



入门

Astra Automation

NetApp
August 11, 2025

This PDF was generated from https://docs.netapp.com/zh-cn/astra-automation/get-started/before_get_started.html on August 11, 2025. Always check docs.netapp.com for the latest.

目录

入门	1
开始之前	1
获取 API 令牌	1
概述	1
创建 Astra API 令牌	1
您的第一次API调用	2
基本 Kubernetes 概念	3
对象	3
命名空间	3
标签	3
准备使用这些工作流	3
简介	3
工作流类别	4
通用输入参数	4
表示令牌和标识符	5
将示例与Bash结合使用	5
联机API参考	6
访问Astra API参考文档	6
问题描述Asta REST API调用	6

入门

开始之前

您可以通过查看以下步骤快速准备开始使用 Astra Control REST API。

拥有 Astra 帐户凭据

您需要使用 Astra 凭据登录到 Astra Web 用户界面并生成 API 令牌。使用 Astra 控制中心，您可以在本地管理这些凭据。使用 Astra 控制服务时，帐户凭据可通过 * 身份验证 0* 服务进行访问。

熟悉 Kubernetes 的基本概念

您应熟悉 Kubernetes 的几个基本概念。请参见 "[基本 Kubernetes 概念](#)" 有关详细信息 ...

查看 REST 概念和实施

请务必查看 "[核心 REST 实施](#)" 有关 REST 概念的信息以及有关如何设计 Astra Control REST API 的详细信息。

获取更多信息

您应了解中建议的追加信息资源 "[其他资源](#)"。

获取 API 令牌

要使用 Astra Control REST API，您需要获取 Astra API 令牌。

概述

API 令牌用于标识 Astra 的调用方，并且必须包含在每个 REST API 调用中。

- 您需要使用Asta Web用户界面生成API令牌。
- 用于生成令牌的操作步骤对于这两种Astra部署模式是相同的。只有用于访问Astra的URL不同。
- 与令牌一起携带的用户身份以及相关权限由创建令牌的用户确定。
- 令牌必须包含在 Authorization HTTP Request 标头中。
- 令牌创建后永不过期。
- 您可以在 Astra Web 用户界面上撤消令牌。

相关信息

- "[撤消 API 令牌](#)"

创建 Astra API 令牌

以下步骤介绍如何创建 Astra API 令牌。

开始之前

您需要 Astra 帐户的凭据。

关于此任务

此任务会在 Astra Web 界面上生成 API 令牌。此外，您还应检索帐户 ID，在进行 API 调用时也需要使用此 ID。

步骤

1. 使用您的帐户凭据登录ASRA、如下所示：
 - Astra控制服务： "<https://astra.netapp.io>"
 - Astra Control Center：使用安装期间为您的本地环境建立的URL
2. 单击页面右上角的图图标并选择 * API access* 。
3. 单击页面上的*生成API令牌*，然后在弹出窗口中单击*生成API令牌*。
4. 单击图标将令牌字符串复制到剪贴板并将其保存在编辑器中。
5. 复制并保存同一页面上提供的帐户 ID 。

完成后

通过cURL或编程语言访问Astra Control REST API时、必须在HTTP中包含API承载令牌 Authorization 请求标题。

您的第一次API调用

您可以在工作站的命令行界面上通过问题描述 执行一个简单的cURL命令来开始使用Astra Control REST API并确认其可用性。

开始之前

您的本地工作站必须提供CURL实用程序。您还必须具有 API 令牌和关联的帐户标识符。有关详细信息、请参见 "[获取 API 令牌](#)" 。

curl 示例

以下cURL命令将检索Asta用户列表。按照说明提供相应的\$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 输出示例

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

基本 Kubernetes 概念

使用 Astra REST API 时，有几个相关的 Kubernetes 概念。

对象

Kubernetes 环境中维护的对象是表示集群配置的永久性实体。这些对象共同描述了系统状态，包括集群工作负载。

命名空间

命名空间提供了一种隔离单个集群中的资源的技术。在划分工作类型、用户类型和资源类型时，此组织结构非常有用。命名空间范围 `_` 的对象在命名空间中必须是唯一的，而具有 `_cluster` 范围 `_` 的对象必须在整个集群中是唯一的。

标签

标签可以与 Kubernetes 对象关联。它们使用键值对描述属性，并可在集群上强制实施任意组织，这对组织可能有用，但不在核心 Kubernetes 操作范围内。

准备使用这些工作流

在将 Astra 工作流用于实时部署之前，您应熟悉这些工作流的组织和格式。

简介

`_Workflow_` 是完成特定管理任务或目标所需的一个或多个步骤的序列。Astra Control 工作流中的每个步骤均为以下步骤之一：

- REST API 调用（包含 curl 和 JSON 示例等详细信息）
- 调用另一个 Astra 工作流
- 其他相关任务（例如做出必要的设计决策）

这些工作流包括完成每个任务所需的核心步骤和参数。它们为自定义自动化环境提供了一个起点。



一个工作流只能包含一个步骤。这些单步工作流的格式与包含多个步骤的工作流略有不同。例如、删除显式步骤名称。操作或操作应根据工作流标题明确。

工作流类别

根据您的部署模式，有两大类 Astra 工作流可用。如果您使用的是 Astra 控制中心，则应先从基础架构工作流开始，然后再继续执行管理工作流。使用 Astra Control Service 时，通常可以直接转到管理工作流。



工作流中的curl示例使用Astra Control Service的URL。在根据您的环境使用内部Astra控制中心时，您需要更改 URL。

基础架构工作流

这些工作流用于处理 Astra 基础架构，包括凭据，存储分段和存储后端。Astra 控制中心需要使用这些控制器，但在大多数情况下，也可以与 Astra 控制服务一起使用。这些工作流侧重于建立和维护 Astra 受管集群所需的任务。

管理工作流

设置受管集群后、您可以使用这些工作流。管理工作流侧重于应用程序保护和支持操作、例如备份、还原和克隆应用程序。

通用输入参数

以下所述的输入参数对于用于说明 REST API 调用的所有 curl 示例通用。



由于这些输入参数是通用的要求，因此在各个工作流中不会对其进行进一步说明。如果在特定的 curl 示例中使用了其他输入参数，则会在 * 其他输入参数 * 一节中进行介绍。

路径参数

每次 REST API 调用使用的端点路径都包括以下参数。另请参见 "[URL 格式](#)" 有关详细信息 ...

