Documentación de Astra Control Automation 22.04

Astra Automation 22.04

NetApp
December 04, 2023

Tabla de contenidos

Documentación de Astra Control Automation 22.04

Notas de la versión

Acerca de esta versión

La documentación de este sitio describe la API Astra Control REST y las tecnologías de automatización relacionadas disponibles con la versión de Astra Control de abril de 2022 (22.04). En concreto, esta versión de LA API REST se incluye con las versiones 22.04 correspondientes de Astra Control Center y Astra Control Service.

Consulte las siguientes páginas y sitios para obtener más información acerca de esta versión, así como las versiones anteriores:

- "Novedades de la API REST de Astra Control"
- "Recursos DE REST y extremos"
- "Documentación de Astra Control Center 22.04"
- "Documentación de Astra Control Service"
- "Versiones anteriores de la documentación de Astra Automation"

Novedades de la API ASTRA Control REST

NetApp actualiza periódicamente la API REST de Astra Control para ofrecerle nuevas funciones, mejoras y correcciones de errores.

26 de abril de 2022 (22.04)

Esta versión incluye una ampliación y actualización de la API DE REST, así como funciones de seguridad y administración mejoradas.

Recursos de Astra nuevos y mejorados

Se han añadido dos nuevos tipos de recursos: **Paquete** y **actualización**. Además, se han actualizado las versiones de varios recursos existentes.

RBAC mejorado con granularidad de espacio de nombres

Al enlazar un rol a un usuario asociado, es posible limitar los espacios de nombres a los que tiene acceso el usuario. Consulte la referencia **Role Binding API** y. "Seguridad RBAC" si quiere más información.

Extracción del cucharón

Puede retirar un cucharón cuando ya no sea necesario o no funcione correctamente.

Compatibilidad con Cloud Volumes ONTAP

Cloud Volumes ONTAP ahora es compatible como back-end de almacenamiento.

Mejoras adicionales del producto

Hay varias mejoras adicionales en las dos implementaciones de productos de Astra Control, que incluyen:

- Entrada genérica para Astra Control Center
- Clúster privado en AKS
- Compatibilidad con Kubernetes 1.22
- Soporte para la cartera de Tanzania de VMware

Consulte la página **Novedades** en los sitios de documentación de Astra Control Center y Astra Control Service.

Información relacionada

- "Astra Control Center: Lo nuevo"
- "Astra Control Service: Novedades"

14 de diciembre de 2021 (21.12)

Esta versión incluye una ampliación de LA API DE REST junto con un cambio en la estructura de documentación para respaldar mejor la evolución de Astra Control con las futuras actualizaciones de versiones.

Separe la documentación de Astra Automation para cada versión de Astra Control

Cada versión de Astra Control incluye una API de REST distinta que se ha mejorado y adaptado a las funciones de la versión específica. La documentación de cada versión de la API REST de Astra Control ya está disponible en su propio sitio web dedicado junto con el repositorio de contenido de GitHub asociado. El sitio del documento principal "Automatización de control de Astra" siempre contiene la documentación de la versión más reciente. Consulte "Versiones anteriores de la documentación de Astra Control Automation" para obtener información acerca de versiones anteriores.

Expansión de los tipos de recursos de REST

El número de tipos de recursos DE REST ha seguido aumentando con un énfasis en los enlaces de ejecución y los back-ends de almacenamiento. Los nuevos recursos incluyen: Cuenta, enlace de ejecución, origen de gancho, anulación de gancho de ejecución, nodo de clúster, gestión del back-end de almacenamiento, espacio de nombres, dispositivo de almacenamiento y nodo de almacenamiento. Consulte "Recursos" si quiere más información.

Kit de desarrollo de software Astra Control Python de NetApp

Astra Control Python SDK de NetApp es un paquete de código abierto que facilita el desarrollo de código de automatización para su entorno de Astra Control. El núcleo es Astra SDK, que incluye un conjunto de clases para abstraer la complejidad de las llamadas API REST. También hay un script de kit de herramientas para ejecutar tareas administrativas específicas empaquetando y extrayendo las clases de Python. Consulte "Kit de desarrollo de software Astra Control Python de NetApp" si quiere más información.

5 de agosto de 2021 (21.08)

Esta versión incluye la introducción de un nuevo modelo de puesta en marcha de Astra y una importante ampliación de LA API DE REST.

Modelo de implementación de Astra Control Center

Además de la oferta existente de Astra Control Service que se proporciona como servicio de cloud público, esta versión incluye también el modelo de puesta en marcha en las instalaciones de Astra Control Center. Puede instalar Astra Control Center en sus instalaciones para gestionar su entorno local de Kubernetes. Los dos modelos de puesta en marcha de Astra Control comparten la misma API DE REST, con pequeñas diferencias observadas en la documentación.

Expansión de los tipos de recursos de REST

El número de recursos a los que se puede acceder mediante la API REST de Astra Control se ha ampliado enormemente y muchos de los nuevos recursos proporcionan una base para la oferta local de Astra Control Center. Los nuevos recursos incluyen: ASUP, autorización, función, licencia, configuración suscripción, bloque, cloud, clúster, clúster gestionado, storage backend y clase de almacenamiento. Consulte "Recursos" si quiere más información.

Puntos finales adicionales compatibles con la implementación de Astra

Además de los recursos REST ampliados, hay varios otros extremos API nuevos disponibles para admitir una puesta en marcha de Astra Control.

Soporte para openapi

Los extremos de OpenAPI proporcionan acceso al documento JSON de OpenAPI actual y a otros recursos relacionados.

Compatibilidad con OpenMetrics

Los extremos de OpenMetrics proporcionan acceso a las métricas de cuentas mediante el recurso OpenMetrics.

15 de abril de 2021 (21.04)

Esta versión incluye las siguientes funciones y mejoras nuevas.

Introducción de la API de REST

La API REST de Astra Control está disponible para su uso con la oferta de Astra Control Service. Se ha creado a partir de tecnologías DE REST y prácticas recomendadas vigentes. La API proporciona una base para la automatización de sus implementaciones de Astra e incluye las siguientes funciones y ventajas.

Recursos

Hay catorce tipos de recursos DE REST disponibles.

Acceso de token de API

El acceso a la API DE REST se proporciona mediante un token de acceso de la API que se puede generar en la interfaz de usuario web de Astra. El token de API proporciona acceso seguro a la API.

Soporte para colecciones

Hay un amplio conjunto de parámetros de consulta que se pueden utilizar para tener acceso a las colecciones de recursos. Algunas de las operaciones admitidas son el filtrado, la ordenación y la paginación.

Problemas conocidos

Debe revisar todos los problemas conocidos de la versión actual relacionados con la API REST de Astra Control. Los problemas conocidos identifican problemas por los que el uso correcto del producto puede resultar imposible.

> No hay nuevos problemas conocidos con la versión 22.04 de la API REST de Astra Control. Los problemas descritos a continuación se detectaron en versiones anteriores y siguen siendo aplicables con la versión actual.

No se detectan todos los dispositivos de almacenamiento de un nodo de almacenamiento back-end

Cuando se emite una llamada API DE REST para recuperar los dispositivos de almacenamiento definidos en un nodo de almacenamiento, no se devuelven todos los dispositivos.

Introducción a la API REST de Astra Control

Astra Control Center y Astra Control Service proporcionan una API de REST común a la que puede acceder directamente a través de un lenguaje de programación o utilidad como Curl. A continuación se muestran los principales aspectos destacados y ventajas de la API.

> Para acceder a la API DE REST, primero debe iniciar sesión en la interfaz de usuario web de Astra y generar un token de API. Debe incluir el token con cada solicitud de API.

Basada en la tecnología DE REST

La API Astra Control se ha creado con la tecnología REST y las mejores prácticas actuales. La tecnología principal incluye HTTP, JSON y RBAC.

Compatibilidad con los dos modelos de puesta en marcha de Astra Control

El servicio Astra Control se utiliza en el entorno de cloud público, mientras que Astra Control Center se utiliza para sus puestas en marcha en las instalaciones. Hay una API DE REST que admite ambos modelos de puesta en marcha.

Borrar la asignación entre los recursos de extremo DE REST y el modelo de objetos

Los extremos DE REST externos utilizados para acceder a los recursos se asignan a un modelo de objetos coherente mantenido internamente por el servicio Astra. El modelo de objetos se ha diseñado utilizando el modelado de relación-entidad (ER), que ayuda a definir claramente las acciones y respuestas de API.

Amplio conjunto de parámetros de consulta

La API REST proporciona un amplio conjunto de parámetros de consulta que se pueden utilizar para acceder a las colecciones de recursos. Algunas de las operaciones admitidas son el filtrado, la ordenación y la paginación.

Alineación con la interfaz de usuario web de Astra Control

El diseño de la interfaz de usuario web de Astra se alinea con la API DE REST, por lo que existe coherencia entre las dos rutas de acceso y la experiencia de usuario.

Depuración robusta y determinación de problemas

La API REST de Astra Control proporciona una sólida capacidad de depuración y determinación de problemas, incluidos eventos del sistema y notificaciones del usuario.

Procesos de flujo de trabajo

Se proporciona un conjunto de flujos de trabajo para ayudar en el desarrollo del código de automatización. Los flujos de trabajo se organizan en dos categorías principales: Infraestructura y gestión.

Base para tecnologías de automatización avanzadas

Además de acceder directamente a la API DE REST, puede usar otras tecnologías de automatización basadas en la API DE REST.

Parte de la documentación de la familia Astra

La documentación de Astra Control Automation forma parte de la documentación más amplia de la familia Astra. Consulte "Documentación de Astra" si quiere más información.

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.

