



# Usando Python

## Astra Automation

NetApp  
August 11, 2025

# Índice

- Usando Python . . . . . 1
  - O NetApp já está disponível . . . . . 1
    - Duas ferramentas de software relacionadas . . . . . 1
    - Como aceder . . . . . 1
    - Instalação e requisitos básicos . . . . . 2
    - Resumo dos recursos úteis . . . . . 2
  - Python nativo . . . . . 2
    - Antes de começar . . . . . 3
    - Liste as aplicações . . . . . 4

# Usando Python

## O NetApp já está disponível

O NetApp é um pacote de código aberto que você pode usar para automatizar uma implantação do Astra Control. O pacote também é um recurso valioso para aprender sobre a API REST do Astra Control, talvez como parte da criação de sua própria plataforma de automação.



Para simplificar, o SDK do NetApp será referido como o **SDK** ao longo do restante desta página.

## Duas ferramentas de software relacionadas

O SDK inclui duas ferramentas diferentes, embora relacionadas, que operam em diferentes níveis de abstração ao acessar a API REST do Astra Control.

### SDK do Astra

O Astra SDK fornece a funcionalidade principal da plataforma. Ele inclui um conjunto de classes Python que abstraem as chamadas de API REST subjacentes. As classes dão suporte a ações administrativas em vários recursos do Astra Control, incluindo aplicações, backups, snapshots e clusters.

O Astra SDK é uma parte do pacote e é fornecido em um único `astraSDK.py` arquivo. Você pode importar esse arquivo para o seu ambiente e usar as classes diretamente.



O **NetApp** é o nome de todo o pacote. O **\*Astra SDK** refere-se às classes Python principais no único arquivo `astraSDK.py`.

### Script do Toolkit

Além do arquivo Astra SDK, o `toolkit.py` script também está disponível. Este script opera em um nível mais alto de abstração, fornecendo acesso a ações administrativas discretas definidas internamente como funções Python. O script importa o Astra SDK e faz chamadas para as classes conforme necessário.

## Como aceder

Você pode acessar o SDK das seguintes maneiras.

### Pacote Python

O SDK está disponível em "[Índice do Pacote Python](#)" sob o nome **actoolkit**. O pacote recebe um número de versão e continuará a ser atualizado conforme necessário. Você deve usar o utilitário de gerenciamento de pacotes **PIP** para instalar o pacote em seu ambiente.

Uma vez instalado, as classes `astraSDK.py` podem ser utilizadas colocando `import astraSDK` em seus scripts. Além disso, `actoolkit` pode ser invocado diretamente no seu prompt de comando e é equivalente a `toolkit.py` (`actoolkit list clusters` é igual a `./toolkit.py list clusters`a` ).

Consulte "[PyPI: NetApp é um dos nossos selecionados Jogos de Plataforma](#)" para obter mais informações.

### Código-fonte do GitHub

O código-fonte do SDK também está disponível no GitHub. O repositório inclui o seguinte:

- `astraSDK.py` (Astra SDK com classes Python)
- `toolkit.py` (script baseado em funções de nível superior)
- Requisitos e instruções de instalação detalhadas
- Scripts de instalação
- Documentação adicional

Você pode clonar o "[GitHub: NetApp/NetApp-astra-toolkits](#)" repositório para o seu ambiente local.

## Instalação e requisitos básicos

Existem várias opções e requisitos a considerar como parte da instalação do pacote e preparação para usá-lo.

### Resumo das opções de instalação

Você pode instalar o SDK de uma das seguintes maneiras:

- Use a imagem preparada "[Docker: Kits de ferramentas NetApp/astra](#)", que tem todas as dependências necessárias instaladas, incluindo `actoolkit`
- Use o PIP para instalar o `actoolkit` pacote do PyPI em seu ambiente Python
- Clone o repositório do GitHub e copie/modifique os dois arquivos Python principais para que eles estejam acessíveis ao seu código de cliente Python

Consulte as páginas PyPI e GitHub para obter mais informações.

### Requisitos para o ambiente Astra Control

Seja usando diretamente as classes Python no Astra SDK ou as funções no `toolkit.py` script, você estará acessando a API REST em uma implantação do Astra Control. Por causa disso, você precisará de uma conta Astra juntamente com um token de API. Consulte "[Antes de começar](#)" e as outras páginas na seção **Introdução** desta documentação para obter mais informações.

### Requisitos para o SDK Python do NetApp

O SDK tem vários pré-requisitos relacionados ao ambiente Python local. Por exemplo, você deve usar Python 3,8 ou posterior. Além disso, existem vários pacotes Python que são necessários. Consulte a página do repositório do GitHub ou a página do pacote PyPI para obter mais informações.

## Resumo dos recursos úteis

Aqui estão alguns dos recursos que você precisará para começar.

- "[PyPI: NetApp é um dos nossos selecionados Jogos de Plataforma](#)"
- "[GitHub: NetApp/NetApp-astra-toolkits](#)"
- "[Docker: Kits de ferramentas NetApp/astra](#)"

## Python nativo

## Antes de começar

Python é uma linguagem de desenvolvimento popular para automação de data center. Antes de usar os recursos nativos do Python juntamente com vários pacotes comuns, você precisa preparar o ambiente e os arquivos de entrada necessários.



Além de acessar a API REST do Astra Control diretamente usando Python, o NetApp também fornece um pacote de kit de ferramentas que abstrai a API e remove algumas das complexidades. Consulte ["O NetApp já está disponível"](#) para obter mais informações.

## Prepare o ambiente

Os requisitos básicos de configuração para executar os scripts Python são descritos abaixo.

