Documentação do Astra Control Automation 22,08

Astra Automation 22.08

NetApp
October 23, 2024

Índice

Documentação do Astra Control Automation 22,08

Notas de lançamento

Sobre esta versão

A documentação neste local descreve a API REST Astra Control e as tecnologias de automação relacionadas disponíveis na versão de agosto de 2022 (22,08) do Astra Control. Em particular, esta versão da API REST está incluída nas versões 22,08 correspondentes do Astra Control Center e do Astra Control Service.

Consulte as páginas e sites a seguir para obter mais informações sobre esta versão, bem como versões anteriores:

- "Novidades da API REST do Astra Control"
- "Recursos REST e endpoints"
- "Documentação do Astra Control Center 22,08"
- "Documentação do Astra Control Service"
- "Versões anteriores da documentação do Astra Automation"

Siga-nos no Twitter. Envie feedback sobre a documentação tornando-se um "Colaborador do GitHub" ou enviando um e-mail para NetApp.com.

Novidades da API REST do Astra Control

O NetApp atualiza regularmente a API REST do Astra Control para oferecer novos recursos, melhorias e correções de bugs.

10 de agosto de 2022 (22,08)

Esta versão inclui uma expansão e atualização da API REST, bem como recursos administrativos e de segurança aprimorados.

Recursos novos e aprimorados do Astra

Três novos tipos de recursos foram adicionados: **Certificado**, **Grupo** e **AppMirror**. Além disso, as versões de vários recursos existentes foram atualizadas.

Autenticação LDAP

Opcionalmente, é possível configurar o Astra Control Center para integração com um servidor LDAP para autenticar usuários selecionados do Astra. Consulte "Configuração LDAP" para obter mais informações.

Gancho de execução melhorado

O suporte para ganchos de execução foi adicionado com a versão Astra Control 21,12. Além dos ganchos de execução pré-snapshot e pós-snapshot existentes, agora você pode configurar os seguintes tipos de ganchos de execução com a versão 22,08:

- Pré-backup

- Pós-backup
- Pós-restauração

O Astra Control agora também permite que o mesmo script seja usado para vários ganchos de execução.

Replicação de aplicativos usando o SnapMirror

Agora é possível replicar alterações de dados e aplicações entre clusters usando a tecnologia NetApp SnapMirror. Esse aprimoramento pode ser usado para melhorar a continuidade dos negócios e os recursos de recuperação.

Informações relacionadas

- "Astra Control Center: Novidades"
- "Astra Control Service: Novidades"

26 de abril de 2022 (22,04)

Esta versão inclui uma expansão e atualização da API REST, bem como recursos administrativos e de segurança aprimorados.

Recursos novos e aprimorados do Astra

Dois novos tipos de recursos foram adicionados: **Pacote** e **Upgrade**. Além disso, as versões de vários recursos existentes foram atualizadas.

RBAC aprimorado com granularidade de namespace

Ao vincular uma função a um usuário associado, você pode limitar os namespaces aos quais o usuário tem acesso. Consulte a referência **Role Binding API** e "Segurança RBAC" para obter mais informações.

Remoção do balde

Você pode remover um balde quando ele não for mais necessário ou não estiver funcionando corretamente.

Suporte para Cloud Volumes ONTAP

Agora, o Cloud Volumes ONTAP é compatível como um back-end de storage.

Melhorias adicionais do produto

Há vários aprimoramentos adicionais nas duas implementações de produtos Astra Control, incluindo:

- Entrada genérica para Astra Control Center
- Cluster privado em AKS
- Suporte para Kubernetes 1,22
- Suporte ao portfólio VMware Tanzu

Consulte a página **Novidades** nos sites de documentação do Astra Control Center e do Astra Control Service.

Informações relacionadas

- "Astra Control Center: Novidades"

- "Astra Control Service: Novidades"

14 de dezembro de 2021 (21,12)

Esta versão inclui uma expansão da API REST, juntamente com uma alteração na estrutura de documentação para dar suporte à evolução do Astra Control com as futuras atualizações de versões.

Documentação separada do Astra Automation para cada versão do Astra Control

Todas as versões do Astra Control incluem uma API REST distinta que foi aprimorada e adaptada aos recursos da versão específica. A documentação para cada versão da API REST do Astra Control agora está disponível em seu próprio site dedicado, juntamente com o repositório de conteúdo associado do GitHub. O site principal do doc "Automação do Astra Control" sempre contém a documentação para a versão mais atual. "Versões anteriores da documentação do Astra Control Automation"Consulte para obter informações sobre versões anteriores.

Expansão dos tipos de recursos REST

O número de tipos de recursos REST continuou a se expandir com ênfase em ganchos de execução e backends de armazenamento. Os novos recursos incluem: Conta, gancho de execução, fonte de gancho, substituição de gancho de execução, nó de cluster, back-end de storage gerenciado, namespace, dispositivo de storage e nó de storage. Consulte "Recursos" para obter mais informações.

O NetApp já está disponível

O NetApp é um pacote de código aberto que facilita o desenvolvimento de código de automação para seu ambiente Astra Control. No centro está o SDK Astra, que inclui um conjunto de classes para abstrair a complexidade das chamadas de API REST. Há também um script de kit de ferramentas para executar tarefas administrativas específicas, envolvendo e abstraindo as classes Python. Consulte "O NetApp já está disponível" para obter mais informações.

5 de agosto de 2021 (21,08)

Esta versão inclui a introdução de um novo modelo de implantação Astra e uma grande expansão da API REST.

Modelo de implantação do Astra Control Center

Além da oferta existente do Astra Control Service fornecida como serviço de nuvem pública, esta versão também inclui o modelo de implantação no local do Astra Control Center. Você pode instalar o Astra Control Center no seu local para gerenciar seu ambiente Kubernetes local. Os dois modelos de implantação do Astra Control compartilham a mesma API REST, com pequenas diferenças observadas conforme necessário na documentação.

Expansão dos tipos de recursos REST

O número de recursos acessíveis por meio da API REST Astra Control foi muito ampliado, com muitos dos novos recursos fornecendo a base para a oferta do Astra Control Center no local. Os novos recursos incluem: ASUP, direito, recurso, licença, configuração, assinatura, bucket, nuvem, cluster, cluster gerenciado, back-end de storage e classe de storage. Consulte "Recursos" para obter mais informações.

Pontos de extremidade adicionais compatíveis com a implantação do Astra

Além dos recursos REST expandidos, há vários outros pontos de extremidade de API novos disponíveis para dar suporte à implantação do Astra Control.

Suporte ao OpenAPI
: Os endpoints OpenAPI fornecem acesso ao documento JSON OpenAPI atual e a outros recursos relacionados.

Suporte ao OpenMetrics
: Os endpoints OpenMetrics fornecem acesso às métricas da conta por meio do recurso OpenMetrics.

15 de abril de 2021 (21,04)

Esta versão inclui os seguintes novos recursos e aprimoramentos.

Introdução da API REST

A API REST do Astra Control está disponível para uso com a oferta do Astra Control Service. Ele foi criado com base em TECNOLOGIAS REST e nas melhores práticas atuais. A API fornece uma base para a automação das implantações do Astra e inclui os recursos e benefícios a seguir.

Recursos
: Existem quatorze tipos de recursos REST disponíveis.

Acesso ao token de API
: O acesso à API REST é fornecido por meio de um token de acesso à API que você pode gerar na interface de usuário da Web Astra. O token de API fornece acesso seguro à API.

Suporte para coleções
: Há um conjunto rico de parâmetros de consulta que podem ser usados para acessar as coleções de recursos. Algumas das operações suportadas incluem filtragem, classificação e paginação.

Problemas conhecidos

Você deve analisar todos os problemas conhecidos da versão atual relacionada à API REST do Astra Control. Os problemas conhecidos identificam problemas que podem impedi-lo de usar o produto com sucesso.

Não há novos problemas conhecidos com a versão 22,08 da API REST do Astra Control. Os problemas descritos abaixo foram descobertos em versões anteriores e ainda são aplicáveis com a versão atual.