Implementación básica de REST

Servicios web REST

La transferencia de estado representacional (REST) es un estilo para crear aplicaciones web distribuidas. Cuando se aplica al diseño de una API de servicios web, establece un conjunto de tecnologías generales y prácticas recomendadas para exponer recursos basados en servidor y administrar sus estados. Aunque REST proporciona una base consistente para el desarrollo de aplicaciones, los detalles de cada API pueden variar en función de las opciones de diseño específicas. Debe conocer las características de la API REST de Astra Control antes de utilizarla con una implementación en directo.

Recursos y representación estatal

Los recursos son los componentes básicos de un sistema basado en la Web. Al crear una aplicación DE SERVICIOS web DE REST, las tareas de diseño más tempranas incluyen:

- Identificación de recursos basados en sistemas o servidores

  Cada sistema utiliza y mantiene los recursos. Un recurso puede ser un archivo, una transacción comercial, un proceso o una entidad administrativa. Una de las primeras tareas en el diseño de una aplicación basada en servicios web DE REST es identificar los recursos.

- Definición de estados de recursos y operaciones estatales asociadas

  Los recursos siempre se encuentran en uno de un número limitado de estados. Los estados, así como las operaciones asociadas utilizadas para afectar los cambios de estado, deben estar claramente definidos.

Extremos de URI

Todos los recursos REST deben definirse y ponerse a disposición mediante un esquema de direccionamiento bien definido. Los extremos en los que se encuentran e identifican los recursos utilizan un identificador uniforme de recursos (URI). El URI proporciona un marco general para crear un nombre único para cada recurso de la red. El Localizador uniforme de recursos (URL) es un tipo de URI que se utiliza con los servicios web para identificar y acceder a los recursos. Los recursos normalmente se exponen en una estructura jerárquica similar a un directorio de archivos.

Mensajes HTTP

El Protocolo de transferencia de hipertexto (HTTP) es el protocolo utilizado por el cliente y servidor de servicios web para intercambiar mensajes de solicitud y respuesta sobre los recursos. Como parte del diseño de una aplicación de servicios web, los métodos HTTP se asignan a los recursos y a las correspondientes acciones de administración del estado. HTTP no tiene estado. Por lo tanto, para asociar un conjunto de solicitudes y respuestas relacionadas como parte de una transacción, se debe incluir información adicional en los encabezados HTTP transportados con los flujos de datos de solicitud y respuesta.

Formato JSON

Aunque la información se puede estructurar y transferir entre un cliente de servicios web y un servidor de varias maneras, la opción más popular es la notación de objetos JavaScript (JSON). JSON es un estándar del

sector para representar estructuras de datos simples en texto sin formato y se utiliza para transferir información de estado que describe los recursos. La API REST de Astra Control utiliza JSON para dar formato a los datos transportados en el cuerpo de cada solicitud y respuesta HTTP.

Recursos y colecciones

La API REST de Astra Control proporciona acceso a las instancias de recursos y las colecciones de instancias de recursos.

> Conceptualmente UN RESTO **recurso** es similar a un **objeto** tal y como se define con los lenguajes y sistemas de programación orientada a objetos (OOP). A veces estos términos se utilizan indistintamente. Pero, en general, se prefiere "recurso" cuando se utiliza en el contexto de la API de REST externa mientras que "objeto" se utiliza para los datos de instancia con estado correspondientes almacenados en el servidor.

Atributos de los recursos de Astra

La API REST de Astra Control cumple los principios de diseño RESTful. Cada instancia de recurso de Astra se crea según un tipo de recurso bien definido. Un conjunto de instancias de recursos del mismo tipo se denomina **colección**. Las llamadas API actúan sobre recursos individuales o colecciones de recursos.

Tipos de recursos

Los tipos de recursos incluidos con la API REST de Astra Control tienen las siguientes características:

- Cada tipo de recurso se define mediante un esquema (normalmente en JSON)
- Cada esquema de recursos incluye el tipo de recurso y la versión
- Los tipos de recursos son globalmente únicos

Instancias de recursos

Las instancias de recursos disponibles a través de la API REST de Astra Control tienen las siguientes características:

- Las instancias de recursos se crean en función de un único tipo de recurso
- El tipo de recurso se indica mediante el valor Tipo de soporte
- Las instancias se componen de datos con estado que el servicio Astra mantiene
- Se puede acceder a cada instancia mediante una dirección URL única y de larga duración
- En los casos en que una instancia de recurso puede tener más de una representación, se pueden utilizar diferentes tipos de medios para solicitar la representación deseada

Colecciones de recursos

Las colecciones de recursos disponibles a través de la API REST de Astra Control tienen las siguientes características:

- El conjunto de instancias de recursos de un único tipo de recurso se conoce como una colección
- Las colecciones de recursos tienen una URL única y de larga duración

Identificadores de instancia

Cada instancia de recurso tiene asignado un identificador cuando se crea. Este identificador es un valor UUIDv4 de 128 bits. Los valores UIDv4 asignados son globalmente únicos e inmutables. Después de emitir

una llamada API que crea una nueva instancia, se devuelve una URL con el ID asociado al llamante en un `Location` Encabezado de la respuesta HTTP. Puede extraer el identificador y utilizarlo en llamadas posteriores cuando haga referencia a la instancia del recurso.

> El identificador de recurso es la clave principal utilizada para las colecciones.

Estructura común para los recursos de Astra

Cada recurso de Astra Control se define mediante una estructura común.

Datos comunes

Cada recurso de Astra contiene los valores clave que se muestran en la siguiente tabla.

Clave	Descripción
tipo	Un tipo de recurso único global que se conoce como el **tipo de recurso**.
versión	Un identificador de versión que se conoce como **versión de recurso**.
id	Identificador único global que se conoce como el **identificador de recurso**.
metadatos	Un objeto JSON que contiene diversa información, incluidas las etiquetas de usuario y del sistema.

Objeto de metadatos

El objeto JSON de metadatos incluido con cada recurso de Astra contiene los valores clave que se muestran en la siguiente tabla.

Clave	Descripción
etiquetas	Cabina JSON de etiquetas especificadas por el cliente asociadas con el recurso.
CreationTimestamp	Cadena JSON que contiene una Marca de tiempo que indica cuándo se creó el recurso.
Modificación.Marca de hora	Cadena JSON que contiene una Marca de tiempo con formato ISO-8601 que indica cuándo se modificó por última vez el recurso.
CreatedBy	Cadena JSON que contiene el identificador UUIDv4 del ID de usuario que creó el recurso. Si el recurso fue creado por un componente interno del sistema y no hay un UUID asociado con la entidad de creación, se utiliza el UUID **nulo**.

Estado del recurso

Recursos seleccionados a `state` valor que se utiliza para orquestar transiciones de ciclo de vida y controlar el acceso.

Detalles de HTTP

La API REST de Astra Control utiliza HTTP y los parámetros relacionados para actuar en los recursos y las colecciones. A continuación se presentan los detalles de la implementación HTTP.

Transacciones API y modelo CRUD

La API REST de Astra Control implementa un modelo transaccional con operaciones bien definidas y transiciones de estado.

Transacción de API de solicitud y respuesta

Cada llamada de API REST se realiza como una solicitud HTTP al servicio Astra. Cada solicitud genera una respuesta asociada al cliente. Este par de solicitud-respuesta puede considerarse una transacción API.

Compatibilidad con el modelo operativo CRUD

Se accede a cada una de las instancias y colecciones de recursos disponibles a través de la API REST de Astra Control basándose en el modelo **CRUD**. Hay cuatro operaciones, cada una de las cuales se asigna a un único método HTTP. Entre las operaciones se incluyen:

- Cree
- Lea
- Actualizar
- Eliminar

Para algunos de los recursos de Astra, sólo se admite un subconjunto de estas operaciones. Debe revisar el "Referencia de API" Para obtener más información acerca de una llamada API específica.

Métodos HTTP

Los métodos o verbos HTTP soportados por la API se presentan en la tabla siguiente.

Método	CRUD	Descripción
OBTENGA	Lea	Recupera propiedades de objeto para una instancia o colección de recursos. Esto se considera una operación de **lista** cuando se utiliza con una colección.
PUBLICAR	Cree	Crea una nueva instancia de recurso basada en los parámetros de entrada. La URL a largo plazo se devuelve en un `Location` encabezado de respuesta.
PUESTO	Actualizar	Actualiza una instancia de recurso completa con el cuerpo de solicitud JSON proporcionado. Se conservan los valores clave que no pueden modificarse el usuario.
ELIMINAR	Eliminar	Elimina una instancia de recurso existente.

Encabezados de solicitud y respuesta

En la siguiente tabla se resumen los encabezados HTTP utilizados con la API REST de Astra Control.

Consulte "RFC 7232" y.. "RFC 7233" si quiere más información.

Encabezado	Tipo	Notas de uso
Acepte	Solicitud	Si el valor es "/" o no se proporciona, `application/json` Se devuelve en el encabezado de respuesta Content-Type. Si el valor se establece en Tipo de medio de recurso de Astra, se devuelve el mismo tipo de medio en el encabezado Tipo de contenido.
Autorización	Solicitud	Token del portador con la clave API para el usuario.
Tipo de contenido	Respuesta	Devuelto en función del `Accept` solicite el encabezado.
ETag	Respuesta	Se incluye con un éxito según se define con RFC 7232. El valor es una representación hexadecimal del valor MD5 para todo el recurso JSON.
Coincidencia IF	Solicitud	Cabecera de solicitud de condición previa implementada como se describe en la sección 3.1 RFC 7232 y soporte para solicitudes **PUT**.
If-Modified-Since	Solicitud	Cabecera de solicitud de condición previa implementada como se describe en la sección 3.4 RFC 7232 y soporte para solicitudes **PUT**.
If-Unmodified-since	Solicitud	Cabecera de solicitud de condición previa implementada como se describe en la sección 3.4 RFC 7232 y soporte para solicitudes **PUT**.
Ubicación	Respuesta	Contiene la dirección URL completa del recurso recién creado.

