# Manos a la obra

## Astra Automation 22.04

NetApp
October 23, 2024

# Tabla de contenidos

# Manos a la obra

## Antes de empezar

Puede prepararse rápidamente para empezar con la API REST de Astra Control revisando los siguientes pasos.

#### Tener credenciales de cuenta de Astra

Necesitará credenciales de Astra para iniciar sesión en la interfaz de usuario web de Astra y generar un token de API. Con Astra Control Center, puede gestionar estas credenciales localmente. Con Astra Control Service, se accede a las credenciales de la cuenta a través del servicio **Auth0**.

#### Familiarícese con los conceptos básicos de Kubernetes

Debería estar familiarizado con varios conceptos básicos de Kubernetes. Consulte "Conceptos básicos de Kubernetes" si quiere más información.

#### Revisar los conceptos e implementación de REST

Asegúrese de revisarlo "Implementación básica de REST" Para obtener información sobre conceptos DE REST y detalles sobre cómo se diseña la API ASTRA Control REST.

#### Obtenga más información

Debe conocer los recursos de información adicionales que se sugieren en la "Recursos adicionales".

## Obtenga un token de API

Necesita obtener un token de API Astra para utilizar la API REST de Astra Control.

### Introducción

Un identificador de API identifica a la persona que llama a Astra y debe incluirse con cada llamada de API DE REST.

- Puede generar un token de API mediante la interfaz de usuario web de Astra.
- La identidad de usuario que se lleva con el token la determina el usuario que crea el token.
- El token debe incluirse en el `Authorization` Encabezado de solicitud HTTP.
- Un token nunca caduca después de que se crea.
- Puede revocar un token en la interfaz de usuario web de Astra.

#### Información relacionada

- "Revocar un token de API"

### Cree un token de API Astra

En los siguientes pasos se describe cómo crear un token de API de Astra.

#### Antes de empezar

Necesita credenciales para una cuenta Astra.

### Acerca de esta tarea

Esta tarea genera un token de API en la interfaz web de Astra. También debe recuperar el ID de cuenta que también se necesita al realizar llamadas API.

### Pasos

1. Inicie sesión en Astra con sus credenciales de cuenta.

   Acceda a las siguientes instalaciones para el servicio Astra Control: "https://astra.netapp.io"

2. Haga clic en el icono de figura situado en la parte superior derecha de la página y seleccione **acceso API**.
3. Haga clic en **generar símbolo de API** en la página y, en la ventana emergente, haga clic en **generar símbolo de API**.
4. Haga clic en el icono para copiar la cadena de token al portapapeles y guardarla en el editor.
5. Copie y guarde el ID de cuenta que está disponible en la misma página.

### Después de terminar

Cuando accede a la API REST de Astra Control mediante Curl o un lenguaje de programación, debe incluir el token del portador de API en HTTP `Authorization` solicite el encabezado.

# Hola mundo

Puede emitir un sencillo comando Curl en la interfaz de línea de comandos de su estación de trabajo para comenzar a utilizar la API ASTRA Control REST y confirmar su disponibilidad.

### Antes de empezar

La utilidad Curl debe estar disponible en la estación de trabajo local. También debe tener un token de API y el identificador de cuenta asociado. Consulte "Obtenga un token de API" si quiere más información.

### Ejemplo de curl

El siguiente comando Curl recupera una lista de usuarios de Astra. Proporcione el <ACCOUNT_ID> y el <API_TOKEN> adecuados según se indica.

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

**Ejemplo de resultado JSON**

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

# Prepárese para usar los flujos de trabajo

Debe estar familiarizado con la organización y el formato de los flujos de trabajo de Astra antes de utilizarlos con una implementación en directo.

## Introducción

Un *Workflow* es una secuencia de uno o más pasos necesarios para llevar a cabo una tarea o un objetivo administrativos específicos. Cada paso de un flujo de trabajo de Astra Control es uno de los siguientes:

- Llamada a API REST (con detalles como ejemplos curl y JSON)
- Invocación de otro flujo de trabajo de Astra
- Tareas relacionadas varias (como tomar una decisión de diseño necesaria)

Los flujos de trabajo incluyen los pasos principales y los parámetros necesarios para realizar cada tarea. Proporcionan un punto de partida para personalizar el entorno de automatización.

## Parámetros de entrada comunes

Los parámetros de entrada descritos a continuación son comunes a todas las muestras de curl utilizadas para ilustrar una llamada a la API DE REST.

> Debido a que estos parámetros de entrada son universalmente necesarios, no se describen más adelante en los flujos de trabajo individuales. Si se utilizan parámetros de entrada adicionales para un ejemplo de rizo específico, se describen en la sección **parámetros de entrada adicionales**.

### Parámetros de ruta

La ruta de extremo utilizada con cada llamada de API DE REST incluye los siguientes parámetros. Consulte también "Formato de URL" si quiere más información.