Nem todos os dispositivos de armazenamento em um nó de armazenamento de back-end são descobertos

Ao emitir uma chamada de API REST para recuperar os dispositivos de storage definidos em um nó de storage, somente os dispositivos Astra Data Store são descobertos. Nem todos os dispositivos são devolvidos.

Visão geral dos recursos e benefícios

O Astra Control Center e o Astra Control Service fornecem uma API REST comum que você pode acessar diretamente por meio de uma linguagem de programação ou utilitário, como o Curl. Os principais destaques e benefícios da API são apresentados abaixo.

Para acessar a API REST, você precisa primeiro fazer login na interface de usuário da Web Astra e gerar um token de API. Você deve incluir o token em cada solicitação de API.

Desenvolvido com base na TECNOLOGIA REST

A API Astra Control foi criada usando a TECNOLOGIA REST e as práticas recomendadas atuais. A tecnologia principal inclui HTTP, JSON e RBAC.

Suporte para os dois modelos de implantação do Astra Control

O Astra Control Service é usado no ambiente de nuvem pública enquanto o Astra Control Center é para suas implantações locais. Há uma API REST compatível com ambos os modelos de implantação.

Limpar o mapeamento entre os recursos de endpoint REST e o modelo de objeto

Os pontos de extremidade REST externos usados para acessar o mapa de recursos a um modelo de objeto consistente mantido internamente pelo serviço Astra. O modelo de objeto é projetado usando modelagem de relacionamento de entidade (ER) que ajuda a definir claramente as ações e respostas da API.

Conjunto rico de parâmetros de consulta

A API REST fornece um conjunto rico de parâmetros de consulta que você pode usar para acessar as coleções de recursos. Algumas das operações suportadas incluem filtragem, classificação e paginação.

Alinhamento com a IU da Web Astra Control

O design da interface de usuário da Web Astra está alinhado com a API REST e, portanto, há consistência entre os dois caminhos de acesso e a experiência do usuário.

Depuração robusta e dados de determinação de problemas

A API REST do Astra Control fornece um recurso robusto de depuração e determinação de problemas, incluindo eventos do sistema e notificações do usuário.

Processos de fluxo de trabalho

Um conjunto de fluxos de trabalho é fornecido para ajudar no desenvolvimento de seu código de automação. Os fluxos de trabalho são organizados em duas categorias principais: Infraestrutura e gerenciamento.

Base para tecnologias avançadas de automação

Além de acessar a API REST diretamente, você pode usar outras tecnologias de automação baseadas na API REST.

Parte da documentação da família Astra

A documentação do Astra Control Automation é parte da documentação maior da família Astra. Consulte "Documentação do Astra" para obter mais informações.

Comece agora

Antes de começar

Você pode se preparar rapidamente para começar a usar a API REST do Astra Control revisando as etapas abaixo.

Ter credenciais de conta Astra

Você precisará de credenciais Astra para fazer login na interface de usuário da Web Astra e gerar um token de API. Com o Astra Control Center, você gerencia essas credenciais localmente. Com o Astra Control Service, as credenciais da conta são acessadas pelo serviço **Auth0**.

Familiarize-se com os conceitos básicos do Kubernetes

Você deve estar familiarizado com vários conceitos básicos do Kubernetes. Consulte "Conceitos básicos do Kubernetes" para obter mais informações.

Rever conceitos REST e implementação

Certifique-se de consultar "Implementação principal do REST"informações sobre conceitos REST e detalhes sobre como a API REST do Astra Control foi projetada.

Obtenha mais informações

Você deve estar ciente dos recursos de informações adicionais, conforme sugerido no "Recursos adicionais".

Obtenha um token de API

Você precisa obter um token da API Astra para usar a API REST do Astra Control.

Introdução

Um token de API identifica o chamador para o Astra e deve ser incluído em todas as chamadas da API REST.

- Você pode gerar um token de API usando a interface de usuário da Web Astra.
- A identidade do usuário carregada com o token é determinada pelo usuário que cria o token.
- O token deve ser incluído no `Authorization` cabeçalho de solicitação HTTP.
- Um token nunca expira depois que ele é criado.
- Você pode revogar um token na interface de usuário da Web Astra.

Informações relacionadas

- "Revogar um token de API"

Crie um token da API Astra

As etapas a seguir descrevem como criar um token da API Astra.

Antes de começar

Você precisa de credenciais para uma conta Astra.

Sobre esta tarefa

Essa tarefa gera um token de API na interface Web do Astra. Você também deve recuperar o ID da conta que também é necessário ao fazer chamadas de API.

Passos

1. Faça login no Astra usando suas credenciais de conta.

   Acesse o seguinte site do Astra Control Service: "https://astra.netapp.io"

2. Clique no ícone de figura no canto superior direito da página e selecione **Acesso à API**.
3. Clique em **Generate API token** na página e na janela pop-up clique em **Generate API token**.
4. Clique no ícone para copiar a string de token para a área de transferência e salvá-la no editor.
5. Copie e salve o ID da conta que está disponível na mesma página.

Depois de terminar

Quando você acessa a API REST do Astra Control por meio do Curl ou de uma linguagem de programação, você deve incluir o token portador da API no cabeçalho da solicitação HTTP `Authorization`.

Olá mundo

Você pode emitir um comando curl simples na CLI da estação de trabalho para começar a usar a API REST Astra Control e confirmar sua disponibilidade.

Antes de começar

O utilitário Curl deve estar disponível na estação de trabalho local. Você também deve ter um token de API e o identificador de conta associado. Consulte "Obtenha um token de API" para obter mais informações.

Curl exemplo

O seguinte comando Curl recupera uma lista de usuários Astra. Forneça o <ACCOUNT_ID> e o <API_TOKEN> apropriados, conforme indicado.

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

Exemplo de saída JSON

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

Prepare-se para usar os fluxos de trabalho

Você deve estar familiarizado com a organização e o formato dos workflows do Astra antes de usá-los em uma implantação ao vivo.

Introdução

Um *workflow* é uma sequência de uma ou mais etapas necessárias para realizar uma tarefa ou objetivo administrativo específico. Cada etapa em um fluxo de trabalho do Astra Control é uma das seguintes:

- Chamada de API REST (com detalhes como exemplos curl e JSON)
- Invocação de outro fluxo de trabalho Astra
- Tarefa relacionada diversos (como tomar uma decisão de projeto necessária)

Os fluxos de trabalho incluem as etapas principais e os parâmetros necessários para realizar cada tarefa. Eles fornecem um ponto de partida para personalizar seu ambiente de automação.

Parâmetros de entrada comuns

Os parâmetros de entrada descritos abaixo são comuns a todas as amostras curl usadas para ilustrar uma chamada de API REST.

> Como esses parâmetros de entrada são universalmente necessários, eles não são descritos mais detalhadamente nos fluxos de trabalho individuais. Se forem utilizados parâmetros de entrada adicionais para um exemplo de curl específico, estes são descritos na secção **parâmetros de entrada adicionais**.

Parâmetros do caminho

O caminho do endpoint usado com cada chamada de API REST inclui os seguintes parâmetros. Consulte também "Formato de URL"para obter mais informações.

ID da conta

Esse é o valor UUIDv4 que identifica a conta Astra onde a operação da API é executada. "Obtenha um token de API"Consulte para obter mais informações sobre como localizar o ID da sua conta.

Cabeçalhos de solicitação

Existem vários cabeçalhos de solicitação que você pode precisar incluir dependendo da chamada da API REST.

Autorização

Todas as chamadas de API nos fluxos de trabalho precisam de um token de API para identificar o usuário. Você deve incluir o token no `Authorization` cabeçalho da solicitação. Consulte "Obtenha um token de API" para obter mais informações sobre como gerar um token de API.

Tipo de conteúdo

Com as solicitações HTTP POST e PUT em que JSON é incluído no corpo da solicitação, você deve declarar o tipo de Mídia com base no recurso Astra. Por exemplo, você pode incluir o cabeçalho `Content-Type: application/astra-appSnap+json` ao criar um snapshot para um aplicativo gerenciado.