Parámetros de consulta

Los siguientes parámetros de consulta están disponibles para su uso con colecciones de recursos. Consulte "Trabajar con colecciones" si quiere más información.

Parámetro de consulta	Descripción
incluya	Contiene los campos que se deben devolver al leer una colección.
filtro	Indica los campos que deben coincidir para que se devuelva un recurso al leer una colección.
OrderBy	Determina el orden de los recursos devueltos al leer una colección.
límite	Limita el número máximo de recursos devueltos al leer una colección.
omitir	Establece el número de recursos que se van a transferir y omitir al leer una colección.
cuente	Indica si se debe devolver el número total de recursos en el objeto de metadatos.

códigos de estado HTTP

A continuación se describen los códigos de estado HTTP utilizados por la API DE REST de Astra Control.

> La API REST de Astra Control también utiliza el estándar **Detalles del problema para API de HTTP**. Consulte "Diagnóstico y soporte" si quiere más información.

Codificación	Significado	Descripción
200	DE ACUERDO	Indica que las llamadas que no crean una nueva instancia de recurso tienen éxito.
201	Creado	Se ha creado correctamente un objeto y el encabezado de respuesta de ubicación incluye el identificador único del objeto.
204	Sin contenido	La solicitud se ha realizado correctamente aunque no se ha devuelto ningún contenido.
400	Solicitud incorrecta	La entrada de la solicitud no se reconoce o no es apropiada.
401	No autorizado	El usuario no está autorizado y debe authentir.
403	Prohibido	Se deniega el acceso debido a un error de autorización.
404	No encontrado	El recurso al que se hace referencia en la solicitud no existe.
409	Conflicto	Error al intentar crear un objeto porque el objeto ya existe.
500	Error interno	Se ha producido un error interno general en el servidor.
503	Servicio no disponible	El servicio no está listo para atender la solicitud por algún motivo.

Formato de URL

La estructura general de la URL utilizada para acceder a una instancia de recurso o a una colección a través de la API DE REST está compuesta por varios valores. Esta estructura refleja el modelo de objeto subyacente y el diseño del sistema.

Cuenta como la raíz

La raíz de la ruta de recursos a cada extremo DE REST es la cuenta Astra. Por lo tanto, todas las rutas de la URL comienzan con `/account/{account_id}` donde `account_id` Es el valor único UUIDv4 de la cuenta. Estructura interna esto refleja un diseño en el que todo el acceso a los recursos se basa en una cuenta específica.

Categoría de recurso de extremo

Los extremos de recursos de Astra se dividen en tres categorías diferentes:

- Núcleo (`/core`)
- Aplicación gestionada (`/k8s`)
- Topología (`/topology`)

Consulte "Recursos" si quiere más información.

Versión de categoría

Cada una de las tres categorías de recursos tiene una versión global que controla la versión de los recursos a los que se tiene acceso. Por convención y definición, pasar a una nueva versión principal de una categoría de recursos (como, por ejemplo, de `/v1` para `/v2`) Introducirá cambios de ruptura en la API.

Instancia o colección de recursos

Se puede usar una combinación de tipos de recursos e identificadores en la ruta de acceso, en función de si

se accede a una instancia de recurso o a una recopilación.

Ejemplo

- Ruta de recursos

  En función de la estructura presentada anteriormente, una ruta típica a un punto final es: `/accounts/{account_id}/core/v1/users`.

- Complete la dirección URL

  La dirección URL completa del punto final correspondiente es: `https://astra.netapp.io/accounts/{account_id}/core/v1/users`.

Recursos y extremos

Puede utilizar los recursos proporcionados a través de la API REST de Astra Control para automatizar una implementación de Astra. Cada recurso tiene acceso mediante uno o varios extremos. La información que se presenta a continuación proporciona una introducción a los recursos DE REST que se pueden usar como parte de una implementación de automatización.

> El formato de la ruta y la dirección URL completa que se utiliza para acceder a los recursos de Astra Control se basa en varios valores. Consulte "Formato de URL" si quiere más información. Consulte también "Referencia de API" Para obtener más información sobre el uso de los recursos y puntos finales de Astra.

Resumen de recursos DE ASTRA Control REST

Los extremos de recursos principales proporcionados en la API REST de Astra Control se organizan en tres categorías. Se puede tener acceso a cada recurso con el conjunto completo de operaciones CRUD (crear, leer, actualizar, eliminar) excepto donde se indique.

La columna **Release** indica la versión Astra cuando se introdujo el recurso por primera vez. Este campo está en negrita para los recursos recién añadidos con la versión actual.

Recursos básicos

Los principales extremos de recursos proporcionan los servicios básicos necesarios para establecer y mantener el entorno de tiempo de ejecución de Astra.

Recurso	Liberar	Descripción
Cuenta	21.12	Los recursos de la cuenta le permiten gestionar los inquilinos aislados dentro del entorno de implementación Multitenant Astra Control.
ASUP	21.08	Los recursos de ASUP representan los paquetes de AutoSupport que se envían al soporte de NetApp.
Credencial	21.04	Los recursos de credenciales contienen información relacionada con la seguridad que se puede utilizar con los usuarios de Astra, los clústeres, los bloques y los back-ends de almacenamiento.
Prestaciones	21.08	Los recursos de derechos representan las características y capacidades disponibles para una cuenta en función de las licencias y suscripciones activas.
Evento	21.04	Los recursos de eventos representan todos los eventos que se producen en el sistema, incluido el subconjunto clasificado como notificaciones.
Gancho de ejecución	21.12	Los recursos del enlace de ejecución representan secuencias de comandos personalizadas que se pueden ejecutar antes o después de realizar una instantánea de una aplicación administrada.
Función	21.08	Los recursos de la función representan las funciones Astra seleccionadas que puede consultar para determinar si están habilitadas o deshabilitadas en el sistema. El acceso está limitado a solo lectura.

Recurso	Liberar	Descripción
Fuente de gancho	21.12	Los recursos de origen de enlace representan el código fuente real utilizado con un enlace de ejecución. Separar el código fuente del control de ejecución tiene varias ventajas, como permitir que se compartan las secuencias de comandos.
Licencia	21.08	Los recursos de licencia representan las licencias disponibles para una cuenta de Astra.
Notificación	21.04	Los recursos de notificación representan eventos Astra que tienen un destino de notificación. El acceso se proporciona por usuario.
Paquete	**22.04**	Los recursos del paquete proporcionan registro y acceso a definiciones de paquetes. Los paquetes de software constan de varios componentes, incluidos archivos, imágenes y otros artefactos.
Vinculación de roles	21.04	Los recursos de vinculación de roles representan las relaciones entre pares específicos de usuarios y cuentas. Además del vínculo entre los dos, se especifica un conjunto de permisos para cada uno mediante una función específica.
Ajuste	21.08	Los recursos de configuración representan una colección de pares clave-valor que describen una función para una cuenta Astra específica.
Suscripción	21.08	Los recursos de suscripción representan las suscripciones activas de una cuenta Astra.
Token	21.04	Los recursos de token representan los tokens disponibles para acceder mediante programación a la API REST de Astra Control.
Notificación sin leer	21.04	Los recursos de notificación no leídos representan notificaciones asignadas a un usuario específico pero aún no leídas.
Renovar	**22.04**	Los recursos de actualización proporcionan acceso a los componentes de software y la capacidad de iniciar actualizaciones.
Usuario	21.04	Los recursos de usuario representan a los usuarios de Astra capaces de acceder al sistema en función de su función definida.

Recursos de aplicaciones gestionados

Los extremos de recursos de la aplicación gestionada proporcionan acceso a las aplicaciones de Kubernetes gestionadas.

Recurso	Liberar	Descripción
Activo de aplicación	21.04	Los recursos de activos de aplicación representan colecciones internas de información de estado necesarias para gestionar las aplicaciones Astra.
Backup de aplicaciones	21.04	Los recursos de backup de la aplicación representan backups de las aplicaciones gestionadas.
Instantánea de aplicación	21.04	Los recursos de instantánea de la aplicación representan instantáneas de las aplicaciones gestionadas.
Anulación de gancho de ejecución	21.12	Los recursos de anulación del enlace para la ejecución le permiten deshabilitar los enlaces de ejecución predeterminados de NetApp preinstalados para aplicaciones específicas según sea necesario.

Recurso	Liberar	Descripción
Aplicación gestionada	21.04	Los recursos de aplicaciones gestionadas representan las aplicaciones de Kubernetes gestionadas por Astra.
Programación	21.04	Los recursos de programación representan operaciones de protección de datos programadas para las aplicaciones gestionadas como parte de una política de protección de datos.

Recursos de topología

Los extremos de recursos de topología proporcionan acceso a las aplicaciones no administradas y a los recursos de almacenamiento.

