



# Python を使用する Astra Automation

NetApp  
August 11, 2025

# 目次

Python を使用する .....	1
NetApp Astra Control Python SDK .....	1
関連する 2 つのソフトウェアツール .....	1
にアクセスする方法 .....	1
インストールと基本的な要件 .....	2
役立つリソースの概要 .....	2
ネイティブ Python .....	3
作業を開始する前に .....	3
アプリを表示します .....	4

# Python を使用する

## NetApp Astra Control Python SDK

NetApp Astra Control Python SDK は、Astra Control の導入を自動化するために使用できるオープンソースパッケージです。また、このパッケージは、独自の自動化プラットフォームを作成する際に、おそらく Astra Control REST API について学ぶ際の貴重な資料となります。



簡単にするために、このページの以降の部分では、NetApp Astra Control Python SDK を \* SDK と呼びます。

### 関連する 2 つのソフトウェアツール

SDK には、Astra Control REST API にアクセスするときに異なる抽象化レベルで動作する 2 つの異なる関連ツールが含まれています。

#### Astra SDK

Astra SDK は、コアプラットフォーム機能を提供します。基盤となる REST API 呼び出しを抽象化する一連の Python クラスが含まれています。このクラスは、アプリケーション、バックアップ、Snapshot、クラスタなど、さまざまな Astra Control リソースに対する管理操作をサポートしています。

Astra SDK はパッケージの一部であり、単一の「astrasdk.py」ファイルで提供されています。このファイルを環境にインポートし、クラスを直接使用できます。



NetApp Astra Control Python SDK \* (または SDK のみ) は、パッケージ全体の名前です。**Astra SDK** は、単一ファイル「astrasdk.py」の中核的な Python クラスを指します。

#### Toolkit スクリプト

Astra SDK ファイルに加えて 'toolkit.py' スクリプトも利用できますこのスクリプトは、Python 関数として内部的に定義されている個別の管理アクションにアクセスできるようにすることで、抽象化のレベルを高めて動作します。このスクリプトは、Astra SDK をインポートし、必要に応じてクラスに呼び出します。

### にアクセスする方法

SDK には、次の方法でアクセスできます。

#### Python パッケージ

SDK は、から入手できます **"Python パッケージインデックス"** 名前\* actoolkit \*。パッケージにはバージョン番号が割り当てられており、必要に応じて更新が継続されます。パッケージを環境にインストールするには、\* PIP \* パッケージ管理ユーティリティを使用する必要があります。

インストールが完了したら、スクリプトに「import astraSDK」を配置することで、astraSDK.py クラスを利用できます。また 'actoolkit' は 'コマンドプロンプトから直接呼び出すことができますこれは 'toolkit.py (actoolkit list clusters' は './toolkit.py list clusters' と同じです)と同じです

を参照してください **"PyPi : NetApp Astra Control Python SDK"** を参照してください。

## GitHub ソースコード

SDK ソースコードは GitHub から入手できます。リポジトリには次のものが含まれます。

- `astrasdk.py` (Python クラスを含む Astra SDK)
- `toolkit.py` (より高度な関数ベースのスクリプト)
- インストール要件と手順の詳細
- インストールスクリプト
- その他のドキュメント

のクローンを作成できます ["GitHub : NetApp / NetApp-Astra - ツールキット"](#) ローカル環境へのリポジトリ。

## インストールと基本的な要件

パッケージのインストールおよび使用準備の一環として、いくつかのオプションと要件を考慮する必要があります。

### インストールオプションの概要

次のいずれかの方法で SDK をインストールできます。

- 準備したを使用します ["Docker : Astra -ツールキット"](#) イメージ(image)は`actoolkit`を含む必要な依存関係をすべてインストールしています
- Pipを使用して`PyPi`の`actoolkit`パッケージをPython環境にインストールします
- GitHubリポジトリのクローンを作成し、Pythonクライアントコードからアクセスできるように、2つのコアPythonファイルをコピー/変更します

詳細については、PyPi および GitHub のページを参照してください。

### Astra Control 環境の要件

Astra SDK の Python クラスを直接使用する場合も、「`toolkit.py`」スクリプトの関数を使用する場合も、最終的には Astra Control 配置で REST API にアクセスすることになります。そのため、API トークンを持つ Astra アカウントが必要になります。を参照してください ["作業を開始する前に"](#) 詳細については、このドキュメントの「\* はじめに \*」セクションの他のページを参照してください。

### NetApp Astra Control Python SDK の要件

SDK には、ローカル Python 環境に関連するいくつかの前提条件があります。たとえば、Python 3.8以降を使用する必要があります。また、必要な Python パッケージもいくつかあります。詳細については、GitHub リポジトリページまたは PyPI パッケージページを参照してください。

## 役立つリソースの概要

開始するために必要なリソースをいくつかご紹介します。

- ["PyPi : NetApp Astra Control Python SDK"](#)
- ["GitHub : NetApp / NetApp-Astra - ツールキット"](#)
- ["Docker : Astra -ツールキット"](#)

# ネイティブ Python

作業を開始する前に

Pythonは、データセンター自動化のための一般的な開発言語です。Python のネイティブ機能をいくつかの共通パッケージとともに使用する前に、環境と必要な入力ファイルを準備する必要があります。



ネットアップは、Python を使用して Astra Control REST API に直接アクセスするだけでなく、API を抽象化して複雑さを排除するツールキットパッケージも提供しています。を参照してください "[NetApp Astra Control Python SDK](#)" を参照してください。

環境を準備

以下に、Python スクリプトを実行するための基本的な設定要件を示します。

## Python 3.

最新バージョンの Python 3 がインストールされている必要があります。

追加ライブラリ

\*Requests \* および \*ullib3 \* ライブラリがインストールされている必要があります。環境に応じて、pip などの Python 管理ツールを使用できます。

ネットワークアクセス

スクリプトを実行するワークステーションは、ネットワークにアクセスし、Astra Control にアクセスできる必要があります。Astra Control Service を使用する場合は、インターネットに接続し、<https://astra.netapp.io> でサービスに接続できる必要があります。

ID 情報

アカウント ID と API トークンを持つ有効な Astra アカウントが必要です。を参照してください "[API トークンを取得します](#)" を参照してください。

**JSON** 入力ファイルを作成します

Python スクリプトは、JSON 入力ファイルに含まれている設定情報に依存します。サンプルファイルは以下のとおりです。



環境に応じて、サンプルを更新する必要があります。

ID 情報

次のファイルに、API トークンと Astra アカウントが含まれています。このファイルは '-i (または --identity)' CLI パラメータを使用して Python スクリプトに渡す必要があります

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

## アプリを表示します

次のスクリプトを使用して、Astraアカウントのアプリケーションを一覧表示できます。



を参照してください ["作業を開始する前に"](#) 必要な JSON 入力ファイルの例を次に示します。

```
#!/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)
```

## 著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および/または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。