Aceitar

Você pode declarar o tipo de Mídia específico do conteúdo esperado na resposta com base no recurso Astra. Por exemplo, você pode incluir o cabeçalho `Accept: application/astra-appBackup+json` ao listar os backups de um aplicativo gerenciado. No entanto, para simplificar, as amostras curl nos fluxos de trabalho aceitam todos os tipos de Mídia.

Apresentação de tokens e identificadores

O token API e outros valores de ID usados com os exemplos curl são opacos sem significado discernível. E assim, para melhorar a legibilidade das amostras, os valores reais de token e ID não são usados. Em vez disso, palavras-chave reservadas menores são usadas que tem vários benefícios:

- As amostras curl e JSON são mais claras e fáceis de entender.
- Como todas as palavras-chave usam o mesmo formato com colchetes e letras maiúsculas, você pode identificar rapidamente o local e o conteúdo a ser inserido ou extraído.
- Nenhum valor é perdido porque os parâmetros originais não podem ser copiados e usados com uma implantação real.

Aqui estão algumas das palavras-chave reservadas comuns usadas nos exemplos curl. Esta lista não é exaustiva e palavras-chave adicionais são usadas conforme necessário. Seu significado deve ser óbvio com base no contexto.

Palavra-chave	Tipo	Descrição
<ACCOUNT_ID>	Caminho	O valor UUIDv4 identificando a conta onde a operação da API é executada.
<API_TOKEN>	Colhedor	O token do portador identificando e autorizando o chamador.

Palavra-chave	Tipo	Descrição
<APP_ID>	Caminho	O valor UUIDv4 identificando o aplicativo para a chamada API.

Categorias de fluxo de trabalho

Há duas grandes categorias de workflows do Astra disponíveis com base no seu modelo de implantação. Se você estiver usando o Astra Control Center, comece com os workflows de infraestrutura e prossiga para os workflows de gerenciamento. Ao usar o Astra Control Service, você costuma ir diretamente aos workflows de gerenciamento.

As amostras curl nos fluxos de trabalho usam o URL do Astra Control Service. Você precisa alterar o URL ao usar o Astra Control Center no local, conforme apropriado para seu ambiente.

Workflows de infraestrutura

Esses workflows lidam com a infraestrutura Astra, incluindo credenciais, buckets e back-ends de storage. Elas são necessárias com o Astra Control Center, mas na maioria dos casos também podem ser usadas com o Astra Control Service. Os fluxos de trabalho concentram-se nas tarefas necessárias para estabelecer e manter um cluster gerenciado do Astra.

Fluxos de trabalho de gerenciamento

Você pode usar esses fluxos de trabalho depois de ter um cluster gerenciado. Os workflows se concentram em operações de suporte e proteção de aplicativos, como backup, restauração e clonagem de um aplicativo.

Conceitos básicos do Kubernetes

Há vários conceitos do Kubernetes que são relevantes ao usar a API REST do Astra.

Objetos

Os objetos mantidos em um ambiente Kubernetes são entidades persistentes que representam a configuração do cluster. Esses objetos descrevem coletivamente o estado do sistema, incluindo a carga de trabalho do cluster.

Namespaces

Namespaces fornecem uma técnica para isolar recursos em um único cluster. Essa estrutura organizacional é útil ao dividir os tipos de trabalho, usuários e recursos. Objetos com um escopo *namespace* precisam ser exclusivos dentro do namespace, enquanto aqueles com um escopo *cluster* devem ser exclusivos em todo o cluster.

Etiquetas

Os rótulos podem ser associados aos objetos Kubernetes. Eles descrevem atributos usando pares de chave-valor e podem impor uma organização arbitrária no cluster que pode ser útil para uma organização, mas estão fora da operação principal do Kubernetes.

Implementação principal do REST

Serviços web REST

Representational State Transfer (REST) é um estilo para a criação de aplicações web distribuídas. Quando aplicado ao design de uma API de serviços da Web, ele estabelece um conjunto de tecnologias e práticas recomendadas principais para expor recursos baseados em servidor e gerenciar seus estados. Embora O REST forneça uma base consistente para o desenvolvimento de aplicativos, os detalhes de cada API podem variar com base nas escolhas específicas de design. Você deve estar ciente das características da API REST do Astra Control antes de usá-la com uma implantação ativa.

Recursos e representação do Estado

Os recursos são os componentes básicos de um sistema baseado na Web. Ao criar um aplicativo REST de serviços da Web, as tarefas iniciais de design incluem:

- Identificação de recursos baseados em sistema ou servidor

  Cada sistema usa e mantém recursos. Um recurso pode ser um arquivo, transação comercial, processo ou entidade administrativa. Uma das primeiras tarefas no projeto de um aplicativo baseado em serviços web REST é identificar os recursos.

- Definição de estados de recursos e operações de estado associadas

  Os recursos estão sempre em um de um número finito de estados. Os estados, bem como as operações associadas usadas para afetar as mudanças de estado, devem ser claramente definidos.

Pontos de extremidade URI

Todos os recursos REST devem ser definidos e disponibilizados usando um esquema de endereçamento bem definido. Os endpoints onde os recursos estão localizados e identificados usam um URI (Uniform Resource Identifier). O URI fornece uma estrutura geral para criar um nome exclusivo para cada recurso na rede. O Uniform Resource Locator (URL) é um tipo de URI usado com serviços da Web para identificar e acessar recursos. Os recursos são normalmente expostos em uma estrutura hierárquica semelhante a um diretório de arquivos.

Mensagens HTTP

O Hypertext Transfer Protocol (HTTP) é o protocolo usado pelo cliente e servidor de serviços da Web para trocar mensagens de solicitação e resposta sobre os recursos. Como parte do projeto de um aplicativo de serviços da Web, os métodos HTTP são mapeados para os recursos e as ações de gerenciamento de estado correspondentes. HTTP está sem estado. Portanto, para associar um conjunto de solicitações e respostas relacionadas como parte de uma transação, informações adicionais devem ser incluídas nos cabeçalhos HTTP carregados com os fluxos de dados de solicitação e resposta.

Formatação JSON

Embora as informações possam ser estruturadas e transferidas entre um cliente e um servidor de serviços da Web de várias maneiras, a opção mais popular é JavaScript Object Notation (JSON). JSON é um padrão da indústria para representar estruturas de dados simples em texto simples e é usado para transferir informações de estado descrevendo os recursos. A API REST DO Astra Control usa JSON para formatar os dados transportados no corpo de cada solicitação e resposta HTTP.

Recursos e coleções

A API REST do Astra Control fornece acesso a instâncias de recursos e coleções de instâncias de recursos.

> Conceitualmente um resource * REST é semelhante a um objeto * conforme definido com as linguagens e sistemas de programação orientada a objetos (OOP). Às vezes, esses termos são usados de forma intercambiável. Mas, em geral, "recurso" é preferível quando usado no contexto da API REST externa enquanto "objeto" é usado para os dados de instância com estado correspondente armazenados no servidor.

Atributos dos recursos do Astra

A API REST do Astra Control está em conformidade com os princípios de design RESTful. Cada instância de recurso Astra é criada com base em um tipo de recurso bem definido. Um conjunto de instâncias de recursos do mesmo tipo é chamado de **Collection**. As chamadas API atuam sobre recursos individuais ou coleções de recursos.

Tipos de recursos

Os tipos de recursos incluídos na API REST do Astra Control têm as seguintes características:

- Cada tipo de recurso é definido usando um esquema (normalmente em JSON)
- Cada esquema de recursos inclui o tipo de recurso e a versão
- Os tipos de recursos são globalmente únicos

Instâncias de recursos

As instâncias de recursos disponíveis pela API REST do Astra Control têm as seguintes características:

- As instâncias de recurso são criadas com base em um único tipo de recurso
- O tipo de recurso é indicado utilizando o valor tipo de material
- As instâncias são compostas por dados com estado que são mantidos pelo serviço Astra
- Cada instância é acessível através de uma URL única e de longa duração
- Nos casos em que uma instância de recurso pode ter mais de uma representação, diferentes tipos de Mídia podem ser usados para solicitar a representação desejada

Coleções de recursos

As coleções de recursos disponíveis pela API REST do Astra Control têm as seguintes características:

- O conjunto de instâncias de recursos de um único tipo de recurso é conhecido como uma coleção
- Coleções de recursos têm uma URL única e de longa duração

Identificadores de instância

Cada instância de recurso recebe um identificador quando é criada. Este identificador é um valor UUIDv4 de 128 bits. Os valores UUIDv4 atribuídos são globalmente únicos e imutáveis. Depois de emitir uma chamada de API que cria uma nova instância, um URL com o ID associado é retornado ao chamador em um `Location` cabeçalho da resposta HTTP. Você pode extrair o identificador e usá-lo em chamadas subsequentes quando se refere à instância de recurso.