Recurso	Liberar	Descripción
APL	21.04	Los recursos de aplicaciones representan todas las aplicaciones de Kubernetes, incluidas las que no son gestionadas por Astra.
Cucharón	21.08	Los recursos de bucket representan los bloques cloud de S3 que se utilizan para almacenar backups de las aplicaciones que gestiona Astra.
Cloud	21.08	Los recursos cloud representan las nubes a las que los clientes de Astra pueden conectarse con el fin de gestionar clústeres y aplicaciones.
Clúster	21.08	Los recursos del clúster representan los clústeres de Kubernetes que no gestiona Kubernetes.
Nodo del clúster	21.12	Los recursos del nodo de clúster proporcionan una resolución adicional al permitirle acceder a cada nodo dentro de un clúster Kubernetes.
Clúster gestionado	21.08	Los recursos de clúster gestionados representan los clústeres de Kubernetes que gestiona actualmente Kubernetes.
Gestión del back-end de almacenamiento	21.12	Los recursos de back-end de almacenamiento gestionados le permiten acceder a representaciones abstraídas de los proveedores de almacenamiento del entorno de administración. Estos back-ends de almacenamiento pueden ser utilizados por los clústeres y las aplicaciones gestionados.
Espacio de nombres	21.12	Los recursos de espacio de nombres ofrecen acceso a los espacios de nombres que se usan en un clúster de Kubernetes.
Back-end de almacenamiento	21.08	Los recursos de back-end de almacenamiento representan proveedores de servicios de almacenamiento que pueden utilizar los clústeres y aplicaciones gestionados de Astra.
Clase de almacenamiento	21.08	Los recursos de la clase de almacenamiento representan diferentes clases o tipos de almacenamiento detectados y disponibles para un clúster gestionado específico.
Volumen	21.04	Los recursos de volúmenes representan los volúmenes de almacenamiento de Kubernetes asociados con las aplicaciones gestionadas.

Nuevos extremos con la versión actual

Se han añadido los siguientes extremos DE REST con la versión actual de Astra Control 22.04. Además, se han actualizado las versiones de varios recursos existentes.

- /accounts/{account_id}/core/v1/packages
- /accounts/{account_id}/core/v1/packages/{package_id}
- /accounts/{account_id}/core/v1/upgrades
- /accounts/{account_id}/core/v1/upgrade//{upgrade_id}
- /Accounts/{account_id}/topolog/v1/appbackups
- /Accounts/{account_id}/topolog/v1/appbackups/{appBackup_id}
- /Accounts/{account_id}/topolog/v1/cloud/{cloud_id}/Clusters/{cluster_id}/clusterNodes
- /Accounts/{account_id}/topolog/v1/cloud/{cloud_id}/Clusters/{cluster_id}/clusterNodes/{clusternode_id}
- /Accounts/{account_id}/topolog/v1/managedClusters/{managedCluster_id}/apps/{app_id}/appAssets
- /Accounts/{account_id}/topolog/v1/managedClusters/{managedCluster_id}/apps/{app_id}/appAssets/{appAsset_id}
- /Accounts/{account_id}/topolog/v1/managedClusters/{managedCluster_id}/clusterNodes
- /Accounts/{account_id}/topolog/v1/managedClusters/{managedCluster_id}/clusterNodes/{clusternode_id}

Recursos adicionales y extremos

Existen varios recursos y puntos finales adicionales que puede utilizar para dar soporte a una implementación de Astra.

> Estos recursos y extremos no se incluyen actualmente con la documentación de referencia de la API REST de Astra Control.

Openapi

Los extremos de OpenAPI proporcionan acceso al documento JSON de OpenAPI actual y a otros recursos relacionados.

OpenMetrics

Los extremos OpenMetrics proporcionan acceso a las métricas de la cuenta mediante el recurso OpenMetrics. Existe soporte para el modelo de puesta en marcha de Astra Control Center.

Consideraciones de uso adicionales

Seguridad RBAC

La API REST de Astra admite el control de acceso basado en roles (RBAC) para restringir el acceso a las funciones del sistema.

Funciones de Astra

Cada usuario de Astra se asigna a una sola función que determina las acciones que se pueden realizar. Los roles se organizan en una jerarquía tal y como se describe en la tabla siguiente.

Función	Descripción
Propietario	Tiene todos los permisos de la función Admin y también puede eliminar cuentas Astra.
Admin	Tiene todos los permisos de la función Miembro y también puede invitar a los usuarios a unirse a una cuenta.
Miembro	Puede gestionar completamente la aplicación Astra y los recursos informáticos.
Visionador	Limitado a sólo ver recursos.

RBAC mejorado con granularidad de espacio de nombres

Esta función se introdujo con la versión 22.04 de la API ASTRA REST.

Cuando se establece un enlace de roles para un usuario específico, se puede aplicar una restricción para limitar los espacios de nombres a los que el usuario tiene acceso. Hay varias formas de definir esta restricción como se describe en la tabla siguiente. Consulte el parámetro `roleContraints` En la API de enlace de roles para obtener más información.

Espacios de nombres	Descripción
Todo	El usuario puede acceder a todos los espacios de nombres mediante el parámetro comodín "*". Este es el valor predeterminado para mantener la compatibilidad con versiones anteriores.
Ninguno	La lista de restricciones se especifica aunque esté vacía. Esto indica que el usuario no puede acceder a ningún espacio de nombres.
Lista Namespace	Se incluye el UUID de un espacio de nombres que restringe al usuario al espacio de nombres único. También se puede utilizar una lista separada por comas para permitir el acceso a varios espacios de nombres.
Etiqueta	Se especifica una etiqueta y se permite el acceso a todos los espacios de nombres coincidentes.

Trabajar con colecciones

La API REST de Astra Control proporciona varias formas diferentes de acceder a las colecciones de recursos a través de los parámetros de consulta definidos.

Selección de valores

Puede especificar qué parejas de clave-valor deben devolverse para cada instancia de recursos mediante el `include` parámetro. Todas las instancias se devuelven en el cuerpo de respuesta.

Filtrado

El filtrado de recursos de recopilación permite al usuario de API especificar condiciones que determinan si un recurso se devuelve en el cuerpo de respuesta. La `filter` el parámetro se utiliza para indicar la condición de filtrado.

Ordenación

La ordenación de recursos de recopilación permite al usuario de API especificar el orden en el que se devuelven los recursos en el cuerpo de respuesta. La `orderBy` el parámetro se utiliza para indicar la condición de filtrado.

Paginación

Puede aplicar la paginación restringiendo el número de instancias de recursos devueltas en una solicitud mediante `limit` parámetro.

Cuente

Si incluye el parámetro booleano `count` establezca en `true`, el número de recursos de la matriz devuelta para una respuesta determinada se proporciona en la sección de metadatos.

Diagnóstico y soporte

Hay varias funciones de soporte disponibles con la API REST de Astra Control que se pueden utilizar para diagnósticos y depuración.

Recursos de API

Existen varias funciones de Astra expuestas a través de los recursos de API que proporcionan información y soporte de diagnóstico.

Tipo	Descripción
Evento	Actividades del sistema que se registran como parte del procesamiento de Astra.
Notificación	Subconjunto de los eventos que se consideran lo suficientemente importantes como para ser presentados al usuario.
Notificación sin leer	Las notificaciones que el usuario debe leer o recuperar aún.

Revocar un token de API

Puede revocar un token de API en la interfaz web de Astra cuando ya no sea necesario.

Antes de empezar

Necesita una cuenta Astra. También debe identificar los tokens que desea revocar.

Acerca de esta tarea

Después de que se revoque un token, éste se puede utilizar de forma inmediata y permanente.

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. Seleccione el token o tokens que desea revocar.
4. En el cuadro desplegable **acciones**, haga clic en **revocar tokens**.

Flujos de trabajo de infraestructura

Antes de empezar

Puede utilizar estos flujos de trabajo para crear y mantener la infraestructura utilizada con el modelo de puesta en marcha de Astra Control Center. En la mayoría de los casos, los flujos de trabajo también se pueden utilizar con Astra Control Service.

> Estos flujos de trabajo pueden ampliarse y mejorarse mediante NetApp en cualquier momento, por lo que debe revisarlos periódicamente.

Preparación general

Antes de utilizar cualquiera de los flujos de trabajo de Astra, asegúrese de revisarlos "Prepárese para usar los flujos de trabajo".

Categorías de flujo de trabajo

Los flujos de trabajo de infraestructuras están organizados en diferentes categorías para facilitar la localización del que desee.

Categoría	Descripción
Identidad y acceso	Estos flujos de trabajo le permiten gestionar la identidad y cómo se accede a Astra. Los recursos incluyen usuarios, credenciales y tokens.
Cucharones	Se pueden usar estos flujos de trabajo para crear y gestionar los bloques S3 que se utilizan para almacenar backups.
Reducida	Estos flujos de trabajo permiten añadir y mantener volúmenes y back-ends de almacenamiento.
De clúster	Es posible añadir clústeres de Kubernetes gestionados, que permiten proteger y admitir las aplicaciones que contienen.

Identidad y acceso

Enumere usuarios

Puede enumerar los usuarios definidos para una cuenta Astra específica.

1. Enumere los usuarios

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
OBTENGA	/Account/{accountID}/core/v1/users

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
incluya	Consulta	No	Opcionalmente, seleccione los valores que desea devolver en la respuesta.

