使用Python

Astra Automation

NetApp
January 31, 2025

This PDF was generated from https://docs.netapp.com/zh-tw/astra-automation-2307/python/astra_toolkits.html on January 31, 2025. Always check docs.netapp.com for the latest.

目錄

使用Python 1
NetApp Astra Control Python SDK 1
原生Python 2

使用Python

NetApp Astra Control Python SDK

NetApp Astra Control Python SDK是開放原始碼套件、可用於自動化Astra Control部署。此套件也是瞭解Astra Control REST API的寶貴資源、可能是建立自己自動化平台的一部分。

> 為了簡化作業、NetApp Astra Control Python SDK將在本頁的其餘部分中稱為* SDK*。

兩種相關軟體工具

SDK包含兩種不同但相關的工具、可在存取Astra Control REST API時以不同的抽象層級運作。

Astra SDK

Astra SDK提供核心平台功能。其中包含一組Python類別、可抽象化基礎REST API呼叫。這些類別可支援各種Astra Control資源的管理動作、包括應用程式、備份、快照和叢集。

Astra SDK是套件的一部分、以單一的形式提供 `astraSDK.py` 檔案：您可以將此檔案匯入您的環境、並直接使用類別。

> * NetApp Astra Control Python SDK*（或僅SDK）是整個套件的名稱。* Astra SDK*是指單一檔案中的核心Python類別 `astraSDK.py`。

工具套件指令碼

除了Astra SDK檔案之外 `toolkit.py` 也可使用指令碼。此指令碼可存取內部定義為Python功能的獨立管理動作、以更高的抽象層級運作。指令碼會匯入Astra SDK、並視需要呼叫類別。

如何存取

您可以使用下列方式存取SDK。

Python套件

SDK可從取得 "Python套件索引" 名稱* actooltooltool包*。套件已指派版本編號、並將視需要繼續更新。您必須使用*子母畫面*套件管理公用程式、將套件安裝到您的環境中。

安裝之後、即可透過放置來使用astraSDK.py類別 `import astraSDK` 在您的指令碼中。此外、 `actoolkit` 可直接在命令提示字元中叫用、且相當於 `toolkit.py` (`actoolkit list clusters` 與相同 `./toolkit.py list clusters`）。

請參閱 "PyPI：NetApp Astra Control Python SDK" 以取得更多資訊。

GitHub原始程式碼

您也可以在GitHub取得SDK原始程式碼。儲存庫包含下列項目：

- `astraSDK.py` （使用Python課程的Astra SDK）
- `toolkit.py` （較高層級的功能型指令碼）

- 詳細的安裝要求與指示
- 安裝指令碼
- 其他文件

您可以複製 "GitHub：NetApp/NetApp-Astra工具套件" 儲存庫。

安裝與基本需求

在安裝套件及準備使用套件時、需要考量幾種選項和需求。

安裝選項摘要

您可以使用下列其中一種方法來安裝SDK：

- 使用準備好的 "Docker：NetApp/Astra工具套件" 映像、其中安裝了所有必要的相依性、包括 `actoolkit`
- 使用Pip來安裝 `actoolkit` 將PyPI封裝至Python環境
- 複製GitHub儲存庫並複製/修改兩個核心Python檔案、以便Python用戶端程式碼存取

如需詳細資訊、請參閱PyPI和GitHub頁面。

Astra Control環境的需求

無論是直接使用Astra SDK中的Python類別、或是中的功能 `toolkit.py` 指令碼、最終您將在Astra Control部署中存取REST API。因此、您需要Astra帳戶和API權杖。請參閱 "開始之前" 如需詳細資訊、請參閱本文件*入門*一節中的其他頁面。

NetApp Astra Control Python SDK的需求

SDK有幾項與本機Python環境相關的先決條件。例如、您必須使用 Python 3.8 或更新版本。此外、還需要幾個Python套件。如需詳細資訊、請參閱GitHub儲存庫頁面或PyPI套件頁面。

實用資源摘要

以下是您開始使用所需的一些資源。

- "PyPI：NetApp Astra Control Python SDK"
- "GitHub：NetApp/NetApp-Astra工具套件"
- "Docker：NetApp/Astra工具套件"

原生Python

開始之前

Python是資料中心自動化的熱門開發語言。在將Python的原生功能與數個通用套件搭配使用之前、您必須先準備環境和所需的輸入檔。

除了直接使用Python存取Astra Control REST API之外、NetApp也提供工具套件、可擷取API並移除部分複雜性。請參閱 "NetApp Astra Control Python SDK" 以取得更多資訊。

準備環境

執行Python指令碼的基本組態需求如下所述。

Python 3.