### Python 3

Você precisa ter a versão mais recente do Python 3 instalada.

### Bibliotecas adicionais

As bibliotecas **requests** e **urllib3** devem ser instaladas. Você pode usar o pip ou outra ferramenta de gerenciamento Python conforme apropriado para o seu ambiente.

### Acesso à rede

A estação de trabalho onde os scripts são executados deve ter acesso à rede e ser capaz de alcançar o Astra Control. Ao usar o Astra Control Service, você deve estar conectado à Internet e ser capaz de se conectar ao serviço em <https://astra.netapp.io>.

### Informações de identidade

Você precisa de uma conta Astra válida com o identificador de conta e token de API. Consulte ["Obtenha um token de API"](#) para obter mais informações.

## Crie os arquivos de entrada JSON

Os scripts Python dependem de informações de configuração contidas em arquivos de entrada JSON. Os arquivos de amostra são fornecidos abaixo.



Você precisa atualizar as amostras conforme apropriado para o seu ambiente.

### Informações de identidade

O arquivo a seguir contém o token de API e a conta Astra. Você precisa passar esse arquivo para scripts Python usando o `-i` parâmetro CLI (ou `--identity`).

```
{
  "api_token": "kH4CA_uVIa8q9UuPzhJaAHaGlAR7-no901DkkrVjIXk=",
  "account_id": "5131dfdf-03a4-5218-ad4b-fe84442b9786"
}
```

## Liste as aplicações

Você pode usar o script a seguir para listar os aplicativos da sua conta Astra.



"[Antes de começar](#)" Consulte para obter um exemplo do arquivo de entrada JSON necessário.

```
#!/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)
```

## **Informações sobre direitos autorais**

Copyright © 2025 NetApp, Inc. Todos os direitos reservados. Impresso nos EUA. Nenhuma parte deste documento protegida por direitos autorais pode ser reproduzida de qualquer forma ou por qualquer meio — gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, gravação em fita ou storage em um sistema de recuperação eletrônica — sem permissão prévia, por escrito, do proprietário dos direitos autorais.

O software derivado do material da NetApp protegido por direitos autorais está sujeito à seguinte licença e isenção de responsabilidade:

ESTE SOFTWARE É FORNECIDO PELA NETAPP "NO PRESENTE ESTADO" E SEM QUAISQUER GARANTIAS EXPRESSAS OU IMPLÍCITAS, INCLUINDO, SEM LIMITAÇÕES, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO, CONFORME A ISENÇÃO DE RESPONSABILIDADE DESTES DOCUMENTOS. EM HIPÓTESE ALGUMA A NETAPP SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, INCIDENTAL, ESPECIAL, EXEMPLAR OU CONSEQUENCIAL (INCLUINDO, SEM LIMITAÇÕES, AQUISIÇÃO DE PRODUTOS OU SERVIÇOS SOBRESSALENTE; PERDA DE USO, DADOS OU LUCROS; OU INTERRUPÇÃO DOS NEGÓCIOS), INDEPENDENTEMENTE DA CAUSA E DO PRINCÍPIO DE RESPONSABILIDADE, SEJA EM CONTRATO, POR RESPONSABILIDADE OBJETIVA OU PREJUÍZO (INCLUINDO NEGLIGÊNCIA OU DE OUTRO MODO), RESULTANTE DO USO DESTES SOFTWARES, MESMO SE ADVERTIDA DA RESPONSABILIDADE DE TAL DANO.

A NetApp reserva-se o direito de alterar quaisquer produtos descritos neste documento, a qualquer momento e sem aviso. A NetApp não assume nenhuma responsabilidade nem obrigação decorrentes do uso dos produtos descritos neste documento, exceto conforme expressamente acordado por escrito pela NetApp. O uso ou a compra deste produto não representam uma licença sob quaisquer direitos de patente, direitos de marca comercial ou quaisquer outros direitos de propriedade intelectual da NetApp.

O produto descrito neste manual pode estar protegido por uma ou mais patentes dos EUA, patentes estrangeiras ou pedidos pendentes.

LEGENDA DE DIREITOS LIMITADOS: o uso, a duplicação ou a divulgação pelo governo estão sujeitos a restrições conforme estabelecido no subparágrafo (b)(3) dos Direitos em Dados Técnicos - Itens Não Comerciais no DFARS 252.227-7013 (fevereiro de 2014) e no FAR 52.227- 19 (dezembro de 2007).

Os dados aqui contidos pertencem a um produto comercial e/ou serviço comercial (conforme definido no FAR 2.101) e são de propriedade da NetApp, Inc. Todos os dados técnicos e software de computador da NetApp fornecidos sob este Contrato são de natureza comercial e desenvolvidos exclusivamente com despesas privadas. O Governo dos EUA tem uma licença mundial limitada, irrevogável, não exclusiva, intransferível e não sublicenciável para usar os Dados que estão relacionados apenas com o suporte e para cumprir os contratos governamentais desse país que determinam o fornecimento de tais Dados. Salvo disposição em contrário no presente documento, não é permitido usar, divulgar, reproduzir, modificar, executar ou exibir os dados sem a aprovação prévia por escrito da NetApp, Inc. Os direitos de licença pertencentes ao governo dos Estados Unidos para o Departamento de Defesa estão limitados aos direitos identificados na cláusula 252.227-7015(b) (fevereiro de 2014) do DFARS.

## **Informações sobre marcas comerciais**

NETAPP, o logotipo NETAPP e as marcas listadas em <http://www.netapp.com/TM> são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respectivos proprietários.