Ejemplo curl: Devuelve todos los datos de todos los usuarios

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/users' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Ejemplo de curl: Devuelve el nombre, apellidos e id de todos los usuarios

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/users?include=first
Name,lastName,id' --header 'Accept: */*' --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": {}
}
```

Cucharones

Cucharones de lista

Puede enumerar los bloques de S3 definidos para una cuenta de Astra específica.

1. Enumere los cucharones

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
OBTENGA	/Account/{accountID}/topolog/v1/buckets

Ejemplo de curl: Devuelve todos los datos de todos los bloques

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/buckets'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Reducida

Enumerar los back-ends de almacenamiento

Es posible enumerar los back-ends de almacenamiento disponibles.

1. Enumere los cucharones

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
OBTENGA	/Account/{accountID}/topolog/v1/storageBackends

Ejemplo de curl: Devuelve todos los datos de todos los back-ends de almacenamiento

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/storageBackends
' --header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Ejemplo de resultado JSON

```
{
  "items": [
    {
      "backendCredentialsName": "10.191.77.177",
      "backendName": "myinchunhcluster-1",
      "backendType": "ONTAP",
      "backendVersion": "9.8.0",
      "configVersion": "Not applicable",
      "health": "Not applicable",
      "id": "46467c16-1585-4b71-8e7f-f0bc5ff9da15",
      "location": "nalab2",
      "metadata": {
        "createdBy": "4c483a7e-207b-4f9a-87b7-799a4629d7c8",
        "creationTimestamp": "2021-07-30T14:26:19Z",
        "modificationTimestamp": "2021-07-30T14:26:19Z"
      },
      "ontap": {
        "backendManagementIP": "10.191.77.177",
        "managementIPs": [
          "10.191.77.177",
          "10.191.77.179"
        ]
      },
      "protectionPolicy": "Not applicable",
      "region": "Not applicable",
      "state": "Running",
      "stateUnready": [],
      "type": "application/astra-storageBackend",
      "version": "1.0",
      "zone": "Not applicable"
    }
  ]
}
```


De clúster

Enumere los clústeres gestionados

Puede enumerar los clústeres de Kubernetes que gestiona actualmente Astra.

1. Enumere los clústeres

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
OBTENGA	/Account/{accountID}/topolog/v1/managedClusters

Ejemplo curl: Devuelve todos los datos de todos los clústeres

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/managedClusters
' --header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Flujos de trabajo de gestión

Antes de empezar

Puede utilizar estos flujos de trabajo como parte de la administración de las aplicaciones dentro de un clúster administrado de Astra.

> Estos flujos de trabajo pueden ampliarse y mejorarse mediante NetApp en cualquier momento, por lo que debe revisarlos periódicamente.

Preparación general

Antes de utilizar cualquiera de los flujos de trabajo de Astra, asegúrese de revisarlos "Prepárese para usar los flujos de trabajo".

Categorías de flujo de trabajo

Los flujos de trabajo de gestión están organizados en diferentes categorías para facilitar la localización del que desee.

Categoría	Descripción
Control de aplicaciones	Estos flujos de trabajo le permiten controlar las aplicaciones administradas y no administradas. Puede enumerar las aplicaciones, así como crear y eliminar una aplicación administrada.
Protección y recuperación	Puede utilizar estos flujos de trabajo para proteger las aplicaciones gestionadas mediante copias Snapshot y backups.
Clonar y restaurar aplicaciones	Este flujo de trabajo describe cómo clonar y restaurar las aplicaciones gestionadas.
Soporte técnico	Existen varios flujos de trabajo disponibles para depurar y dar soporte a las aplicaciones, así como al entorno de Kubernetes general.

Consideraciones adicionales

Existen varios aspectos adicionales que se deben tener en cuenta al utilizar los flujos de trabajo de gestión.

Clonar una aplicación

Hay algunos aspectos que hay que tener en cuenta a la hora de clonar una aplicación. Los parámetros descritos a continuación forman parte de la entrada JSON.

Identificador de clúster de origen

Valor de `sourceClusterID` identifica siempre el clúster en el que se instaló la aplicación original.

Identificador del clúster

Valor de `clusterID` identifica el clúster en el que se instalará la nueva aplicación.

- Al clonar dentro del mismo clúster, `clusterID` y.. `sourceClusterID` tienen el mismo valor.

- Al clonar entre clústeres, los dos valores son diferentes y. `clusterID` Debe ser el ID del clúster de destino.

Espacios de nombres

La `namespace` el valor debe ser diferente de la aplicación de origen original. Además, el espacio de nombres para el clon no puede existir y Astra lo creará.

Backups y snapshots

Opcionalmente, se puede clonar una aplicación desde una copia de Snapshot o backup existente mediante la `backupID` o. `snapshotID` parámetros. Si no proporciona una copia de seguridad o una copia Snapshot, Astra creará primero una copia de seguridad de la aplicación y, a continuación, clonará a partir de la copia de seguridad.

Restaurar una aplicación

Estos son algunos aspectos que deben tenerse en cuenta a la hora de restaurar una aplicación.

- Restaurar una aplicación es muy similar a la operación de clonado.
- Al restaurar una aplicación, debe proporcionar una copia de seguridad o una instantánea.

Control de aplicaciones

Enumere las aplicaciones no administradas

Puede enumerar las aplicaciones que actualmente no están gestionadas por Astra. Puede hacerlo como parte de la selección de una aplicación que se va a gestionar.

> El extremo DE REST utilizado en estos flujos de trabajo devuelve todas las aplicaciones Astra de forma predeterminada. Puede utilizar el `filter` Parámetro de consulta en la llamada a la API para solicitar sólo que se devuelvan las aplicaciones no administradas. Como alternativa, puede omitir el parámetro de filtro para devolver todas las aplicaciones y examinar la `managedState` en la salida para determinar qué aplicaciones se encuentran en `unmanaged` estado.

Enumere sólo las aplicaciones con managedState iguales a no administradas

Este flujo de trabajo utiliza la `filter` parámetro de consulta para devolver sólo las aplicaciones no administradas.

1. Enumere las aplicaciones no administradas

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
OBTENGA	/Account/{accountID}/topolog/v1/apps

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
filtro	Consulta	No	Utilice un filtro para especificar qué aplicaciones se deben devolver.
incluya	Consulta	No	Opcionalmente, seleccione los valores que desea devolver en la respuesta.

Ejemplo de curl: Devuelve el nombre, id y managedState para las aplicaciones no administradas

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/apps?filter=man
agedState%20eq%20'unmanaged'&include=name,id,managedState' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Ejemplo de resultado JSON

```
{
  "items": [
    [
      "maria",
      "eed19f78-0884-4792-bb7a-313258c6b0b1",
      "unmanaged"
    ],
    [
      "test-postgres-app",
      "1ee6235b-cda1-45cb-8d4c-630bdb8b41a5",
      "unmanaged"
    ],
    [
      "postgres1-postgresql",
      "e591ee59-ea90-4a9f-8e6c-d2b6e8647096",
      "unmanaged"
    ],
    [
      "kube-system",
      "077a2f73-4b51-4d04-8c6c-f63b3b069755",
      "unmanaged"
    ],
    [
      "trident",
      "5b6fc28f-e308-4653-b9d2-6d66a764d2e1",
      "unmanaged"
    ],
    [
      "postgres1-postgresql-clone",
      "06be05c5-763e-4d73-bd06-1f27f5f2e130",
      "unmanaged"
    ]
  ],
  "metadata": {}
}
```


Enumere todas las aplicaciones y seleccione las aplicaciones no administradas

Este flujo de trabajo devuelve todas las aplicaciones. Debe examinar el resultado para determinar cuáles no son administrados.

1. Enumerar todas las aplicaciones

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
OBTENGA	/Account/{accountID}/topolog/v1/apps

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
incluya	Consulta	No	Opcionalmente, seleccione los valores que desea devolver en la respuesta.

Ejemplo de curl: Devuelve todos los datos de todas las aplicaciones

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/apps' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Ejemplo de curl: Devuelve el nombre, id y managedState para todas las aplicaciones

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/apps?include=na
me,id,managedState' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Ejemplo de resultado JSON

```
{
  "items": [
    [
      "maria",
      "eed19f78-0884-4792-bb7a-313258c6b0b1",
      "unmanaged"
    ],
    [
      "mariadb-mariadb",
      "8da20fff-c69c-4170-bb0d-e4f91c5a1333",
      "managed"
    ],
    [
      "test-postgres-app",
      "1ee6235b-cda1-45cb-8d4c-630bdb8b41a5",
      "unmanaged"
    ],
    [
      "postgres1-postgresql",
      "e591ee59-ea90-4a9f-8e6c-d2b6e8647096",
      "unmanaged"
    ],
    [
      "kube-system",
      "077a2f73-4b51-4d04-8c6c-f63b3b069755",
      "unmanaged"
    ],
    [
      "trident",
      "5b6fc28f-e308-4653-b9d2-6d66a764d2e1",
      "unmanaged"
    ],
    [
      "postgres1-postgresql-clone",
      "06be05c5-763e-4d73-bd06-1f27f5f2e130",
      "unmanaged"
    ],
    [
      "davidns-postgres-app",
      "11e046b7-ec64-4184-85b3-debcc3b1da4d",
      "managed"
    ]
  ],
  "metadata": {}
}
```

2. Seleccione las aplicaciones no administradas

Revise la salida de la llamada API y seleccione manualmente las aplicaciones con `managedState` igual a. `unmanaged`.

Enumere las aplicaciones gestionadas

Puede enumerar las aplicaciones que gestiona actualmente Astra. Puede hacerlo como parte de la búsqueda de las instantáneas o backups de una aplicación específica.

1. Enumere las aplicaciones

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
OBTENGA	/Account/{accountID}/k8s/v1/managedApps

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
incluya	Consulta	No	Opcionalmente, seleccione los valores que desea devolver en la respuesta.

Ejemplo de curl: Devuelve todos los datos de todas las aplicaciones

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Ejemplo de curl: Devuelve el nombre, ID y estado de todas las aplicaciones

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps?include=
name,id,state' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Ejemplo de resultado JSON

