



入门

Astra Automation 22.04

NetApp
December 04, 2023

目录

- 入门 1
 - 开始之前 1
 - 获取 API 令牌 1
 - 大家好 2
 - 准备使用这些工作流 3
 - 基本 Kubernetes 概念 5

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。

简介

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。

- 您可以使用 Astra Web 用户界面生成 API 令牌。
- 令牌附带的用户身份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到 Astra。

访问以下站点以获取 Astra 控制服务：["https://astra.netapp.io"](https://astra.netapp.io)

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 Authorization Request 标头中包含 API 承载令牌。

大家好

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

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。请参见 ["获取 API 令牌"](#) 有关详细信息 ...

curl 示例

以下 Curl 命令将检索 Astra 用户列表。按指示提供相应的 <account_ID> 和 <api_token>。

```
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 输出示例

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

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。

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。

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_token>	标题	标识和授权调用方的令牌。
<managed_app_ID>	路径	用于标识 API 调用的受管应用程序的 UUIDv4 值。

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 控制服务的 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 workflow

这些 workflow 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 workflow 侧重于建立和维护 Astra 受管集群所需的任务。

管理工作流

您可以在拥有受管集群后使用这些 workflow。这些 workflow 侧重于应用程序保护和支持操作，例如备份，还原和克隆受管应用程序。

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的状态，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。

版权信息

版权所有 © 2023 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。