#### ID de cuenta

Este es el valor UUIDv4 que identifica la cuenta Astra en la que se ejecuta la operación API. Consulte "Obtenga un token de API" Para obtener más información acerca de cómo localizar su ID de cuenta.

### Solicitar encabezados

Existen varios encabezados de solicitud que puede necesitar incluir en función de la llamada a la API DE REST.

#### Autorización

Todas las llamadas API de los flujos de trabajo necesitan un token de API para identificar al usuario. Debe incluir el token en el `Authorization` solicite el encabezado. Consulte "Obtenga un token de API" Para obtener más información acerca de la generación de un token de API.

#### Tipo de contenido

Con LA POST HTTP y LAS peticiones DE PONER donde JSON está incluido en el cuerpo de la solicitud, debe declarar el tipo de medio basado en el recurso Astra. Por ejemplo, puede incluir el encabezado `Content-Type: application/astra-appSnap+json` al crear una instantánea para una aplicación administrada.

#### Acepte

Puede declarar el tipo de medio específico del contenido que espera en la respuesta en función del recurso Astra. Por ejemplo, puede incluir el encabezado `Accept: application/astra-appBackup+json` al enumerar los backups de una aplicación gestionada. Sin embargo, para mayor simplicidad, las muestras curl de los flujos de trabajo aceptan todos los tipos de medios.

## Presentación de tokens e identificadores

El token de la API y otros valores de ID utilizados con los ejemplos curl son opacos sin significado discernible. Para mejorar la legibilidad de las muestras, no se utilizan los valores de identificador y token reales. Más bien, se utilizan palabras clave reservadas más pequeñas que tiene varias ventajas:

- Las muestras curl y JSON son más claras y fáciles de entender.
- Puesto que todas las palabras clave utilizan el mismo formato con corchetes y letras mayúsculas, puede identificar rápidamente la ubicación y el contenido que se debe insertar o extraer.
- No se pierde ningún valor porque los parámetros originales no se pueden copiar y utilizar con una implementación real.

Aquí están algunas de las palabras clave reservadas comunes usadas en los ejemplos curl. Esta lista no es exhaustiva y se utilizan palabras clave adicionales según sea necesario. Su significado debe ser obvio basado en el contexto.

| Palabra clave | Tipo | Descripción |
|---|---|---|
| <ACCOUNT_ID> | Ruta | El valor UUIDv4 que identifica la cuenta en la que se ejecuta la operación API. |
| <API_TOKEN> | Encabezado | El token del portador identifica y autoriza al llamante. |

| Palabra clave | Tipo | Descripción |
|---|---|---|
| <MANAGED_APP_ID> | Ruta | El valor UUIDv4 que identifica la aplicación gestionada para la llamada API. |

## Categorías de flujo de trabajo

Existen dos amplias categorías de flujos de trabajo de Astra disponibles en función de su modelo de puesta en marcha. Si utiliza Astra Control Center, debería empezar con los flujos de trabajo de la infraestructura y, a continuación, proceder a los flujos de trabajo de gestión. Cuando utilice Astra Control Service, normalmente puede dirigirse directamente a los flujos de trabajo de gestión.

> Los ejemplos de curl de los flujos de trabajo utilizan la dirección URL del servicio de control Astra. Debe cambiar la dirección URL cuando utilice el Centro de control Astra de las instalaciones según sea necesario para su entorno.

### Flujos de trabajo de infraestructura

Estos flujos de trabajo hacen frente a la infraestructura Astra, que incluye credenciales, bloques y back-ends. Se necesitan con Astra Control Center, pero en la mayoría de los casos también se pueden utilizar con Astra Control Service. Los flujos de trabajo se centran en las tareas necesarias para establecer y mantener un clúster gestionado por Astra.

### Flujos de trabajo de gestión

Puede utilizar estos flujos de trabajo después de tener un clúster gestionado. Los flujos de trabajo se centran en la protección de las aplicaciones y en operaciones de soporte como la copia de seguridad, la restauración y la clonado de una aplicación gestionada.

# Conceptos básicos de Kubernetes

Hay varios conceptos de Kubernetes que son relevantes cuando se usa la API REST de Astra.

## Objetos

Los objetos que se mantienen en un entorno de Kubernetes son entidades persistentes que representan la configuración del clúster. Estos objetos describen de manera colectiva el estado del sistema, incluida la carga de trabajo del clúster.

## Espacios de nombres

Los espacios de nombres proporcionan una técnica para aislar recursos dentro de un único clúster. Esta estructura organizativa es útil cuando se dividen los tipos de trabajo, los usuarios y los recursos. Los objetos con un *Namespace Scope* deben ser únicos dentro del espacio de nombres, mientras que los que tienen un *cluster scope* deben ser únicos en todo el clúster.

## Etiquetas

Las etiquetas pueden asociarse con los objetos de Kubernetes. Describen los atributos mediante pares de clave-valor y pueden aplicar una organización arbitraria en el clúster que puede ser útil para una organización pero que se encuentra fuera del funcionamiento de Kubernetes principal.