



# 開始使用

## Astra Automation

NetApp  
August 11, 2025

This PDF was generated from [https://docs.netapp.com/zh-tw/astra-automation/get-started/before\\_get\\_started.html](https://docs.netapp.com/zh-tw/astra-automation/get-started/before_get_started.html) on August 11, 2025. Always check [docs.netapp.com](https://docs.netapp.com) for the latest.

# 目錄

開始使用 . . . . .	1
開始之前 . . . . .	1
取得API權杖 . . . . .	1
總覽 . . . . .	1
建立Astra API權杖 . . . . .	1
您的第一個API呼叫 . . . . .	2
Kubernetes基本概念 . . . . .	3
物件 . . . . .	3
命名空間 . . . . .	3
標籤 . . . . .	3
準備好使用工作流程 . . . . .	3
簡介 . . . . .	3
工作流程類別 . . . . .	4
通用輸入參數 . . . . .	4
呈現權杖和識別碼 . . . . .	5
使用範例搭配 Bash . . . . .	5
線上API參考 . . . . .	6
存取 Astra API 參考文件 . . . . .	6
發出 Astra REST API 呼叫 . . . . .	6

# 開始使用

## 開始之前

您可以檢閱下列步驟、快速準備開始使用Astra Control REST API。

### 擁有Astra帳戶認證資料

您需要Astra認證資料才能登入Astra網路使用者介面並產生API權杖。使用Astra Control Center、您可以在本機管理這些認證資料。Astra Control Service可透過\*驗證0\*服務存取帳戶認證資料。

### 熟悉Kubernetes的基本概念

您應該熟悉幾個基本的Kubernetes概念。請參閱 "[Kubernetes基本概念](#)" 以取得更多資訊。

### 檢視REST概念與實作

請務必詳閱 "[核心REST實作](#)" 如需REST概念的相關資訊、以及Astra Control REST API的設計細節。

### 取得更多資訊

您應該瞭解中建議的其他資訊資源 "[其他資源](#)"。

## 取得API權杖

您需要取得Astra API權杖、才能使用Astra Control REST API。

### 總覽

API權杖可識別Astra的呼叫者、且必須包含在每個REST API呼叫中。

- 您需要使用 Astra Web 使用者介面來產生 API 權杖。
- 對於兩個 Astra 部署模型、產生權杖的程序相同。只有用於存取 Astra 的 URL 不同。
- 使用權杖及相關權限所攜帶的使用者身分、是由建立權杖的使用者所決定。
- 權杖必須包含在「授權」HTTP要求標頭中。
- 權杖建立後永遠不會過期。
- 您可以在Astra網路使用者介面上撤銷權杖。

### 相關資訊

- "[撤銷API權杖](#)"

## 建立Astra API權杖

下列步驟說明如何建立Astra API權杖。

### 開始之前

您需要Astra帳戶的認證資料。

## 關於這項工作

此工作會在Astra網路介面產生API權杖。您也應該擷取進行API呼叫時所需的帳戶ID。

### 步驟

1. 使用您的帳戶認證登入 Astra 、如下所示：
  - Astra Control 服務 : "<https://astra.netapp.io>"
  - Astra Control Center : 依照安裝期間所建立的 URL 、使用本機環境的 URL
2. 按一下頁面右上角的圖示、然後選取「\* API access\* (\* API存取\*)」。
3. 按一下頁面上的 \* 產生 API 權杖 \* 、然後在快顯視窗中按一下 \* 產生 API 權杖 \* 。
4. 按一下圖示、將權杖字串複製到剪貼簿、然後儲存到編輯器中。
5. 複製並儲存相同頁面上的帳戶ID。

### 完成後

當您透過 Curl 或程式設計語言存取 Astra Control REST API 時、必須在 HTTP 中加入 API 承載權杖 Authorization 要求標頭：

## 您的第一個API呼叫

您可以在工作站的CLI上發出簡單的Curl命令、開始使用Astra Control REST API並確認其可用度。

### 開始之前

您的本機工作站必須具備 Curl 公用程式。您也必須擁有API權杖和相關的帳戶識別碼。如需詳細資訊、請參閱 "[取得API權杖](#)" 。

### Curl範例

下列 Curl 命令會擷取 Astra 使用者清單。請依指示提供適當的 \$account\_ID 和 \$API\_token 。

```
curl --request GET  \
--location "https://astra.netapp.io/accounts/$ACCOUNT_ID/core/v1/users" \
--include \
--header "Content-Type: application/json" \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

## Json輸出範例