帐户 ID

这是用于标识运行 API 操作的 Astra 帐户的 UUIDv4 值。请参见 "[获取 API 令牌](#)" 有关查找帐户 ID 的详细信息，请参见。

请求标题

根据 REST API 调用，您可能需要包含多个请求标头。

Authorization

工作流中的所有 API 调用都需要 API 令牌来标识用户。您必须在 Authorization Request 标题中包含令牌。请参见 "[获取 API 令牌](#)" 有关生成 API 令牌的详细信息。

内容类型

对于请求正文中包含 JSON 的 HTTP POST 和 PUT 请求，您应根据 Astra 资源声明介质类型。例如，在为受管应用程序创建快照时，可以包括标题 Content-Type : application/Astra-appsnap+json。

接受

您可以根据 Astra 资源声明响应中预期内容的特定介质类型。例如，在列出受管应用程序的备份时，您可以包括标题 accept : application/Astra-appBackup+json。但是，为了简单起见，工作流中的 curl 样本接受所有介质类型。

表示令牌和标识符

在 curl 示例中使用的 API 令牌和其他 ID 值是不透明的，没有明显的含义。因此，为了提高示例的可读性，不会使用实际令牌和 ID 值。而是使用较小的保留关键字，它具有以下几个优势：

- curl 和 JSON 示例更清晰，更易于理解。
- 由于所有关键字都使用相同的格式以及括号和大写字母，因此您可以快速确定要插入或提取的位置和内容。
- 不会丢失任何值，因为无法复制原始参数并将其用于实际部署。

这些变量经过格式设置、可在 bash shell 环境中使用。每个变量都以美元符号开头、并根据需要用双引号括起来。这使得他们可以被巴什所识别。名称始终使用大写。

以下是在 curl 示例中使用的一些常见保留关键字。此列表并非详尽无遗，我们会根据需要使用其他关键字。根据具体情况，其含义应该是显而易见的。

关键字	Type	Description
\$account_ID	路径	用于标识运行 API 操作的帐户的 UUIDv4 值。
\$API_TOAUTHOR	标题	标识和授权调用方的令牌。
\$APP_ID	路径	用于标识 API 调用的应用程序的 UUIDv4 值。

将示例与 Bash 结合使用

如果您直接使用工作流卷曲示例，则必须使用适合您环境的值更新其包含的变量。您可以手动编辑这些示例、也可以依靠 bash shell 为您执行替换、如下所述。



使用 "Bash" 的一个优点是、您可以在 shell 会话中设置一次变量值、而不是在每个 CURL 命令中设置一次。

步骤

1. 打开随 Linux 或类似操作系统提供的 bash shell。
2. 设置要运行的 CURL 示例中包含的变量值。例如：

```
$API_TOKEN=SGgpXHeco6M8PLxzIlgbztA4k3_ex4UCa842hOXHBFA=
```

3. 从工作流页面复制卷曲示例并将其粘贴到shell终端中。

4. 按*enter*执行以下任务：

a. 替换您设置的变量值。

b. 执行Curl命令。

联机API参考

访问Astra API参考文档

您可以访问Astra Control REST API调用的详细信息、包括HTTP方法、输入参数和响应。在使用 REST API 开发自动化应用程序时，此完整参考非常有用。

开始之前

您需要凭据才能登录到适用于您的部署的ASRA Web用户界面。用于访问参考文档的操作步骤对于Astra控制服务和Astra控制中心是相同的。只有URL不同。访问和查看参考文档不需要API令牌。

步骤

1. 使用您的帐户凭据登录ASRA、如下所示：

- Astra控制服务： "<https://astra.netapp.io>"
- Astra Control Center： 使用安装期间为您的本地环境建立的URL

2. 单击页面右上角的图图标并选择 * API access*。

3. 在页面顶部，单击 * API Documentage* 下显示的 URL。

结果

Swagger页面将在新窗口或选项卡中启动。请注意、此URL包含您登录的帐户的帐户ID。

下一步是什么？

您可以选择从Swagger页面执行API调用。请参见 "[问题描述Asta REST API调用](#)" 有关详细信息 ...

问题描述Asta REST API调用

您可以从问题描述参考文档页面对Astra Control REST API调用执行API调用。

开始之前

您必须登录Astra并访问API参考页面。请参见 "[访问Astra API参考文档](#)" 有关详细信息 ...您还需要一个令牌才能使用REST API。请参见 "[获取 API 令牌](#)" 有关生成API令牌的详细信息。

步骤

1. 在API参考页面顶部、单击*授权*。
2. 将您的API令牌值复制并粘贴到弹出窗口的字段中，单击*Authorize*，然后单击*Close*。
3. 向下滚动页面并打开所需的API调用。
4. 在右键单击*试用*。

5. 在同一API调用中向下滚动。提供任何所需的参数值，然后单击*Execute (执行)*以问题描述此调用。

结果

此时将执行API调用并显示HTTP状态代码。

版权信息

版权所有 © 2025 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。