式"](#) 以找出應用程式。

### 1. 列出備份

執行下列REST API呼叫。

HTTP方法	路徑
取得	/Accounts/ {account_id} /k8s/v1/apps/ {app_id} /appBackups

### 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
應用程式ID	路徑	是的	識別擁有所列備份的託管應用程式。

## Curl範例：傳回應用程式的所有備份

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appBackups' --header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

## Json輸出範例

```
{
  "items": [
    {
      "type": "application/astra-appBackup",
      "version": "1.1",
      "id": "8edeb4a4-fd8b-4222-a559-1013145b28fc",
      "name": "backup-david-oct28-1",
      "bucketID": "a443e58f-59bd-4d45-835a-1bc7813f659a",
      "snapshotID": "dfe237cb-57b7-4576-af4d-00ba3a8f2828",
      "state": "completed",
      "stateUnready": [],
      "hookState": "success",
      "totalBytes": 205219132,
      "bytesDone": 205219132,
      "percentDone": 100,
      "metadata": {
        "labels": [
          {
            "name": "astra.netapp.io/labels/read-only/triggerType",
            "value": "backup"
          }
        ],
        "creationTimestamp": "2022-10-28T21:58:37Z",
        "modificationTimestamp": "2022-10-28T21:58:55Z",
        "createdBy": "a530e865-23e8-4e2e-8020-e92c419a3867"
      }
    }
  ],
  "metadata": {}
}
```

## 建立應用程式的快照

您可以為特定應用程式建立快照。

開始之前

您必須擁有想要建立快照的應用程式ID。如有需要、您可以使用工作流程 ["列出應用程式"](#) 以找出應用程式。

### 1. 建立快照

執行下列REST API呼叫。

HTTP方法	路徑
貼文	/Accounts / {account_id} /k8s/v1/apps/ {app_id} /應用程式快照

其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
應用程式ID	路徑	是的	識別要建立快照的託管應用程式。
JSON	本文	是的	提供快照的參數。請參閱以下範例。

### JSON輸入範例

```
{
  "type": "application/astra-appSnap",
  "version": "1.1",
  "name": "snapshot-david-1"
}
```

### Curl範例：建立應用程式的快照

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appSnap
ps' --header 'Content-Type: application/astra-appSnap+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --data
@JSONinput
```

### 為應用程式建立備份

您可以為特定應用程式建立備份、然後使用備份來還原或複製應用程式。

開始之前

您必須擁有要備份的應用程式ID。如有需要、您可以使用工作流程 ["列出應用程式"](#) 以找出應用程式。

## 1. 建立備份

執行下列REST API呼叫。

HTTP方法	路徑
貼文	/Accounts / {account_id} /k8s/v1/apps/ {app_id} /appBackups

### 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
應用程式ID	路徑	是的	識別要建立備份的應用程式。
JSON	本文	是的	提供備份參數。請參閱以下範例。

### JSONN輸入範例

```
{
  "type": "application/astra-appBackup",
  "version": "1.1",
  "name": "backup-david-1"
}
```

### Curl範例：為應用程式建立備份

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appBackups' --header 'Content-Type: application/astra-appBackup+json' --header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --data @JSONinput
```

## 刪除快照

您可以刪除與應用程式相關的快照。

開始之前

您必須具備下列條件：

- 擁有快照的應用程式ID。如有需要、您可以使用工作流程 ["列出應用程式"](#) 以找出應用程式。
- 您要刪除的快照ID。如有需要、您可以使用工作流程 ["列出快照"](#) 以找出快照。

## 1. 刪除快照

執行下列REST API呼叫。

HTTP方法	路徑
刪除	/accounts/{aper_id}/k8s/v1/apps/ {app_id} /應用程式快照/ {appsnap_id}

### 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
應用程式ID	路徑	是的	識別擁有快照的託管應用程式。
Snapshot ID	路徑	是的	識別要刪除的快照。

### Curl範例：刪除應用程式的單一快照

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appSnapshots/<SNAPSHOT_ID>' --header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

## 刪除備份

您可以刪除與應用程式相關的備份。

開始之前

您必須具備下列條件：

- 擁有備份的應用程式ID。如有需要、您可以使用工作流程 ["列出應用程式"](#) 以找出應用程式。
- 您要刪除的備份ID。如有需要、您可以使用工作流程 ["列出備份"](#) 以找出快照。

## 1. 刪除備份

執行下列REST API呼叫。



您可以使用選用的要求標頭強制刪除失敗的備份、如下所述。

HTTP方法	路徑
刪除	/accounts/{aper_id}/k8s/v1/apps/ {app_id} /appBackups/{appBackup_id}

### 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
應用程式ID	路徑	是的	識別擁有備份的託管應用程式。
備份ID	路徑	是的	識別要刪除的備份。
強制刪除	標頭	否	用於強制刪除失敗的備份。