您需要安裝最新版本的Python 3。

其他程式庫

必須安裝*請求*和* urllib3*程式庫。您可以根據環境使用pip或其他Python管理工具。

網路存取

執行指令碼的工作站必須具備網路存取權、而且必須能夠連線到Astra Control。使用Astra Control Service時、您必須連線至網際網路、才能連線至的服務 `https://astra.netapp.io`。

身分識別資訊

您需要具有帳戶識別碼和API權杖的有效Astra帳戶。請參閱 "取得API權杖" 以取得更多資訊。

建立**Json**輸入檔

Python指令碼仰賴Json輸入檔中包含的組態資訊。範例檔案如下所示。

您需要視環境的需求更新範例。

身分識別資訊

下列檔案包含API權杖和Astra帳戶。您需要使用將此檔案傳遞給Python指令碼 `-i` （或 `--identity`）CLI參數。

```
{
  "api_token": "kH4CA_uVIa8q9UuPzhJaAHaGlaR7-no901DkkrVjIXk=",
  "account_id": "5131dfdf-03a4-5218-ad4b-fe84442b9786"
}
```

列出應用程式

您可以使用下列指令碼來列出Astra帳戶的應用程式。

請參閱 "開始之前" 以取得所需Json輸入檔的範例。

```
#!/usr/bin/env python3
##-----------------------------------------------------------------------------
------
#
# Usage: python3 list_man_apps.py -i identity_file.json
#
# (C) Copyright 2022 NetApp, Inc.
```

```
#
# This sample code is provided AS IS, with no support or warranties of
# any kind, including but not limited for warranties of merchantability
# or fitness of any kind, expressed or implied. Permission to use,
# reproduce, modify and create derivatives of the sample code is granted
# solely for the purpose of researching, designing, developing and
# testing a software application product for use with NetApp products,
# provided that the above copyright notice appears in all copies and
# that the software application product is distributed pursuant to terms
# no less restrictive than those set forth herein.
#
##------------------------------------------------------------------------
------

import argparse
import json
import requests
import urllib3
import sys

# Global variables
api_token = ""
account_id = ""

def get_managed_apps():
    ''' Get and print the list of apps '''

    # Global variables
    global api_token
    global account_id

    # Create an HTTP session
    sess1 = requests.Session()

    # Suppress SSL unsigned certificate warning
    urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

    # Create URL
    url1 = "https://astra.netapp.io/accounts/" + account_id +
"/k8s/v2/apps"

    # Headers and response output
    req_headers  = {}
    resp_headers = {}
    resp_data    = {}
```

```
    # Prepare the request headers
    req_headers.clear
    req_headers['Authorization'] = "Bearer " + api_token
    req_headers['Content-Type'] = "application/astra-app+json"
    req_headers['Accept'] = "application/astra-app+json"

    # Make the REST call
    try:
        resp1 = sess1.request('get', url1, headers=req_headers,
allow_redirects=True, verify=False)

    except requests.exceptions.ConnectionError:
        print("Connection failed")
        sys.exit(1)

    # Retrieve the output
    http_code = resp1.status_code
    resp_headers = resp1.headers

    # Print the list of apps
    if resp1.ok:
        resp_data = json.loads(resp1.text)
        items = resp_data['items']
        for i in items:
            print(" ")
            print("Name: " + i['name'])
            print("ID: " + i['id'])
            print("State: " + i['state'])
    else:
        print("Failed with HTTP status code: " + str(http_code))

    print(" ")

    # Close the session
    sess1.close()

    return

def read_id_file(idf):
    ''' Read the identity file and save values '''

    # Global variables
    global api_token
    global account_id

    with open(idf) as f:
        data = json.load(f)
```

```
    api_token = data['api_token']
    account_id = data['account_id']

    return

def main(args):
    ''' Main top level function '''

    # Global variables
    global api_token
    global account_id

    # Retrieve name of JSON input file
    identity_file = args.id_file

    # Get token and account
    read_id_file(identity_file)

    # Issue REST call
    get_managed_apps()

    return

def parseArgs():
        ''' Parse the CLI input parameters '''

        parser = argparse.ArgumentParser(description='Astra REST API -
List the apps',
                                add_help = True)
        parser.add_argument("-i", "--identity", action="store", dest
="id_file", default=None,
                                help='(Req) Name of the identity input file',
required=True)

        return parser.parse_args()

if __name__ == '__main__':
    ''' Begin here '''

    # Parse input parameters
    args = parseArgs()

    # Call main function
    main(args)
```

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。