O identificador de recurso é a chave primária usada para coleções.

Estrutura comum para recursos do Astra

Todos os recursos do Astra Control são definidos usando uma estrutura comum.

Dados comuns

Cada recurso Astra contém os valores-chave mostrados na tabela a seguir.

Chave	Descrição
tipo	Um tipo de recurso globalmente único que é conhecido como **tipo de recurso**.
versão	Um identificador de versão que é conhecido como a versão **resource**.
id	Um identificador global único que é conhecido como **resource identifier**.
metadados	Um objeto JSON contendo várias informações, incluindo rótulos de usuário e sistema.

Objeto de metadados

O objeto JSON de metadados incluído com cada recurso Astra contém os valores-chave mostrados na tabela a seguir.

Chave	Descrição
etiquetas	Matriz JSON de rótulos especificados pelo cliente associados ao recurso.
CriaçãoTimestamp	String JSON contendo um carimbo de data/hora indicando quando o recurso foi criado.
Alteração do Timestamp	String JSON contendo um carimbo de data/hora formatado ISO-8601 indicando quando o recurso foi alterado pela última vez.
CreatedBy	String JSON contendo o identificador UUIDv4 do ID do usuário que criou o recurso. Se o recurso foi criado por um componente interno do sistema e não houver UUID associado à entidade criadora, o UUID **null** é usado.

Estado do recurso

Recursos selecionados um `state` valor que é usado para orquestrar transições de ciclo de vida e controlar o acesso.

Detalhes HTTP

A API REST do Astra Control usa HTTP e parâmetros relacionados para agir nas instâncias e coleções de recursos. Detalhes da implementação HTTP são apresentados abaixo.

Transações de API e o modelo CRUD

A API REST do Astra Control implementa um modelo transacional com operações bem definidas e transições de estado.

Transação de API de solicitação e resposta

Cada chamada de API REST é executada como uma solicitação HTTP para o serviço Astra. Cada solicitação gera uma resposta associada de volta ao cliente. Esse par de solicitação-resposta pode ser considerado uma transação de API.

Suporte para o modelo operacional CRUD

Cada uma das instâncias e coleções de recursos disponíveis por meio da API REST do Astra Control é acessada com base no modelo **CRUD**. Existem quatro operações, cada uma delas mapeia para um único método HTTP. As operações incluem:

- Criar
- Leia
- Atualização
- Eliminar

Para alguns recursos do Astra, apenas um subconjunto dessas operações é suportado. Você deve revisar o "Referência da API" para obter mais informações sobre uma chamada de API específica.

Métodos HTTP

Os métodos HTTP ou verbos suportados pela API são apresentados na tabela abaixo.

Método	CRUD	Descrição
OBTER	Leia	Recupera propriedades de objeto para uma instância ou coleção de recursos. Isso é considerado uma operação **list** quando usado com uma coleção.
POST	Criar	Cria uma nova instância de recurso com base nos parâmetros de entrada. O URL de longo prazo é retornado em um `Location` cabeçalho de resposta.
COLOQUE	Atualização	Atualiza uma instância de recurso inteira com o corpo de solicitação JSON fornecido. Os valores-chave que não são modificáveis pelo usuário são preservados.
ELIMINAR	Eliminar	Exclui uma instância de recurso existente.

Cabeçalhos de solicitação e resposta

A tabela a seguir resume os cabeçalhos HTTP usados com a API REST do Astra Control.

"RFC 7232"Consulte e "RFC 7233" para obter mais informações.

Colhedor	Tipo	Notas de utilização
Aceitar	Pedido	Se o valor for "/" ou não for fornecido, `application/json` será retornado no cabeçalho de resposta Content-Type. Se o valor estiver definido para o tipo de Mídia de recurso Astra, o mesmo tipo de Mídia será retornado no cabeçalho Content-Type.
Autorização	Pedido	Token de portador com a chave API para o usuário.
Tipo de conteúdo	Resposta	Devolvido com base no `Accept` cabeçalho da solicitação.
ETAG	Resposta	Incluído com um sucesso como definido com RFC 7232. O valor é uma representação hexadecimal do valor MD5 para todo o recurso JSON.
If-Match	Pedido	Um cabeçalho de solicitação de pré-condição implementado conforme descrito na seção 3,1 RFC 7232 e suporte para solicitações **PUT**.
Se-modificado-desde	Pedido	Um cabeçalho de solicitação de pré-condição implementado conforme descrito na seção 3,4 RFC 7232 e suporte para solicitações **PUT**.
If-Unmodified-since	Pedido	Um cabeçalho de solicitação de pré-condição implementado conforme descrito na seção 3,4 RFC 7232 e suporte para solicitações **PUT**.
Localização	Resposta	Contém a URL completa do recurso recém-criado.

Parâmetros de consulta

Os seguintes parâmetros de consulta estão disponíveis para uso com coleções de recursos. Consulte "Trabalhando com coleções" para obter mais informações.

Parâmetro de consulta	Descrição
incluir	Contém os campos que devem ser retornados ao ler uma coleção.
filtro	Indica os campos que devem corresponder para que um recurso seja retornado ao ler uma coleção.
Ordenar	Determina a ordem de classificação dos recursos retornados ao ler uma coleção.
limite	Limita o número máximo de recursos retornados ao ler uma coleção.
ignorar	Define o número de recursos para passar e pular ao ler uma coleção.
contar	Indica se o número total de recursos deve ser retornado no objeto metadados.

Códigos de status HTTP

Os códigos de status HTTP usados pela API REST do Astra Control são descritos abaixo.

A API REST do Astra Control também usa o padrão **Detalhes do problema para APIs HTTP**. Consulte "Diagnóstico e suporte" para obter mais informações.

Código	Significado	Descrição
200	OK	Indica sucesso para chamadas que não criam uma nova instância de recurso.
201	Criado	Um objeto é criado com sucesso e o cabeçalho de resposta de localização inclui o identificador exclusivo para o objeto.
204	Nenhum conteúdo	A solicitação foi bem-sucedida, embora nenhum conteúdo tenha sido retornado.
400	Pedido incorreto	A entrada de solicitação não é reconhecida ou é inadequada.
401	Não autorizado	O usuário não está autorizado e deve autorizar.
403	Proibido	O acesso é negado devido a um erro de autorização.
404	Não encontrado	O recurso referido na solicitação não existe.
409	Conflito	Uma tentativa de criar um objeto falhou porque o objeto já existe.
500	Erro interno	Ocorreu um erro interno geral no servidor.
503	Serviço indisponível	O serviço não está pronto para lidar com a solicitação por algum motivo.

Formato de URL

A estrutura geral da URL usada para acessar uma instância ou coleção de recursos através da API REST é composta por vários valores. Esta estrutura reflete o modelo de objeto subjacente e o design do sistema.

Conta como raiz

A raiz do caminho do recurso para cada endpoint REST é a conta Astra. E assim todos os caminhos no URL começam com `/account/{account_id}` onde `account_id` é o valor UUIDv4 exclusivo para a conta. Estrutura interna isso reflete um design em que todo o acesso a recursos é baseado em uma conta específica.

Categoria de recurso de endpoint

Os pontos de extremidade de recursos do Astra se enquadram em três categorias diferentes:

- (`/core`Núcleo)
- Aplicativo gerenciado (`/k8s`)
- Topologia (`/topology`)

Consulte "Recursos" para obter mais informações.

Versão da categoria

Cada uma das três categorias de recursos tem uma versão global que controla a versão dos recursos acessados. Por convenção e definição, mover para uma nova versão principal de uma categoria de recurso (como, de `/v1` para `/v2`) introduzirá alterações de quebra na API.