### Curl範例：刪除應用程式的單一備份

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appBackups/<BACKUP_ID>' --header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

### Curl範例：使用force選項刪除應用程式的單一備份

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appBackups/<BACKUP_ID>' --header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --header 'Force-Delete: true'
```

## 建立資料保護原則

您可以根據一或多個排程來建立資料保護原則。

### 1. 選取應用程式

執行工作流程 ["列出應用程式"](#) 並選取所需的應用程式。

### 2. 建立保護機制

執行下列REST API呼叫、為特定應用程式建立保護原則。

HTTP方法	路徑
貼文	/Accounts / {account_id} /k8s/v1/apps/ {app_id} /排程

### JSONN輸入範例

```
{
  "type": "application/astra-schedule",
  "version": "1.3",
  "name": "Backup Schedule",
  "enabled": "true",
  "granularity": "monthly",
  "minute": "0",
  "hour": "0",
  "dayOfMonth": "1",
  "snapshotRetention": "12",
  "backupRetention": "12"
}
```

## Curl範例

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/schedules' --header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
--data @JSONinput
```

# 複製及還原應用程式

## 複製應用程式

您可以透過複製現有應用程式來建立新的應用程式。

開始之前

請注意下列關於此工作流程的資訊：

- 未使用應用程式備份或快照
- 複製作業會在同一個叢集內執行
- 新應用程式會放置在不同的命名空間中



若要將應用程式複製到不同的叢集、您需要更新 `clusterId` Json輸入中的參數（視您的環境而定）。

### 1. 選取要複製的應用程式

執行工作流程 ["列出應用程式"](#) 並選取您要複製的應用程式。用於複製應用程式的REST呼叫需要幾個資源值。

### 2. 複製應用程式

執行下列REST API呼叫來複製應用程式。

HTTP方法	路徑
貼文	/Accounts / {account_id} /k8s/v2/apps

### 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
JSON	本文	是的	提供複製應用程式的參數。請參閱以下範例。

### JSON輸入範例

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "name": "mysql-clone",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "mysql-ns",
  "sourceAppID": "e591ee59-ea90-4a9f-8e6c-d2b6e8647096"
}
```

### Curl範例：複製應用程式

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Content-Type: application/astra-app+json' --header '*/*' --header
'Authorization: Bearer <API_TOKEN>' --data @JSONinput
```

## 從快照複製應用程式

您可以從快照複製新的應用程式、以建立新的應用程式。

開始之前

請注意下列關於此工作流程的資訊：

- 使用應用程式快照
- 複製作業會在同一個叢集內執行



若要將應用程式複製到不同的叢集、您需要更新 clusterId Json輸入中的參數（視您的環境而定）。



### 1. 選取要複製的應用程式

執行工作流程 "列出應用程式" 並選取您要複製的應用程式。用於複製應用程式的REST呼叫需要幾個資源值。

### 2. 選取要使用的快照

執行工作流程 "列出快照" 然後選取您要使用的快照。

### 3. 複製應用程式

執行下列REST API呼叫。

HTTP方法	路徑
貼文	/Accounts / {account_id} /k8s/v2/apps

### 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
JSON	本文	是的	提供複製應用程式的參數。請參閱以下範例。

### JSON輸入範例

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "name": "mysql-clone2",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "mysql",
  "snapshotID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"
}
```

### Curl範例：從快照複製應用程式

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Content-Type: application/astra-app+json' --header '*/*' --header
'Authorization: Bearer <API_TOKEN>' --data @JSONinput
```

### 從備份複製應用程式

您可以從備份複製新的應用程式、以建立新的應用程式。

開始之前

請注意下列關於此工作流程的資訊：

- 使用應用程式備份
- 複製作業會在同一個叢集內執行



若要將應用程式複製到不同的叢集、您需要更新 `clusterId` JSON輸入中的參數（視您的環境而定）。

### 1. 選取要複製的應用程式

執行工作流程 "列出應用程式" 並選取您要複製的應用程式。用於複製應用程式的REST呼叫需要幾個資源值。

### 2. 選取要使用的備份

執行工作流程 "列出備份" 並選取您要使用的備份。

### 3. 複製應用程式

執行下列REST API呼叫。

HTTP方法	路徑
貼文	/Accounts / {account_id} /k8s/v2/qpps

### 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
JSON	本文	是的	提供複製應用程式的參數。請參閱以下範例。

### JSON輸入範例

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "name": "mysql-clone3",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "mysql",
  "backupID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"
}
```

## Curl範例：從備份複製應用程式

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Content-Type: application/astra-app+json' --header '*/*' --header
'Authorization: Bearer <API_TOKEN>' --data @JSONinput
```