```
{  
  "items": [  
    [  
      "David",  
      "Anderson",  
      "844ec6234-11e0-49ea-8434-a992a6270ec1"  
    ],  
    [  
      "Jane",  
      "Cohen",  
      "2a3e227c-fda7-4145-a86c-ed9aa0183a6c"  
    ]  
,  
  "metadata": {}  
}
```

## Kubernetes基本概念

使用Astra REST API時、有幾個Kubernetes概念是相關的。

### 物件

Kubernetes環境中維護的物件是代表叢集組態的持續實體。這些物件共同說明系統狀態、包括叢集工作負載。

### 命名空間

命名空間提供一種技術、可用來隔離單一叢集內的資源。這種組織架構在劃分工作類型、使用者和資源時非常實用。命名空間範圍為`_namespac_`的物件必須在命名空間內是唯一的、而具有`_cluster`範圍\_的物件則必須在整個叢集內是唯一的。

### 標籤

標籤可與Kubernetes物件建立關聯。它們使用金鑰值配對來描述屬性、並可在叢集上強制執行任意組織、這對組織而言很有用、但卻不在核心Kubernetes作業範圍內。

## 準備好使用工作流程

您應該先熟悉Astra工作流程的組織和格式、再將其用於即時部署。

### 簡介

工作流程\_是完成特定管理工作或目標所需的一或多個步驟順序。Astra Control工作流程的每個步驟都是下列其中一個步驟：

- REST API呼叫（詳細資料如Curl和Json範例）
- 呼叫另一個Astra工作流程
- 其他相關工作（例如做出必要的設計決策）

工作流程包括完成每項工作所需的核心步驟和參數。這些工具可提供自訂自動化環境的起點。



工作流程只能包含一個步驟。這些單一步驟工作流程的格式與包含多個步驟的工作流程略有不同。例如、會移除明確的步驟名稱。應根據工作流程標題來清除行動或作業。

## 工作流程類別

根據您的部署模式、Astra工作流程分為兩大類。如果您使用Astra Control Center、則應從基礎架構工作流程開始、然後繼續進行管理工作流程。使用Astra Control Service時、您通常可以直接前往管理工作流程。



工作流程中的捲髮樣本使用 Astra Control Service 的 URL。當您根據環境使用內部部署的Astra Control Center時、需要變更URL。

### 基礎架構工作流程

這些工作流程適用於Astra基礎架構、包括認證、儲存庫和儲存後端。Astra Control Center需要這些工具、但在大多數情況下、也可搭配Astra Control Service使用。工作流程著重於建立及維護Astra託管叢集所需的工作。

### 管理工作流程

您可以在設定託管叢集之後使用這些工作流程。管理工作流程著重於應用程式保護及支援作業、例如備份、還原及複製應用程式。

## 通用輸入參數

以下所述的輸入參數適用於用來說明REST API呼叫的所有Curl範例。



由於這些輸入參數是通用需求、因此不會在個別工作流程中進一步說明。如果特定的捲曲範例使用其他輸入參數、請參閱\*其他輸入參數\*一節。

### 路徑參數

每次REST API呼叫所使用的端點路徑包括下列參數。另請參閱 "[URL格式](#)" 以取得更多資訊。

#### 帳戶ID

這是UUIDv4值、可識別執行API作業的Astra帳戶。請參閱 "[取得API權杖](#)" 如需尋找帳戶ID的詳細資訊、請參閱。

#### 要求標頭

視REST API呼叫而定、您可能需要包含數個要求標頭。

#### 授權

工作流程中的所有API呼叫都需要API權杖來識別使用者。您必須在「授權」要求標頭中加入該標誌。請參閱

["取得API權杖"](#) 以取得有關產生API權杖的詳細資訊。

## 內容類型

使用HTTP POST並將Json包含在要求本文中的要求放入、您應該根據Astra資源來宣告媒體類型。例如、您可以在建立託管應用程式的快照時、加入標題「Content-Type：application/Astra-appsnap+json」。

## 接受

您可以根據Astra資源、宣告回應中預期內容的特定媒體類型。例如、您可以在列出託管應用程式的備份時、加入標題「Accept: application/Astra appBackup + json」。不過為了簡化、工作流程中的捲曲範例可接受所有媒體類型。

## 呈現權杖和識別碼

與捲動範例搭配使用的API權杖和其他ID值不透明、沒有明顯的意義。因此為了改善範例的讀取性、不會使用實際的權杖和ID值。而是使用較小的保留關鍵字、其優點如下：

- Curl和Json樣本更清楚、更容易瞭解。
- 由於所有關鍵字的格式都與方括弧和大寫字母相同、因此您可以快速識別要插入或擷取的位置和內容。
- 不會遺失任何值、因為無法複製原始參數、並與實際部署搭配使用。

這些變數已格式化、可在 Bash Shell 環境中使用。每個變數都以美元符號開頭、並視需要以雙引號括住。這讓 Bash 能夠辨識他們的能力。大寫字母會持續用於名稱。

以下是Curl範例中使用的一些常用保留關鍵字。此清單並非詳盡無遺、並會視需要使用其他關鍵字。其意義應根據背景而明確。

關鍵字	類型	說明
\$account_ID	路徑	UUIDv4值可識別執行API作業的帳戶。
\$API_Token	標頭	識別及授權呼叫者的承載權杖。
\$APP_ID	路徑	UUIDv4值可識別API呼叫的應用程式。

## 使用範例搭配 Bash

如果您直接使用工作流程捲曲範例、則必須使用適合您環境的值來更新這些範例所包含的變數。您可以手動編輯範例、或是仰賴 Bash Shell 為您進行替代、如下所述。



使用 Bash 的優點之一是、您可以在 Shell 工作階段中一次設定變數值、而非每個 Curl 命令一次。

### 步驟

1. 開啟 Linux 或類似作業系統隨附的 Bash Shell 。
2. 設定您要執行的捲髮範例所包含的變數值。例如：