Instância ou coleção de recursos

Uma combinação de tipos de recursos e identificadores pode ser usada no caminho, com base no acesso a uma instância ou coleção de recursos.

Exemplo

- Caminho do recurso

  Com base na estrutura apresentada acima, um caminho típico para um endpoint é: `/accounts/{account_id}/core/v1/users`.

- URL completo

  O URL completo para o endpoint correspondente é: `https://astra.netapp.io/accounts/{account_id}/core/v1/users`.

Recursos e endpoints

Você pode acessar os recursos fornecidos pela API REST do Astra Control para automatizar uma implantação Astra. Cada recurso está disponível por meio de um ou mais endpoints. Uma introdução aos RECURSOS REST que você pode usar como parte de uma implantação de automação é fornecida abaixo.

O formato do caminho e URL completo usados para acessar os recursos do Astra Control é baseado em vários valores. Consulte "Formato de URL" para obter mais informações. Consulte também "Referência da API"para obter mais detalhes sobre como usar os recursos e pontos de extremidade do Astra.

Resumo dos RECURSOS REST do Astra Control

Os principais pontos de extremidade de recurso fornecidos na API REST do Astra Control são organizados em três categorias. Cada recurso pode ser acessado com o conjunto completo de operações CRUD (criar, ler, atualizar, excluir), exceto onde indicado.

A coluna **Release** indica a versão Astra quando o recurso foi introduzido pela primeira vez. Este campo está em negrito para recursos recém-adicionados com a versão atual.

Recursos básicos

Os principais pontos de extremidade dos recursos fornecem os serviços básicos necessários para estabelecer e manter o ambiente de tempo de execução do Astra.

Recurso	Solte	Descrição
Conta	21,12	Os recursos da conta permitem gerenciar locatários isolados no ambiente de implantação com alocação a vários clientes Astra Control.
ASUP	21,08	Os recursos do ASUP representam os pacotes do AutoSupport encaminhados para o suporte do NetApp.
Certificado	**22,08**	Os recursos de certificado representam os certificados instalados usados para autenticação forte para conexões de saída.
Credencial	21,04	Os recursos de credenciais contêm informações relacionadas à segurança que podem ser usadas com usuários, clusters, buckets e back-ends de storage do Astra.
Direitos	21,08	Os recursos de direito representam os recursos e capacidades disponíveis para uma conta com base nas licenças e assinaturas ativas.
Evento	21,04	Os recursos do evento representam todos os eventos que ocorrem no sistema, incluindo o subconjunto classificado como notificações.
Gancho de execução	21,12	Os recursos do gancho de execução representam scripts personalizados que você pode executar antes ou depois que um snapshot de um aplicativo gerenciado é executado.
Recurso	21,08	Os recursos do recurso representam os recursos selecionados do Astra que você pode consultar para determinar se eles estão ativados ou desativados no sistema. O acesso é limitado a somente leitura.

Recurso	Solte	Descrição
Grupo	**22,08**	Os recursos do grupo representam os grupos Astra e os recursos associados. Apenas os grupos LDAP são suportados na versão atual.
Fonte do gancho	21,12	Os recursos de origem do gancho representam o código-fonte real usado com um gancho de execução. Separar o código-fonte do controle de execução tem vários benefícios, como permitir que os scripts sejam compartilhados.
Licença	21,08	Os recursos de licença representam as licenças disponíveis para uma conta Astra.
Notificação	21,04	Os recursos de notificação representam eventos Astra que têm um destino de notificação. O acesso é fornecido por usuário.
Pacote	22,04	Os recursos do pacote fornecem o Registro e o acesso às definições do pacote. Pacotes de software consistem em vários componentes, incluindo arquivos, imagens e outros artefatos.
Vinculação de função	21,04	Os recursos de vinculação de função representam as relações entre pares específicos de usuários e contas. Além da ligação entre os dois, um conjunto de permissões é especificado para cada um através de uma função específica.
Definição	21,08	A configuração de recursos representa um conjunto de pares de valor-chave que descrevem um recurso para uma conta Astra específica.
Subscrição	21,08	Os recursos de subscrição representam as subscrições ativas de uma conta Astra.
Token	21,04	Os recursos do token representam os tokens disponíveis para acessar programaticamente a API REST DO Astra Control.
Notificação não lida	21,04	Os recursos de notificação não lidos representam notificações atribuídas a um usuário específico, mas ainda não lidas.
Atualização	22,04	Os recursos de atualização fornecem acesso a componentes de software e a capacidade de iniciar atualizações.
Utilizador	21,04	Os recursos de usuário representam usuários do Astra capazes de acessar o sistema com base em sua função definida.

Recursos de aplicativos gerenciados

Os pontos de extremidade de recursos de aplicação gerenciados fornecem acesso às aplicações Kubernetes gerenciadas.

Recurso	Solte	Descrição
Ativo da aplicação	21,04	Os recursos de ativos da aplicação representam coleções internas de informações de estado necessárias para gerenciar as aplicações Astra.
Backup de aplicativos	21,04	Os recursos de backup de aplicativos representam backups dos aplicativos gerenciados.
Snapshot da aplicação	21,04	Os recursos de snapshot do aplicativo representam snapshots dos aplicativos gerenciados.

Recurso	Solte	Descrição
Substituição do gancho de execução	21,12	Os recursos de substituição do gancho de execução permitem desativar os ganchos de execução predefinidos do NetApp pré-carregados para aplicações específicas, conforme necessário.
Programação	21,04	Os recursos do cronograma representam operações de proteção de dados agendadas para os aplicativos gerenciados como parte de uma política de proteção de dados.

Recursos de topologia

Os pontos de extremidade dos recursos de topologia fornecem acesso aos aplicativos não gerenciados e aos recursos de storage.

Recurso	Solte	Descrição
Aplicação	21,04	Os recursos da aplicação representam todas as aplicações Kubernetes, incluindo as não gerenciadas pelo Astra.
AppMirror	**22,08**	Os recursos do AppMirror representam os recursos do AppMirror a fornecer para o gerenciamento de relacionamentos de espelhamento de aplicativos.
Balde	21,08	Os recursos de bucket representam os buckets em nuvem do S3 usados para armazenar backups das aplicações gerenciadas pelo Astra.
Nuvem	21,08	Os recursos de nuvem representam nuvens às quais os clientes Astra podem se conectar para gerenciar clusters e aplicações.
Cluster	21,08	Os recursos do cluster representam os clusters do Kubernetes não gerenciados pelo Kubernetes.
Nó de cluster	21,12	Os recursos do nó do cluster fornecem resolução adicional, permitindo que você acesse os nós individuais em um cluster do Kubernetes.
Cluster gerenciado	21,08	Os recursos do cluster gerenciado representam os clusters do Kubernetes atualmente gerenciados pelo Kubernetes.
Back-end de storage gerenciado	21,12	Os recursos de back-end de storage gerenciado permitem acessar representações abstratas dos provedores de storage de back-end. Esses back-ends de storage podem ser usados pelos clusters e aplicativos gerenciados.
Namespace	21,12	Os recursos de namespace fornecem acesso aos namespaces usados em um cluster do Kubernetes.
Back-end de storage	21,08	Os recursos de back-end de storage representam fornecedores de serviços de storage que podem ser usados pelos clusters e aplicações gerenciados do Astra.
Classe de armazenamento	21,08	Os recursos da classe de armazenamento representam diferentes classes ou tipos de armazenamento descobertos e disponíveis para um cluster gerenciado específico.
Volume	21,04	Os recursos de volume representam os volumes de storage do Kubernetes associados às aplicações gerenciadas.

Recursos e endpoints adicionais

Há vários recursos e pontos de extremidade adicionais que você pode usar para dar suporte a uma implantação do Astra.

Esses recursos e pontos de extremidade não estão incluídos atualmente na documentação de referência da API REST do Astra Control.

OpenAPI

Os endpoints OpenAPI fornecem acesso ao documento JSON OpenAPI atual e a outros recursos relacionados.

OpenMetrics

Os endpoints OpenMetrics fornecem acesso às métricas da conta por meio do recurso OpenMetrics. O suporte está disponível com o modelo de implantação do Astra Control Center.

