



# はじめに

## Astra Automation

NetApp  
December 01, 2023

# 目次

はじめに .....	1
作業を開始する前に .....	1
API トークンを取得します .....	1
最初のAPI呼び出し .....	2
ワークフローを使用する準備をします .....	3
Kubernetes の基本概念 .....	5
オンラインのAPIリファレンス .....	6

# はじめに

## 作業を開始する前に

次の手順を確認して、Astra Control REST API の使用を開始するための準備を簡単に行うことができます。

### Astra アカウントのクレデンシャルが必要

Astra のクレデンシャルが必要になるのは、Astra Web ユーザインターフェイスにサインインして、API トークンを生成する場合です。Astra Control Center を使用すると、これらの資格情報をローカルで管理できます。Astra Control Service を使用すると、アカウントの資格情報に \* Auth0 \* サービスからアクセスできます。

### Kubernetes の基本概念を理解する

Kubernetes のいくつかの基本概念を理解しておく必要があります。を参照してください "[Kubernetes の基本概念](#)" を参照してください。

### REST の概念と実装を確認

必ず確認してください "[コア REST の実装](#)" REST の概念と、Astra Control REST API の設計に関する詳細については、を参照してください。

詳細はこちらをご覧ください

に示す追加情報のリソースを確認しておく必要があります "[その他のリソース](#)"。

## API トークンを取得します

Astra Control REST API を使用するには、Astra API トークンを取得する必要があります。

### はじめに

API トークンは、Astra の呼び出し元を識別し、すべての REST API 呼び出しに含める必要があります。

- Astra Web ユーザインターフェイスを使用して、API トークンを生成できます。
- トークンを使用して伝送されるユーザ ID は、トークンを作成するユーザによって決まります。
- トークンがに含まれている必要があります Authorization HTTP 要求ヘッダー。
- トークンは作成後に期限切れになることはありません。
- トークンは、Astra の Web ユーザインターフェイスから取り消すことができます。

### 関連情報

- "[API トークンを取り消します](#)"

## Astra API トークンを作成

Astra API トークンを作成する手順を次に示します。

作業を開始する前に

Astra アカウントのクレデンシャルが必要です。


このタスクについて

このタスクは、Astra Web インターフェイスで API トークンを生成します。また、API 呼び出しの際にも必要なアカウント ID を取得する必要があります。

手順

1. アカウントのクレデンシャルを使用して Astra にサインインします。

Astra Control Service の次のサイトにアクセスします。"<https://astra.netapp.io>"

2. ページの右上にあるのアイコンをクリックし、\*API access\* を選択します。
3. ページで [API トークンの生成] をクリックし、ポップアップウィンドウで [API トークンの生成] をクリックします。
4. アイコンをクリックしてトークン文字列をクリップボードにコピーし、エディタに保存します。
5. 同じページにあるアカウント ID をコピーして保存します。

完了後

Curlまたはプログラミング言語を使用してAstra Control REST APIにアクセスする場合は、API BearerトークンをHTTPに含める必要があります Authorization 要求ヘッダー。

## 最初のAPI呼び出し

ワークステーションのCLIで簡単なcurlコマンドを問題して、Astra Control REST APIの使用を開始し、利用可能かどうかを確認することができます。

作業を開始する前に

Curl ユーティリティがローカルワークステーションで使用可能である必要があります。また、API トークンと関連付けられているアカウント識別子も必要です。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。

カールの例

次の Curl コマンドは、Astra ユーザのリストを取得します。図に示すように、適切な <account\_ID> と <api\_ctoken> を指定します。

```
curl --location --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/users' --header
'Content-Type: application/json' --header 'Authorization: Bearer
<API_TOKEN>'
```

## JSON 出力例

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

## ワークフローを使用する準備をします

実際の環境で使用する前に、Astra ワークフローの構成と形式を理解しておく必要があります。

### はじめに

`a_workflow_` は、特定の管理タスクまたは目標を達成するために必要な 1 つ以上のステップのシーケンスです。Astra Control ワークフローの各手順は、次のいずれかです。

- REST API 呼び出し（cURL や JSON の例などの詳細を含む）
- 別の Astra ワークフローの呼び出し
- その他の関連タスク（必要な設計決定の実行など）

ワークフローには、各タスクを実行するために必要な主要な手順とパラメータが含まれています。自動化環境をカスタマイズするための出発点となります。

### 共通の入力パラメータ

以下に記載する入力パラメータは、REST API 呼び出しを示すために使用するすべての cURL サンプルに共通しています。



これらの入力パラメータは汎用的に必要なため、個々のワークフローでは詳しく説明していません。特定のコールの例に追加の入力パラメータが使用される場合は、「\* その他の入力パラメータ \*」セクションで説明します。

## パスパラメータ

すべての REST API 呼び出しで使用されるエンドポイントパスには、次のパラメータが含まれています。も参照してください ["URL 形式"](#) を参照してください。

### アカウント ID

これは、API 処理を実行する Astra アカウントを識別する UUIDv4 値です。を参照してください ["API トークンを取得します"](#) アカウント ID の検索の詳細については、を参照してください。