```
$API_TOKEN=SGgpXHeco6M8PLxzIlgbztA4k3_ex4UCa842hOXHBFA=
```

3. 從工作流程頁面複製捲曲範例、然後貼到 Shell 終端機。
4. 按 **enter** 執行下列工作：

a. 請取代您設定的變數值。

b. 執行 curl 命令。

## 線上API參考

### 存取 Astra API 參考文件

您可以存取Astra Control REST API呼叫的詳細資料、包括HTTP方法、輸入參數和回應。這份完整參考資料有助於使用REST API開發自動化應用程式。

#### 開始之前

您需要認證才能登入 Astra 網路使用者介面進行部署。Astra Control Service 和 Astra Control Center 的參考文件存取程序相同。只有 URL 不同。存取和檢視參考文件不需要 API 權杖。

#### 步驟

1. 使用您的帳戶認證登入 Astra 、如下所示：
  - Astra Control 服務 : "<https://astra.netapp.io>"
  - Astra Control Center : 依照安裝期間所建立的 URL 、使用本機環境的 URL
2. 按一下頁面右上角的圖示、然後選取「\* API access\* (\* API存取\*)」。
3. 在頁面頂端、按一下「\* API Documentation ( API文件\*)」下方顯示的URL。

#### 結果

Swagger 頁面會在新的視窗或索引標籤中啟動。請注意、 URL 包含您登入帳戶的帳戶 ID 。

#### 接下來呢？

您可以選擇從 Swagger 頁面執行 API 呼叫。請參閱 "[發出 Astra REST API 呼叫](#)" 以取得更多資訊。

## 發出 Astra REST API 呼叫

您可以從 API 參考文件頁面發出 Astra Control REST API 呼叫。

#### 開始之前

您必須登入 Astra 並存取 API 參考頁面。請參閱 "[存取 Astra API 參考文件](#)" 以取得更多資訊。您也需要 Token 才能使用 REST API 。請參閱 "[取得API權杖](#)" 以取得有關產生 API Token 的詳細資訊。

#### 步驟

1. 在 API 參考頁面頂端、按一下 \* 授權 \* 。
2. 將您的 API 權杖值複製並貼到快顯視窗的欄位中、然後按一下 \* 授權 \* 、再按一下 \* 關閉 \* 。
3. 向下捲動頁面並開啟所需的 API 呼叫。
4. 按一下滑鼠右鍵 \* 試用 \* 。
5. 在相同的 API 呼叫中向下捲動。提供任何必要的參數值、然後按一下 \* 執行 \* 以發出呼叫。

#### 結果

執行 API 呼叫、並顯示 HTTP 狀態代碼。

## 版權資訊

Copyright © 2025 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。