Considerações de uso adicionais

Segurança RBAC

A API REST do Astra dá suporte ao controle de acesso baseado em funções (RBAC) para conceder e restringir o acesso às funções do sistema.

Funções do Astra

Cada usuário do Astra é atribuído a uma única função que determina as ações que podem ser executadas. As funções são organizadas em uma hierarquia conforme descrito na tabela abaixo.

Função	Descrição
Proprietário	Tem todas as permissões da função Admin e também pode excluir contas Astra.
Administrador	Tem todas as permissões da função Membro e também pode convidar os usuários para participar de uma conta.
Membro	Pode gerenciar totalmente a aplicação Astra e os recursos de computação.
Visualizador	Restrito à visualização apenas de recursos.

RBAC aprimorado com granularidade de namespace

Esse recurso foi introduzido com a versão 22,04 da API REST do Astra.

Quando uma vinculação de função é estabelecida para um usuário específico, uma restrição pode ser aplicada para limitar os namespaces aos quais o usuário tem acesso. Existem várias maneiras de definir essa restrição como descrito na tabela abaixo. Consulte o parâmetro `roleContraints` na API de vinculação de função para obter mais informações.

Namespaces	Descrição
Tudo	O usuário pode acessar todos os namespaces através do parâmetro curinga "*". Este é o valor padrão para manter a compatibilidade com versões anteriores.
Nenhum	A lista de restrições é especificada embora esteja vazia. Isso indica que o usuário não pode acessar nenhum namespace.
Lista de namespace	O UUID de um namespace está incluído, o que restringe o usuário ao namespace único. Uma lista separada por vírgulas também pode ser usada para permitir o acesso a vários namespaces.
Etiqueta	Um rótulo é especificado e o acesso é permitido a todos os namespaces correspondentes.

Trabalhe com coleções

A API REST do Astra Control fornece várias maneiras diferentes de acessar coleções de recursos por meio dos parâmetros de consulta definidos.

Selecionar valores

Você pode especificar quais pares de chave-valor devem ser retornados para cada instância de recurso usando o `include` parâmetro. Todas as instâncias são retornadas no corpo de resposta.

Filtragem

A filtragem de recursos de coleta permite que um usuário da API especifique condições que determinam se um recurso é retornado no corpo da resposta. O `filter` parâmetro é usado para indicar a condição de filtragem.

Ordenação

A classificação de recursos de coleta permite que um usuário da API especifique a ordem em que os recursos são retornados no corpo de resposta. O `orderBy` parâmetro é usado para indicar a condição de filtragem.

Paginação

Você pode impor a paginação restringindo o número de instâncias de recursos retornadas em uma solicitação usando o `limit` parâmetro.

Contar

Se você incluir o parâmetro booleano `count` definido como `true`, o número de recursos na matriz retornada para uma determinada resposta será fornecido na seção metadados.

Diagnóstico e suporte

Há vários recursos de suporte disponíveis com a API REST do Astra Control que podem ser usados para diagnósticos e depuração.

Recursos de API

Há vários recursos do Astra expostos por meio de recursos de API que fornecem informações de diagnóstico e suporte.

Tipo	Descrição
Evento	Atividades do sistema registradas como parte do processamento do Astra.
Notificação	Um subconjunto dos Eventos que são considerados importantes o suficiente para ser apresentado ao usuário.
Notificação não lida	As notificações que ainda não foram lidas ou recuperadas pelo usuário.

Revogar um token de API

Você pode revogar um token de API na interface da Web Astra quando não for mais necessário.

Antes de começar

Você precisa de uma conta Astra. Você também deve identificar os tokens que deseja revogar.

Sobre esta tarefa

Depois que um token é revogado, ele é imediatamente e permanentemente inutilizável.

Passos

1. Faça login no Astra usando suas credenciais de conta.

   Acesse o seguinte site do Astra Control Service: "https://astra.netapp.io"

2. Clique no ícone de figura no canto superior direito da página e selecione **Acesso à API**.
3. Selecione o token ou tokens que você deseja revogar.
4. Na caixa suspensa **ações**, clique em **revogar tokens**.

Workflows de infraestrutura

Antes de começar

Use esses workflows para criar e manter a infraestrutura usada com uma implantação do Astra Control Center. Em muitos casos, os workflows também podem ser usados com o Astra Control Service.

Esses fluxos de trabalho podem ser expandidos e aprimorados pelo NetApp a qualquer momento, portanto, você deve revisá-los periodicamente.

Preparação geral

Antes de usar qualquer um dos workflows do Astra, revise "Prepare-se para usar os fluxos de trabalho"o .

Categorias de fluxo de trabalho

Os fluxos de trabalho de infraestrutura são organizados em diferentes categorias para facilitar a localização do que você deseja.

Categoria	Descrição
Identidade e acesso	Esses fluxos de trabalho permitem gerenciar identidade e como o Astra é acessado. Os recursos incluem usuários, credenciais e tokens.
Configuração LDAP	Opcionalmente, você pode configurar o Astra Control Center para usar o LDAP para autenticar usuários selecionados.
Baldes	Você pode usar esses fluxos de trabalho para criar e gerenciar os buckets do S3 usados para armazenar backups.
Armazenamento	Esses fluxos de trabalho permitem adicionar e manter backends e volumes de armazenamento.
Clusters	É possível adicionar clusters gerenciados do Kubernetes que permitem proteger e dar suporte às aplicações que eles contêm.

Identidade e acesso

Listar usuários

Você pode listar os usuários que estão definidos para uma conta Astra específica.

1. Liste os usuários

Execute a seguinte chamada de API REST.

Método HTTP	Caminho
OBTER	/conta/_id/core/v1/users

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
incluir	Consulta	Não	Opcionalmente, selecione os valores que você deseja retornar na resposta.

Curl exemplo: Retorna todos os dados para todos os usuários

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

Curl exemplo: Retorna o primeiro nome, sobrenome e id para todos os usuários

```
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>'
```

Exemplo de saída JSON


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

Configuração LDAP

Prepare-se para a configuração LDAP

Opcionalmente, é possível integrar o Astra Control Center a um servidor LDAP (Lightweight Directory Access Protocol) para executar a autenticação para usuários selecionados do Astra. O LDAP é um protocolo padrão do setor para acessar informações de diretórios distribuídos e uma escolha popular para autenticação empresarial.

Informações relacionadas

- "Mapa rodoviário da especificação técnica LDAP"
- "LDAP versão 3"

Visão geral do processo de implementação

Em um alto nível, há várias etapas que você precisa executar para configurar um servidor LDAP para fornecer autenticação para usuários Astra.

> Enquanto as etapas apresentadas abaixo estão em uma sequência, em alguns casos você pode executá-las em uma ordem diferente. Por exemplo, você pode definir os usuários e grupos Astra antes de configurar o servidor LDAP.

1. Revise "Requisitos e limitações" para entender as opções, os requisitos e as limitações.
2. Selecione um servidor LDAP e as opções de configuração desejadas (incluindo segurança).
3. Execute o fluxo de trabalho "Configure o Astra para usar um servidor LDAP" para integrar o Astra ao servidor LDAP.
4. Reveja os utilizadores e grupos no servidor LDAP para se certificar de que estão definidos corretamente.
5. Execute o fluxo de trabalho apropriado em "Adicionar entradas LDAP ao Astra" para identificar os usuários a serem autenticados usando LDAP.

Requisitos e limitações

Antes de configurar o Astra para usar LDAP para autenticação, você deve consultar os fundamentos de configuração do Astra apresentados abaixo, incluindo limitações e opções de configuração.

Compatível apenas com Astra Control Center

A plataforma Astra Control oferece dois modelos de implantação. A autenticação LDAP só é compatível com implantações do Astra Control Center.

Somente configuração da API REST

A versão atual do Astra Control Center só dá suporte à configuração de autenticação LDAP usando a API REST Astra Control. Um aspeto importante dessa limitação é que os usuários LDAP não são exibidos na guia usuários da interface Web Astra. Eles estão disponíveis através da API REST no endpoint `../core/v1/users`.