```
{
  "items": [
    [
      "test-postgres-app",
      "1ee6235b-cda1-45cb-8d4c-630bdb8b41a5",
      "running"
    ]
  ],
  "metadata": {}
}
```

Consigue una aplicación gestionada

Puede recuperar todas las variables de recursos que describen una única aplicación administrada.

Antes de empezar

Debe tener el ID de la aplicación gestionada que desea recuperar. Si es necesario, puede usar el flujo de trabajo "Enumere las aplicaciones gestionadas" para localizar la aplicación.

1. Obtenga la aplicación

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
OBTENGA	/Accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
id de aplicación gestionada	Ruta	Sí	ID valor de la aplicación administrada que se va a recuperar.

Ejemplo de curl: Devuelve todos los datos de la aplicación

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Gestionar una aplicación

Puede crear una aplicación gestionada basada en una aplicación ya conocida por Astra. Cuando se gestiona una aplicación, se puede proteger realizando backups y snapshots regulares.

Antes de empezar

Debe tener el ID de la aplicación detectada que desea gestionar. Si es necesario, puede usar el flujo de trabajo "Enumere las aplicaciones no administradas" para localizar la aplicación.

1. Administrar la aplicación

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
PUBLICAR	/Account/{accountID}/k8s/v1/managedApps

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
JSON	Cuerpo	Sí	Proporciona los parámetros necesarios para identificar la aplicación que se va a gestionar. Vea el ejemplo siguiente.

Ejemplo de entrada JSON

```
{
  "type": "application/astra-managedApp",
  "version": "1.1",
  "id": "7da20fff-c69d-4270-bb0d-a4f91c5a1333"
}
```

Ejemplo de curl: Gestione una aplicación

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps'
--header 'Content-Type: application/astra-managedApp+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'  --d @JSONinput
```

Desgestionar una aplicación

Puede eliminar una aplicación gestionada cuando ya no sea necesaria. Al quitar una

aplicación administrada también se eliminan las programaciones asociadas.

Antes de empezar

Debe tener el ID de la aplicación gestionada que desea anular la gestión. Si es necesario, puede usar el flujo de trabajo "Enumere las aplicaciones gestionadas" para localizar la aplicación.

Los backups y las instantáneas de la aplicación no se eliminan automáticamente cuando se eliminan. Si ya no necesita los backups ni las snapshots, debe eliminarlos antes de eliminar la aplicación.

1. No se ha administrado la aplicación

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
ELIMINAR	/Accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
id de aplicación gestionada	Ruta	Sí	Identifica la aplicación administrada que se va a eliminar.

Ejemplo de curl: Eliminar una aplicación administrada

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Protección de aplicaciones

Enumere las instantáneas

Puede enumerar las instantáneas que se han realizado para una aplicación administrada específica.

Antes de empezar

Debe tener el ID de la aplicación gestionada para la que desea mostrar las instantáneas. Si es necesario, puede usar el flujo de trabajo "Enumere las aplicaciones gestionadas" para localizar la aplicación.

1. Enumere las instantáneas

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
OBTENGA	/Accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}/apps

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
id de aplicación gestionada	Ruta	Sí	Identifica la aplicación administrada a la que pertenecen las instantáneas enumeradas.
cuente	Consulta	No	Si `count=true` el número de instantáneas se incluye en la sección de metadatos de la respuesta.

Ejemplo de curl: Devuelve todas las instantáneas de la aplicación

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appSnaps' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Ejemplo curl: Devuelve todas las instantáneas de la aplicación y el recuento

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appSnaps?count=true' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>'
```

Ejemplo de resultado JSON

```
{
  "items": [
    {
      "id": "dc2974ae-f71d-4c81-91b5-f96cf72dc3ba",
      "metadata": {
        "createdBy": "fb093413-b6fc-4a64-a48a-afc32ada8537",
        "creationTimestamp": "2021-06-04T21:23:14Z",
        "modificationTimestamp": "2021-06-04T21:23:14Z",
        "labels": []
      },
      "snapshotAppAsset": "4547658d-cc06-4c1d-ad8a-4a05274d0db0",
      "snapshotCreationTimestamp": "2021-06-04T21:23:47Z",
      "name": "test-postgres-app-snapshot-20210604212213",
      "state": "completed",
      "stateUnready": [],
      "type": "application/astra-appSnap",
      "version": "1.0"
    }
  ],
  "metadata": {
    "count": 1
  }
}
```


Enumere los backups

Es posible enumerar los backups que se crearon para una aplicación gestionada específica.

Antes de empezar

Debe tener el ID de la aplicación gestionada para la que desea enumerar las copias de seguridad. Si es necesario, puede usar el flujo de trabajo "Enumere las aplicaciones gestionadas" para localizar la aplicación.

1. Enumere las copias de seguridad

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
OBTENGA	/Accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}/appbackups

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
id de aplicación gestionada	Ruta	Sí	Identifica la aplicación gestionada que posee las copias de seguridad mostradas.

Ejemplo curl: Devuelve todos los backups de la aplicación

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appBackups' --header 'Accept: */*' --header 'Authorization:
Bearer <API_TOKEN>'
```

Ejemplo de resultado JSON

```
{
  "items": [
    {
      "type": "application/astra-appBackup",
      "version": "1.0",
      "id": "ed39fdb0-12db-497b-9e46-20036c1fb0d2",
      "name": "mariadb-mariadb-backup-20210617175900",
      "state": "completed",
      "stateUnready": [],
      "bytesDone": 0,
      "percentDone": 100,
      "metadata": {
        "labels": [],
        "creationTimestamp": "2021-06-17T17:59:09Z",
        "modificationTimestamp": "2021-06-17T17:59:09Z",
        "createdBy": "fb093413-b6fc-4a64-a48a-afc32ada8537"
      }
    }
  ],
  "metadata": {}
}
```

Cree una instantánea para una aplicación gestionada

Puede crear una instantánea para una aplicación gestionada específica.

Antes de empezar

Debe tener el ID de la aplicación gestionada para la que desea crear una instantánea. Si es necesario, puede usar el flujo de trabajo "Enumere las aplicaciones gestionadas" para localizar la aplicación.

1. Crear una snapshot

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
PUBLICAR	/Accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}/apps

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
id de aplicación gestionada	Ruta	Sí	Identifica la aplicación administrada donde se creará la instantánea.
JSON	Cuerpo	Sí	Proporciona los parámetros para la instantánea. Vea el ejemplo siguiente.

Ejemplo de entrada JSON

```
{
  "type": "application/astra-appSnap",
  "version": "1.0",
  "name": "snapshot-david-1"
}
```

Ejemplo de curl: Cree una instantánea para la aplicación

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appSnaps' --header 'Content-Type: application/astra-appSnap+json'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --d
@JSONinput
```

Cree una copia de seguridad para una aplicación administrada

Es posible crear un backup para una aplicación gestionada específica. Puede utilizar el backup para restaurar o clonar la aplicación.

Antes de empezar

Debe tener el ID de la aplicación gestionada para la que desea crear una copia de seguridad. Si es necesario, puede usar el flujo de trabajo "Enumere las aplicaciones gestionadas" para localizar la aplicación.

1. Cree una copia de seguridad

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
PUBLICAR	/Accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}/appbackups

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
id de aplicación gestionada	Ruta	Sí	Identifica la aplicación administrada donde se creará el backup.
JSON	Cuerpo	Sí	Proporciona los parámetros para la copia de seguridad. Vea el ejemplo siguiente.

Ejemplo de entrada JSON

```
{
  "type": "application/astra-appBackup",
  "version": "1.0",
  "name": "backup-david-1"
}
```

Ejemplo curl: Cree una copia de seguridad para la aplicación

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appBackups' --header 'Content-Type: application/astra-
appBackup+json' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>' --d @JSONinput
```

Eliminar una copia de Snapshot

Es posible eliminar una instantánea asociada con una aplicación gestionada.

Antes de empezar

Debe tener lo siguiente:

- ID de la aplicación administrada a la que pertenece la instantánea. Si es necesario, puede usar el flujo de trabajo "Enumere las aplicaciones gestionadas" para localizar la aplicación.
- El ID de la copia de Snapshot que desea eliminar. Si es necesario, puede usar el flujo de trabajo "Enumere

las instantáneas" para localizar la snapshot.

1. Elimine la instantánea

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
ELIMINAR	/Accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}/apps/{appSnap_id}

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
id de aplicación gestionada	Ruta	Sí	Identifica la aplicación gestionada que posee la snapshot.
id de snapshot	Ruta	Sí	Identifica la snapshot que se eliminará.

Ejemplo de curva: Elimine una única instantánea de la aplicación

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appSnaps/<SNAPSHOT_ID>' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>'
```

Eliminar una copia de seguridad

Es posible eliminar un backup asociado a una aplicación gestionada.

Antes de empezar

Debe tener lo siguiente:

- ID de la aplicación gestionada a la que pertenece la copia de seguridad. Si es necesario, puede usar el flujo de trabajo "Enumere las aplicaciones gestionadas" para localizar la aplicación.
- ID del backup que desea eliminar. Si es necesario, puede usar el flujo de trabajo "Enumere los backups" para localizar la snapshot.

1. Eliminar la copia de seguridad

Realice la siguiente llamada de API de REST.

> Puede forzar la eliminación de una copia de seguridad fallida usando el encabezado de solicitud opcional como se describe a continuación.

Método HTTP	Ruta
ELIMINAR	/Accounts/{account_id}/k8s/v1/managedApps/{managedApp_id}/appbackups/{appBackup_id}

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
id de aplicación gestionada	Ruta	Sí	Identifica a la aplicación gestionada que posee el backup.
id de copia de seguridad	Ruta	Sí	Identifica el backup que se eliminará.
forzar eliminación	Encabezado	No	Se utiliza para forzar la eliminación de una copia de seguridad fallida.

Ejemplo curl: Eliminar un único backup para la aplicación

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appBackups/<BACKUP_ID>' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>'
```

Ejemplo de curl: Elimine una copia de seguridad única para la aplicación con la opción force