### 要求ヘッダー

REST API 呼び出しに応じて、いくつかの要求ヘッダーを含める必要があります。

### 承認

ワークフロー内のすべての API 呼び出しで、ユーザを識別するための API トークンが必要です。にトークンを含める必要があります `Authorization` 要求ヘッダー。を参照してください ["API トークンを取得します"](#) API トークンの生成の詳細については、を参照してください。

### コンテンツタイプ

要求の本文に JSON が含まれている HTTP POST 要求と PUT 要求では、Astra リソースに基づいてメディアタイプを宣言する必要があります。たとえば、ヘッダーを含めることができます `Content-Type: application/astra-appSnap+json` 管理対象アプリケーションのスナップショットを作成する場合。

### 同意します

アストラリソースに基づいて、応答で想定されるコンテンツの特定のメディアタイプを宣言できます。たとえば、ヘッダーを含めることができます `Accept: application/astra-appBackup+json` 管理対象アプリケーションのバックアップをリスト表示する場合。ただし、簡単にするために、ワークフロー内のコールサンプルはすべてのメディアタイプに対応しています。

## トークンと識別子の表示

cURL の例で使用される API トークンおよびその他の ID 値は不透明で、認識不能な意味はありません。サンプルの読みやすさを向上させるために、実際のトークンと ID 値は使用されません。代わりに、小さい予約済みキーワードが使用されます。これには次のような利点があります。

- cURL と JSON のサンプルは、より明確でわかりやすくなっています。
- すべてのキーワードは角かっこ大文字で同じ形式を使用するため、挿入または抽出する場所とコンテンツをすばやく識別できます。
- 元のパラメータをコピーして実際の配置で使用することはできないため、値は失われません。

次に、curl の例で使用される一般的な予約済みキーワードの一部を示します。このリストはすべてを網羅しているわけではなく、必要に応じてその他のキーワードが使用されているその意味はコンテキストに基づいて明確になる必要があります。

キーワード	を入力します	説明
<account_ID>	パス	API 処理を実行するアカウントを識別する UUIDv4 値。
<api_ctoken> のように指定します	ヘッダー	発信者を識別および認可するベアラトークン。

キーワード	を入力します	説明
<APP_ID>	パス	API呼び出しのアプリケーションを識別するUUIDv4値。

## ワークフローのカテゴリ

導入モデルに応じて、幅広い種類の Astra ワークフローが用意されています。Astra Control Center を使用している場合は、インフラワークフローから始めて、管理ワークフローに進みます。Astra Control Service を使用すると、通常は管理ワークフローに直接移動できます。



ワークフロー内の cURL のサンプルでは、Astra Control Service の URL を使用します。オンプレミスの Astra Control Center を使用している場合は、環境に応じて URL を変更する必要があります。

### インフラワークフロー

これらのワークフローは、クレデンシャル、バケット、ストレージバックエンドなどの Astra インフラを処理します。Astra Control Center で必要ですが、ほとんどの場合は Astra Control Service でも使用できます。このワークフローでは、マネージドクラスタの構築と保守に必要なタスクを中心に説明します。

### 管理ワークフロー

これらのワークフローは、管理対象クラスタを作成したあとに使用できます。このワークフローでは、アプリケーションのバックアップ、リストア、クローニングなど、アプリケーションの保護とサポートの処理に重点を置いています。

## Kubernetes の基本概念

Kubernetes の概念については、Astra REST API の使用時にいくつか関連するものがあります。

### オブジェクト

Kubernetes 環境で管理されるオブジェクトは、クラスタの構成を表す永続的なエンティティです。これらのオブジェクトのことから、クラスタワークロードを含むシステムの状態がわかります。

### ネームスペース

ネームスペースは、単一のクラスタ内でリソースを分離する手法を提供します。この組織構造は、作業、ユーザ、およびリソースのタイプを分割する場合に便利です。namespace scope\_ を持つオブジェクトはネームスペース内で一意である必要があり、\_cluster scope\_ を持つオブジェクトはクラスタ全体で一意である必要があります。

### ラベル

ラベルは Kubernetes オブジェクトに関連付けることができます。キーと値のペアを使用して属性を記述します。また、Kubernetes の中核的な処理には含まれない、組織には役立つ任意の組織をクラスタに適用できます。

# オンラインのAPIリファレンス

HTTPメソッド、入力パラメータ、応答など、Astra Control REST API呼び出しの詳細にアクセスできます。このリファレンスは、REST API を使用して自動化アプリケーションを開発する場合に役立ちます。



REST API のリファレンスドキュメントは、現在のところ Astra Control に付属しており、オンラインで入手できます。

作業を開始する前に

Astra Control Center または Astra Control Service のアカウントが必要です。

手順

1. アカウントのクレデンシャルを使用して Astra にサインインします。

Astra Control Service の次のサイトにアクセスします。 "<https://astra.netapp.io>"

2. ページの右上にある図のアイコンをクリックし、 \* API access \* を選択します。
3. ページの上部で、 \* API ドキュメント \* の下に表示される URL をクリックします。
4. プロンプトが表示されたら、アカウントのクレデンシャルを再度入力し



## 著作権に関する情報

Copyright © 2023 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。