Servidor LDAP necessário

É necessário ter um servidor LDAP para aceitar e processar as solicitações de autenticação Astra. O ative Directory da Microsoft é compatível com a versão atual do Astra Control Center.

Ligação segura ao servidor LDAP

Ao configurar o servidor LDAP no Astra, você pode definir opcionalmente uma conexão segura. Neste caso, é necessário um certificado para o protocolo LDAPS.

Configurar usuários ou grupos

Você precisa selecionar os usuários a serem autenticados usando LDAP. Você pode fazer isso identificando os usuários individuais ou um grupo de usuários. As contas devem ser definidas no servidor LDAP. Eles também precisam ser identificados no Astra (tipo LDAP), o que permite que as solicitações de autenticação sejam encaminhadas para LDAP.

Restrição de função ao vincular um usuário ou grupo

Com a versão atual do Astra Control Center, o único valor suportado para `roleConstraint` é "*". Isso indica que o usuário não está restrito a um conjunto limitado de namespaces e pode acessar todos eles. Consulte "Adicionar entradas LDAP ao Astra" para obter mais informações.

Credenciais LDAP

As credenciais usadas pelo LDAP incluem o nome de usuário (endereço de e-mail) e a senha associada.

Endereços de e-mail exclusivos

Todos os endereços de e-mail atuando como nomes de usuário em uma implantação do Astra Control Center devem ser exclusivos. Não é possível adicionar um usuário LDAP com um endereço de e-mail que já esteja definido para Astra. Se houver um e-mail duplicado, você precisará primeiro excluí-lo do Astra. Consulte "Remover usuários" o site de documentação do Astra Control Center para obter mais informações.

Opcionalmente, defina primeiro os usuários e grupos LDAP

Você pode adicionar os usuários e grupos LDAP ao Astra Control Center mesmo que eles ainda não existam no LDAP ou se o servidor LDAP não estiver configurado. Isto permite pré-configurar os utilizadores e grupos antes de configurar o servidor LDAP.

Um usuário definido em vários grupos LDAP

Se um usuário LDAP pertencer a vários grupos LDAP e os grupos tiverem sido atribuídos diferentes funções no Astra, a função efetiva do usuário ao autenticar será a mais privilegiada. Por exemplo, se um usuário for atribuído a `viewer` função com group1, mas tiver a `member` função em group2, a função do usuário será `member`. Isso é baseado na hierarquia usada pelo Astra (mais alto a mais baixo):

- Proprietário
- Administrador
- Membro
- Visualizador

Sincronização periódica de contas

O Astra sincroniza os usuários e grupos de TI com o servidor LDAP aproximadamente a cada 60 segundo. Portanto, se um usuário ou grupo for adicionado ou removido do LDAP, pode levar até um minuto antes de estar disponível no Astra.

Desativar e repor a configuração LDAP

Antes de tentar repor a configuração LDAP, tem de desativar primeiro a autenticação LDAP. Além disso, para alterar o servidor LDAP (`connectionHost`), é necessário executar ambas as operações. Consulte "Desativar e repor LDAP" para obter mais informações.

Parâmetros da API REST

Os fluxos de trabalho de configuração LDAP fazem chamadas de API REST para realizar as tarefas específicas. Cada chamada de API pode incluir parâmetros de entrada como mostrado nas amostras fornecidas. Consulte "Referência da API" para obter informações sobre como localizar a documentação de referência.

Configure o Astra para usar um servidor LDAP

Você precisa selecionar um servidor LDAP e configurar o Astra para usar o servidor como um provedor de autenticação. A tarefa de configuração consiste nos passos descritos abaixo. Cada etapa inclui uma única chamada de API REST.

1. Adicione um certificado de CA

Execute a seguinte chamada de API REST para adicionar um certificado de CA ao Astra.

Esta etapa é opcional e somente necessária se você quiser que o Astra e o LDAP se comuniquem em um canal seguro usando o LDAPS.

Método HTTP	Caminho
POST	/account//core/v1/certificates

Exemplo de entrada JSON

```
{
  "type": "application/astra-certificate",
  "version": "1.0",
  "certUse": "rootCA",
  "cert": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMyVEN",
  "isSelfSigned": "true"
}
```

Observe o seguinte sobre os parâmetros de entrada:

- `cert` É uma string JSON contendo um certificado formatado PKCS-11 codificado base64 (codificado PEM).
- `isSelfSigned` deve ser definido como `true` se o certificado for auto-assinado. A predefinição é `false`.

Curl exemplo

```
curl --location -i --request POST --data @JSONinput
'https://astra.example.com/accounts/<ACCOUNT_ID>/core/v1/certificates'
--header 'Content-Type: application/astra-certificate+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Exemplo de resposta JSON


```
{
  "type": "application/astra-certificate",
  "version": "1.0",
  "id": "a5212e7e-402b-4cff-bba0-63f3c6505199",
  "certUse": "rootCA",
  "cert": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMyVEN",
  "cn": "adldap.example.com",
  "expiryTimestamp": "2023-07-08T20:22:07Z",
  "isSelfSigned": "true",
  "trustState": "trusted",
  "trustStateTransitions": [
  {
    "from": "untrusted",
    "to": [
      "trusted",
      "expired"
    ]
  },
  {
    "from": "trusted",
    "to": [
      "untrusted",
      "expired"
    ]
  },
  {
    "from": "expired",
    "to": [
     "untrusted",
     "trusted"
    ]
  }
  ],
  "trustStateDesired": "trusted",
  "trustStateDetails": [],
  "metadata": {
    "creationTimestamp": "2022-07-21T04:16:06Z",
    "modificationTimestamp": "2022-07-21T04:16:06Z",
    "createdBy": "8a02d2b8-a69d-4064-827f-36851b3e1e6e",
    "modifiedBy": "8a02d2b8-a69d-4064-827f-36851b3e1e6e",
    "labels": []
    }
}
```

2. Adicione as credenciais de vinculação

Execute a seguinte chamada de API REST para adicionar as credenciais de vinculação.

Método HTTP	Caminho
POST	/account/_id/core/v1/credentials

Exemplo de entrada JSON

```
{
  "name": "ldapBindCredential",
  "type": "application/astra-credential",
  "version": "1.1",
  "keyStore": {
    "bindDn": "dWlkPWFkbWluLG91PXN5c3RlbQ==",
    "password": "cGFzc3dvcmQ="
  }
}
```

Observe o seguinte sobre os parâmetros de entrada:

- `bindDn` E `password` são as credenciais de vinculação codificadas base64 do usuário de administrador LDAP que é capaz de conetar e pesquisar o diretório LDAP. `bindDn` É o endereço de e-mail do usuário LDAP.

Curl exemplo

```
curl --location -i --request POST --data @JSONinput
'https://astra.example.com/accounts/<ACCOUNT_ID>/core/v1/credentials'
--header 'Content-Type: application/astra-credential+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Exemplo de resposta JSON

```
{
    "type": "application/astra-credential",
    "version": "1.1",
    "id": "3bd9c8a7-f5a4-4c44-b778-90a85fc7d154",
    "name": "ldapBindCredential",
    "metadata": {
        "creationTimestamp": "2022-07-21T06:53:11Z",
        "modificationTimestamp": "2022-07-21T06:53:11Z",
        "createdBy": "527329f2-662c-41c0-ada9-2f428f14c137"
    }
}
```


Observe os seguintes parâmetros de resposta:

- `id`O da credencial é utilizado em etapas subsequentes do fluxo de trabalho.

3. Recupere o UUID da configuração LDAP

Execute a seguinte chamada de API REST para recuperar o UUID da `astra.account.ldap` configuração incluída no Astra Control Center.

> O exemplo curl abaixo usa um parâmetro de consulta para filtrar a coleção de configurações. Em vez disso, você pode remover o filtro para obter todas as configurações e, em seguida, procurar `astra.account.ldap` .