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<MANAGED
_APP_ID>/appBackups/<BACKUP_ID>' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>' --header 'Force-Delete: true'
```

Clonar y restaurar una aplicación

Clonar una aplicación gestionada

Puede crear una nueva aplicación clonando una aplicación administrada existente.

Antes de empezar

Tenga en cuenta lo siguiente acerca de este flujo de trabajo:

- No se utiliza una copia de seguridad o una instantánea de la aplicación
- La operación de clonado se ejecuta dentro del mismo clúster

Para clonar una aplicación en un clúster diferente, debe actualizar el `clusterId` Parámetro en la entrada JSON según sea apropiado para su entorno.

1. Seleccione la aplicación gestionada para clonar

Realice el flujo de trabajo "Enumere las aplicaciones gestionadas" y seleccione la aplicación que desea clonar. Se necesitan varios valores de recursos para la llamada DE REST utilizada para clonar la aplicación.

2. Clone la aplicación

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
PUBLICAR	/Account/{accountID}/k8s/v1/managedApps

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
JSON	Cuerpo	Sí	Proporciona los parámetros para la aplicación clonada. Vea el ejemplo siguiente.

Ejemplo de entrada JSON

```
{
  "type": "application/astra-managedApp",
  "version": "1.0",
  "name": "postgres1-postgresql-clone",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "davidns-postgres-app",
  "sourceAppID": "e591ee59-ea90-4a9f-8e6c-d2b6e8647096"
}
```

Ejemplo de curl: Clonar una aplicación

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps'
--header 'Content-Type: application/astra-managedApp+json' --header '*/*'
--header 'Authorization: Bearer <API_TOKEN>' --d @JSONinput
```

Clonar una aplicación administrada desde una instantánea

Puede crear una nueva aplicación clonándola a partir de una instantánea de la aplicación.

Antes de empezar

Tenga en cuenta lo siguiente acerca de este flujo de trabajo:

- Se utiliza una instantánea de aplicación
- La operación de clonado se ejecuta dentro del mismo clúster

> Para clonar una aplicación en un clúster diferente, debe actualizar el `clusterId` Parámetro en la entrada JSON según sea apropiado para su entorno.

1. Seleccione la aplicación gestionada para clonar

Realice el flujo de trabajo "Enumere las aplicaciones gestionadas" y seleccione la aplicación que desea clonar. Se necesitan varios valores de recursos para la llamada DE REST utilizada para clonar la aplicación.

2. Seleccione la instantánea que desea utilizar

Realice el flujo de trabajo "Enumere las instantáneas" y seleccione la copia de snapshot que desea usar.

3. Clone la aplicación

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
PUBLICAR	/Account/{accountID}/k8s/v1/managedApps

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
JSON	Cuerpo	Sí	Proporciona los parámetros para la aplicación clonada. Vea el ejemplo siguiente.

Ejemplo de entrada JSON

```
{
  "type": "application/astra-managedApp",
  "version": "1.0",
  "name": "postgres1-postgresql-clone",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "davidns-postgres-app",
  "snapshotID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb",
  "sourceAppID": "e591ee59-ea90-4a9f-8e6c-d2b6e8647096"
}
```

Ejemplo curl: Clone una aplicación de una snapshot

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps'
--header 'Content-Type: application/astra-managedApp+json' --header '*/*'
--header 'Authorization: Bearer <API_TOKEN>' --d @JSONinput
```

Clonar una aplicación gestionada desde un backup

Puede crear una nueva aplicación gestionada clonándola a partir de una copia de seguridad de la aplicación.

Antes de empezar

Tenga en cuenta lo siguiente acerca de este flujo de trabajo:

- Se utiliza una copia de seguridad de la aplicación
- La operación de clonado se ejecuta dentro del mismo clúster

> Para clonar una aplicación en un clúster diferente, debe actualizar el `clusterId` Parámetro en la entrada JSON según sea apropiado para su entorno.

1. Seleccione la aplicación gestionada para clonar

Realice el flujo de trabajo "Enumere las aplicaciones gestionadas" y seleccione la aplicación que desea clonar. Se necesitan varios valores de recursos para la llamada DE REST utilizada para clonar la aplicación.

2. Seleccione la copia de seguridad que desea utilizar

Realice el flujo de trabajo "Enumere los backups" y seleccione la copia de seguridad que desea usar.

3. Clone la aplicación

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
PUBLICAR	/Account/{accountID}/k8s/v1/managedApps

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
JSON	Cuerpo	Sí	Proporciona los parámetros para la aplicación clonada. Vea el ejemplo siguiente.

Ejemplo de entrada JSON

```
{
  "type": "application/astra-managedApp",
  "version": "1.0",
  "name": "postgres1-postgresql-clone",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "davidns-postgres-app",
  "backupID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb",
  "sourceAppID": "e591ee59-ea90-4a9f-8e6c-d2b6e8647096"
}
```

Ejemplo curl: Clone una aplicación desde un backup

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps'
--header 'Content-Type: application/astra-managedApp+json' --header '*/*'
--header 'Authorization: Bearer <API_TOKEN>' --d @JSONinput
```

Restaurar una aplicación gestionada desde una copia de seguridad

Puede restaurar una aplicación administrada creando una nueva aplicación a partir de una copia de seguridad.

1. Seleccione la aplicación administrada que desea restaurar

Realice el flujo de trabajo "Enumere las aplicaciones gestionadas" y seleccione la aplicación que desea clonar. Se necesitan varios valores de recursos para la llamada DE REST utilizada para clonar la aplicación.

2. Seleccione la copia de seguridad que desea utilizar

Realice el flujo de trabajo "Enumere los backups" y seleccione la copia de seguridad que desea usar.

3. Restaure la aplicación

Realice la siguiente llamada de API de REST. Debe proporcionar el ID para un backup (como se muestra a continuación) o una copia Snapshot.

Método HTTP	Ruta
PUESTO	/Account/{accountID}/k8s/v1/managedApps/{AppID}

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
JSON	Cuerpo	Sí	Proporciona los parámetros para la aplicación clonada. Vea el ejemplo siguiente.

Ejemplo de entrada JSON

```
{
  "type": "application/astra-managedApp",
  "version": "1.2",
  "backupID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"
}
```

Ejemplo de curl: Restaure una aplicación in situ a partir de un backup

```
curl --location -i --request PUT
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/managedApps/<APP_ID>
' --header 'Content-Type: application/astra-managedApp+json' --header
'*/*' --header 'ForceUpdate: true' --header 'Authorization: Bearer
<API_TOKEN>' --d @JSONinput
```

Soporte técnico

Enumere las notificaciones

Puede enumerar las notificaciones de una cuenta Astra específica. Esto se puede hacer como parte de la supervisión de la actividad del sistema o de la depuración de un problema.

1. Enumere las notificaciones

Realice la siguiente llamada de API de REST.

Método HTTP	Ruta
OBTENGA	/Account/{accountID}/core/v1/notificaciones

Parámetros de entrada adicionales

Además de los parámetros comunes con todas las llamadas API DE REST, en los ejemplos curl de este paso se incluyen los siguientes parámetros.

Parámetro	Tipo	Obligatorio	Descripción
filtro	Consulta	No	Opcionalmente, filtre las notificaciones que desea devolver en la respuesta.
incluya	Consulta	No	Opcionalmente, seleccione los valores que desea devolver en la respuesta.

Curl ejemplo: Devuelve todas las notificaciones

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/notifications'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Ejemplo de curl: Devuelve la descripción de las notificaciones cuando hay gravedad de advertencia

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/notifications?filte
r=severity%20eq%20'warning'&include=description' --header 'Accept: */*'
--header 'Authorization: Bearer <API_TOKEN>'
```

Ejemplo de resultado JSON