## 從備份還原應用程式

您可以從備份建立新的應用程式來還原應用程式。

### 1. 選取要還原的應用程式

執行工作流程 ["列出應用程式"](#) 並選取您要複製的應用程式。用來還原應用程式的REST呼叫需要幾個資源值。

### 2. 選取要使用的備份

執行工作流程 ["列出備份"](#) 並選取您要使用的備份。

### 3. 還原應用程式

執行下列REST API呼叫。您必須提供備份（如下所示）或快照的ID。

HTTP方法	路徑
放入	/Accounts / {account_id} /k8s/v2/apps / {app_id}

### 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
JSON	本文	是的	提供複製應用程式的參數。請參閱以下範例。

### JSON輸入範例

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "backupID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"
}
```

## Curl範例：從備份中就地還原應用程式

```
curl --location -i --request PUT
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps/<APP_ID>'
--header 'Content-Type: application/astra-app+json' --header '*/*'
--header 'ForceUpdate: true' --header 'Authorization: Bearer <API_TOKEN>'
--data @JSONinput
```

## 命名空間

### 列出命名空間

您可以列出可用的命名空間。

#### 1. 列出命名空間

執行下列REST API呼叫、以列出命名空間。

HTTP方法	路徑
取得	/Accounts / {account_id} /拓撲/ v1/命名 空間

**Curl範例：**傳回所有命名空間的所有資料

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/namespaces'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

**Curl範例：**所有命名空間的傳回名稱、狀態和叢集ID

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/namespaces?include=name,namespaceState,clusterID' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>'
```

### Json輸出範例

```
{
  "items": [
    [
      "default",
      "discovered",
      "922f924a-a476-4a79-97f6-472571698154"
    ],
    [
```

```
    "kube-node-lease",
    "discovered",
    "922f924a-a476-4a79-97f6-472571698154"
  ],
  [
    "kube-public",
    "discovered",
    "922f924a-a476-4a79-97f6-472571698154"
  ],
  [
    "kube-system",
    "discovered",
    "922f924a-a476-4a79-97f6-472571698154"
  ],
  [
    "mysql",
    "discovered",
    "922f924a-a476-4a79-97f6-472571698154"
  ],
  [
    "mysql-clone1",
    "discovered",
    "922f924a-a476-4a79-97f6-472571698154"
  ],
  [
    "netapp-acc-operator",
    "discovered",
    "922f924a-a476-4a79-97f6-472571698154"
  ],
  [
    "openshift",
    "discovered",
    "922f924a-a476-4a79-97f6-472571698154"
  ],
  [
    "trident",
    "discovered",
    "922f924a-a476-4a79-97f6-472571698154"
  ]
],
"metadata": {}
}
```

# 支援

## 列出通知

您可以列出特定Astra帳戶的通知。您可以在監控系統活動或偵錯問題時執行此動作。

### 1. 列出通知

執行下列REST API呼叫。

HTTP方法	路徑
取得	/Accounts / {account_id} /核心/ v1/notifications

### 其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
篩選器	查詢	否	選擇性地篩選您要在回應中傳回的通知。
包括	查詢	否	選擇性地選取您要傳回回應中的值。

### Curl範例：傳回所有通知

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/notifications'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

### Curl範例：傳回嚴重性為警告的通知說明

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/notifications?filter=severity%20eq%20'warning'&include=description' --header 'Accept: */*'
--header 'Authorization: Bearer <API_TOKEN>'
```

### Json輸出範例

```
{
  "items": [
    [
      "Trident on cluster david-ie-00 has failed or timed out;
installation of the Trident operator failed or is not yet complete;
operator failed to reach an installed state within 300.00 seconds;
container trident-operator not found in operator deployment"
    ],
    [
      "Trident on cluster david-ie-00 has failed or timed out;
installation of the Trident operator failed or is not yet complete;
operator failed to reach an installed state within 300.00 seconds;
container trident-operator not found in operator deployment"
    ]
  ],
  "metadata": {}
}
```

## 刪除失敗的應用程式

如果託管應用程式的備份或快照處於故障狀態、您可能無法移除該應用程式。在此情況下、您可以使用下列工作流程手動移除應用程式。

### 1. 選取要刪除的應用程式

執行工作流程 ["列出應用程式"](#) 並選取您要移除的應用程式。

### 2. 列出應用程式的現有備份

執行工作流程 ["列出備份"](#)。

### 3. 刪除所有備份

執行工作流程、刪除所有應用程式備份 ["刪除備份"](#) 針對清單中的每個備份。

### 4. 列出應用程式的現有快照

執行工作流程 ["列出快照"](#)。

### 5. 刪除所有快照

執行工作流程 ["刪除快照"](#) 從清單中的每個快照。

### 6. 移除應用程式

執行工作流程 ["取消管理應用程式"](#) 移除應用程式。

## 版權資訊

Copyright © 2023 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

## 商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。