```
{
  "items": [
    [
      "Trident on cluster david-ie-00 has failed or timed out;
installation of the Trident operator failed or is not yet complete;
operator failed to reach an installed state within 300.00 seconds;
container trident-operator not found in operator deployment"
    ],
    [
      "Trident on cluster david-ie-00 has failed or timed out;
installation of the Trident operator failed or is not yet complete;
operator failed to reach an installed state within 300.00 seconds;
container trident-operator not found in operator deployment"
    ]
  ],
  "metadata": {}
}
```


Eliminar una aplicación fallida

Es posible que no pueda eliminar una aplicación gestionada si tiene una copia de seguridad o una instantánea en estado fallido. En este caso, puede eliminar manualmente la aplicación mediante el flujo de trabajo que se describe a continuación.

1. Seleccione la aplicación administrada que desea eliminar

Realice el flujo de trabajo "Enumere las aplicaciones gestionadas" y seleccione la aplicación que desea eliminar.

2. Enumere las copias de seguridad existentes de la aplicación

Realice el flujo de trabajo "Enumere los backups".

3. Eliminar todos los backups

Elimine todos los backups de aplicaciones realizando el flujo de trabajo "Eliminar una copia de seguridad" para cada backup de la lista.

4. Enumera las instantáneas existentes para la aplicación

Realice el flujo de trabajo "Enumere las instantáneas".

5. Elimine todas las instantáneas

Realice el flujo de trabajo "Eliminar una copia de Snapshot" de cada instantánea de la lista.

6. Retire la aplicación

Realice el flujo de trabajo "Desgestionar una aplicación" para eliminar la aplicación.

Uso de Python

Kit de desarrollo de software Astra Control Python de NetApp

NetApp Astra Control Python SDK es un paquete de código abierto que puede utilizar para automatizar la puesta en marcha de Astra Control. El paquete es también un recurso valioso para conocer la API REST de Astra Control, quizás como parte de la creación de su propia plataforma de automatización.

> Por su simplicidad, el SDK NetApp Astra Control Python se conoce como **SDK** durante el resto de esta página.

Dos herramientas de software relacionadas

El SDK incluye dos herramientas diferentes a través de las relacionadas que funcionan en diferentes niveles de abstracción al acceder a la API REST de Astra Control.

SDK de Astra

Astra SDK proporciona las funciones principales de la plataforma. Incluye un conjunto de clases Python que abstraen las llamadas de la API DE REST subyacente. Estas clases admiten acciones administrativas sobre diversos recursos de Astra Control, como aplicaciones, copias de seguridad, instantáneas y clusters.

El Astra SDK forma parte del paquete y se proporciona en un único `astraSDK.py` archivo. Puede importar este archivo al entorno y utilizar las clases directamente.

> * NetApp Astra Control Python SDK* (o solo SDK) es el nombre de todo el paquete. **Astra SDK** se refiere a las clases básicas de Python en un único archivo `astraSDK.py`.

Guión del kit de herramientas

Además del archivo Astra SDK, la `toolkit.py` también está disponible el guión. Este script funciona a un nivel más alto de abstracción al proporcionar acceso a acciones administrativas discretas definidas internamente como funciones Python. La secuencia de comandos importa Astra SDK y realiza llamadas a las clases según sea necesario.

Cómo acceder

Puede acceder al SDK de las siguientes maneras.

Paquete Python

SDK está disponible en "Índice de paquetes Python" con el nombre * netapp-astra-kits*. Al paquete se le asigna un número de versión y se seguirá actualizando según sea necesario. Debe utilizar la utilidad de administración de paquetes **PiP** para instalar el paquete en su entorno.

Consulte "PyPI: Kit de desarrollo de software Astra Control Python de NetApp" si quiere más información.

Código fuente GitHub

El código fuente del SDK también está disponible en GitHub. El repositorio incluye lo siguiente:

- `astraSDK.py` (Astra SDK con clases Python)
- `toolkit.py` (script basado en funciones de mayor nivel)
- Instrucciones y requisitos de instalación detallados
- Scripts de instalación
- Documentación adicional

Puede clonar el "GitHub: NetApp/netapp-astra-kits" repositorio en el entorno local.

Requisitos básicos y de instalación

Hay varias opciones y requisitos que se deben considerar como parte de la instalación del paquete y como parte de la preparación para utilizarlo.

Resumen de las opciones de instalación

Puede instalar el SDK de una de las siguientes maneras:

- Utilice PIP para instalar el paquete de PyPI en su entorno Python
- Clone el repositorio de Git Hub y:
  - Ponga en marcha el paquete como contenedor Docker (que incluye todo lo que necesita)
  - Copie los dos archivos principales de Python para que puedan acceder a su código de cliente Python

Consulte las páginas PyPI y GitHub para obtener más información.

Requisitos para el entorno de Astra Control

Ya sea utilizando directamente las clases Python en Astra SDK o las funciones de `toolkit.py` Script, en última instancia, accederá a la API DE REST en una implementación de Astra Control. Gracias a esto, necesitará una cuenta Astra junto con un token de API. Consulte "Antes de empezar" Y las otras páginas de la sección **Introducción** de esta documentación para obtener más información.

Requisitos del SDK de Astra Control Python de NetApp

El SDK tiene varios requisitos previos relacionados con el entorno local de Python. Por ejemplo, debe utilizar Python 3.5 o posterior. Además, hay varios paquetes Python que son necesarios. Consulte la página del repositorio de GitHub o la página del paquete PyPI para obtener más información.

Resumen de recursos útiles

Estos son algunos de los recursos que necesitará para comenzar.

- "PyPI: Kit de desarrollo de software Astra Control Python de NetApp"
- "GitHub: NetApp/netapp-astra-kits"

Python nativo

Antes de empezar

Python es un lenguaje de desarrollo popular especialmente para la automatización del centro de datos. Antes de utilizar las características nativas de Python junto con varios paquetes comunes, debe preparar el entorno y los archivos de entrada necesarios.

Además de acceder a la API DE REST de Astra Control directamente con Python, NetApp también ofrece un paquete de herramientas que abstrae la API y elimina algunas de las complejidades. Consulte "Kit de desarrollo de software Astra Control Python de NetApp" si quiere más información.

Preparar el entorno de

A continuación se describen los requisitos básicos de configuración para ejecutar los scripts de Python.

Python 3

Necesita tener instalada la última versión de Python 3.

Bibliotecas adicionales

Las bibliotecas **Requests** y **urllib3** deben estar instaladas. Puede utilizar pip u otra herramienta de gestión Python según sea necesario para su entorno.

Acceso a la red

La estación de trabajo donde se ejecuten las secuencias de comandos debe tener acceso a la red y poder llegar a Astra Control. Cuando utilice Astra Control Service, debe estar conectado a Internet y poder conectarse al servicio en `https://astra.netapp.io`.

Información de identidad

Necesita una cuenta Astra válida con el identificador de cuenta y el token de API. Consulte "Obtenga un token de API" si quiere más información.

Cree los archivos de entrada JSON

Los scripts Python se basan en la información de configuración contenida en los archivos de entrada JSON. A continuación se proporcionan archivos de ejemplo.

Debe actualizar las muestras según sea necesario para su entorno.

Información de identidad

El siguiente archivo contiene el token de la API y la cuenta de Astra. Debe pasar este archivo a los scripts de Python mediante `-i` (o. `--identity`) Parámetro CLI.

```
{
  "api_token": "kH4CA_uVIa8q9UuPzhJaAHaGlaR7-no901DkkrVjIXk=",
  "account_id": "5131dfdf-03a4-5218-ad4b-fe84442b9786"
}
```

Enumere las aplicaciones gestionadas

Puede utilizar la siguiente secuencia de comandos para enumerar las aplicaciones gestionadas de su cuenta Astra.

Consulte "Antes de empezar" Por ejemplo del archivo de entrada JSON requerido.

```
#!/usr/bin/env python3
##-----------------------------------------------------------------------
------
#
# Usage: python3 list_man_apps.py -i identity_file.json
#
# (C) Copyright 2021 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 managed 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/v1/managedApps"

    # 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-managedApp+json"
    req_headers['Accept'] = "application/astra-managedApp+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 managed 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 managed 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)
```

Referencia de API

Puede acceder a los detalles de todas las llamadas a la API REST de Astra Control, incluidos los métodos HTTP, los parámetros de entrada y las respuestas. Esta referencia completa es útil cuando se desarrollan aplicaciones de automatización mediante la API DE REST.

> La documentación de referencia de la API DE REST se proporciona actualmente con Astra Control y está disponible en línea.

Antes de empezar

Necesita una cuenta para Astra Control Center o Astra Control Service.

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. En la parte superior de la página, haga clic en la dirección URL que aparece en **Documentación de API**.
4. Vuelva a introducir las credenciales de su cuenta si se le solicita.

Recursos adicionales

Hay recursos adicionales a los que puede acceder para obtener ayuda y obtener más información sobre los servicios cloud de NetApp y la compatibilidad, así como sobre conceptos generales DE REST y cloud.

Astra

- "Documentación de Astra Control Center 22.04"

  Documentación para la versión actual del software Astra Control Center implementado en las instalaciones del cliente.

- "Documentación de Astra Control Service"

  Documentación para el lanzamiento actual del software Astra Control Service disponible en la nube pública.

- "Documentación de Astra Trident"

  Documentación para la versión actual del software Astra Trident, un orquestador de almacenamiento de código abierto que mantiene NetApp.

- "Documentación de la familia Astra"

  Ubicación central para acceder a toda la documentación de Astra para puestas en marcha tanto en las instalaciones como en cloud público.

Recursos de cloud de NetApp

- "Soluciones cloud de NetApp"

  Sitio central de las soluciones cloud de NetApp.

- "Consola Cloud Central de NetApp"

  Consola de servicio Cloud Central de NetApp con inicio de sesión.

- "Soporte de NetApp"

  Acceda a herramientas de solución de problemas, documentación y asistencia de soporte técnico.

Conceptos DE REST y cloud

- Doctorado "disertación" Por Roy Fiding

  En esta publicación se introdujo y se estableció el modelo de desarrollo de aplicaciones DE REST.

- "Auth0"

  Se trata del servicio de plataforma de autenticación y autorización utilizado por el servicio Astra para el

acceso a la Web.

- "Editor RFC"

  Fuente autoritativa para estándares web e Internet que se mantiene como una colección de documentos RFC con números únicos.

Versiones anteriores de la documentación de Astra Control Automation

Puede acceder a la documentación de automatización de versiones anteriores de Astra Control en los siguientes enlaces.

- "Documentación de Astra Control Automation 21.12"
- "Documentación de Astra Control Automation 21.08"

Avisos legales

Los avisos legales proporcionan acceso a las declaraciones de copyright, marcas comerciales, patentes y mucho más.

Derechos de autor

"https://www.netapp.com/company/legal/copyright/"

Marcas comerciales

NETAPP, el logotipo de NETAPP y las marcas enumeradas en la página de marcas comerciales de NetApp son marcas comerciales de NetApp, Inc. Los demás nombres de empresas y productos son marcas comerciales de sus respectivos propietarios.

"https://www.netapp.com/company/legal/trademarks/"

Estadounidenses

Puede encontrar una lista actual de las patentes propiedad de NetApp en:

https://www.netapp.com/pdf.html?item=/media/11887-patentspage.pdf

Política de privacidad

"https://www.netapp.com/company/legal/privacy-policy/"

Licencia Astra Control API

https://docs.netapp.com/us-en/astra-automation/media/astra-api-license.pdf