Método HTTP	Caminho
OBTER	/account/_id/core/v1/settings

Curl exemplo

```
curl --location -i --request GET
'https://astra.example.com/accounts/<ACCOUNT_ID>/core/v1/settings?filter=n
ame%20eq%20'astra.account.ldap'&include=name,id' --header 'Accept: */*'
--header 'Authorization: Bearer <API_TOKEN>'
```

Exemplo de resposta JSON

```
{
  "items": [
    ["astra.account.ldap",
    "12072b56-e939-45ec-974d-2dd83b7815df"
    ]
  ],
  "metadata": {}
}
```

4. Atualize a definição LDAP

Execute a seguinte chamada de API REST para atualizar a configuração LDAP e concluir a configuração. Use o `id` valor da chamada de API anterior para o `<SETTING_ID>` valor no caminho de URL abaixo.

Você pode emitir uma SOLICITAÇÃO GET para a configuração específica primeiro para ver o configSchema. Isso fornecerá mais informações sobre os campos obrigatórios na configuração.

Método HTTP	Caminho
COLOQUE	/account/_id/core/v1/settings/ [definições_id]

Exemplo de entrada JSON

```
{
  "type": "application/astra-setting",
  "version": "1.0",
  "desiredConfig": {
    "connectionHost": "myldap.example.com",
    "credentialId": "3bd9c8a7-f5a4-4c44-b778-90a85fc7d154",
    "groupBaseDN": "OU=groups,OU=astra,DC=example,DC=com",
    "isEnabled": "true",
    "port": 686,
    "secureMode": "LDAPS",
    "userBaseDN": "OU=users,OU=astra,DC=example,dc=com",
    "userSearchFilter": "((objectClass=User))",
    "vendor": "Active Directory"
    }
}
```

Observe o seguinte sobre os parâmetros de entrada:

- `isEnabled` deve ser definido como `true` ou pode ocorrer um erro.
- `credentialId` é o id da credencial de ligação criada anteriormente.
- `secureMode` deve ser definido como `LDAP` ou `LDAPS` com base na sua configuração na etapa anterior.

- Apenas o 'ative Directory' é suportado como fornecedor.

Curl exemplo

```
curl --location -i --request PUT --data @JSONinput
'https://astra.example.com/accounts/<ACCOUNT_ID>/core/v1/settings/<SETTING
_ID>' --header 'Content-Type: application/astra-setting+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Se a chamada for bem-sucedida, a resposta HTTP 204 será retornada.

5. Recupere a definição LDAP

Opcionalmente, você pode executar a seguinte chamada de API REST para recuperar as configurações LDAP e confirmar a atualização.

Método HTTP	Caminho
OBTER	/account/_id/core/v1/settings/ [definições_id]

Curl exemplo

```
curl --location -i --request GET
'https://astra.example.com/accounts/<ACCOUNT_ID>/core/v1/settings/<SETTING
_ID>' --header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Exemplo de resposta JSON


```
{
  "items": [
  {
    "type": "application/astra-setting",
    "version": "1.0",
    "metadata": {
      "creationTimestamp": "2022-06-17T21:16:31Z",
      "modificationTimestamp": "2022-07-21T07:12:20Z",
      "labels": [],
      "createdBy": "system",
      "modifiedBy": "00000000-0000-0000-0000-000000000000"
    },
    "id": "12072b56-e939-45ec-974d-2dd83b7815df",
    "name": "astra.account.ldap",
    "desiredConfig": {
      "connectionHost": "10.193.61.88",
      "credentialId": "3bd9c8a7-f5a4-4c44-b778-90a85fc7d154",
      "groupBaseDN": "ou=groups,ou=astra,dc=example,dc=com",
```

```
      "isEnabled": "true",
      "port": 686,
      "secureMode": "LDAPS",
      "userBaseDN": "ou=users,ou=astra,dc=example,dc=com",
      "userSearchFilter": "((objectClass=User))",
      "vendor": "Active Directory"
    },
    "currentConfig": {
      "connectionHost": "10.193.160.209",
      "credentialId": "3bd9c8a7-f5a4-4c44-b778-90a85fc7d154",
      "groupBaseDN": "ou=groups,ou=astra,dc=example,dc=com",
      "isEnabled": "true",
      "port": 686,
      "secureMode": "LDAPS",
      "userBaseDN": "ou=users,ou=astra,dc=example,dc=com",
      "userSearchFilter": "((objectClass=User))",
      "vendor": "Active Directory"
    },
    "configSchema": {
      "$schema": "http://json-schema.org/draft-07/schema#",
      "title": "astra.account.ldap",
      "type": "object",
      "properties": {
        "connectionHost": {
          "type": "string",
          "description": "The hostname or IP address of your LDAP server."
        },
        "credentialId": {
          "type": "string",
          "description": "The credential ID for LDAP account."
        },
        "groupBaseDN": {
          "type": "string",
          "description": "The base DN of the tree used to start the group
search. The system searches the subtree from the specified location."
        },
        "groupSearchCustomFilter": {
          "type": "string",
          "description": "Type of search that controls the default group
search filter used."
        },
        "isEnabled": {
          "type": "string",
          "description": "This property determines if this setting is
enabled or not."
        },
```

```
        "port": {
          "type": "integer",
          "description": "The port on which the LDAP server is running."
        },
        "secureMode": {
          "type": "string",
          "description": "The secure mode LDAPS or LDAP."
        },
        "userBaseDN": {
          "type": "string",
          "description": "The base DN of the tree used to start the user
search. The system searches the subtree from the specified location."
        },
        "userSearchFilter": {
          "type": "string",
          "description": "The filter used to search for users according a
search criteria."
        },
        "vendor": {
          "type": "string",
          "description": "The LDAP provider you are using.",
          "enum": ["Active Directory"]
        }
      },
      "additionalProperties": false,
      "required": [
        "connectionHost",
        "secureMode",
        "credentialId",
        "userBaseDN",
        "userSearchFilter",
        "groupBaseDN",
        "vendor",
        "isEnabled"
      ]
      },
      "state": "valid",
    }
  ],
  "metadata": {}
}
```

Localize o `state` campo na resposta que terá um dos valores na tabela abaixo.

Estado	Descrição
pendente	O processo de configuração ainda está ativo e ainda não foi concluído.
válido	A configuração foi concluída com sucesso e `currentConfig` na resposta corresponde `desiredConfig`.
erro	O processo de configuração LDAP falhou.

Adicionar entradas LDAP ao Astra

Depois que o LDAP é configurado como um provedor de autenticação para o Astra Control Center, você pode selecionar os usuários LDAP que o Astra autenticará usando as credenciais LDAP. Cada usuário precisa ter uma função no Astra antes de poder acessar o Astra por meio da API REST Astra Control.

Há duas maneiras de configurar o Astra para atribuir funções. Escolha o que é apropriado para o seu ambiente.

- "Adicione e vincule um usuário individual"
- "Adicione e vincule um grupo"

As credenciais LDAP estão na forma de um nome de usuário como endereço de e-mail e a senha LDAP associada.

Adicione e vincule um usuário individual

Você pode atribuir uma função a cada usuário Astra que é usado após a autenticação LDAP. Isso é apropriado quando há um pequeno número de usuários e cada um pode ter caraterísticas administrativas diferentes.

1. Adicionar um utilizador

Execute a seguinte chamada de API REST para adicionar um usuário ao Astra e indicar que o LDAP é o provedor de autenticação.

Método HTTP	Caminho
POST	/conta/_id/core/v1/users

Exemplo de entrada JSON

```
{
  "type" :  "application/astra-user",
  "version" : "1.1",
  "authID" : "cn=JohnDoe,ou=users,ou=astra,dc=example,dc=com",
  "authProvider" : "ldap",
  "firstName" : "John",
  "lastName" : "Doe",
  "email" : "john.doe@example.com"
}
```

Observe o seguinte sobre os parâmetros de entrada:

- São necessários os seguintes parâmetros:
  - `authProvider`
  - `authID`
  - `email`
- `authID` É o nome distinto (DN) do usuário no LDAP
- `email` Deve ser exclusivo para todos os usuários definidos no Astra

Se o `email` valor não for exclusivo, ocorrerá um erro e um código de status HTTP 409 será retornado na resposta.

Curl exemplo

```
curl --location -i --request POST --data @JSONinput
'https://astra.example.com/accounts/<ACCOUNT_ID>/core/v1/users' --header
'Content-Type: application/astra-user+json' --header 'Accept: */*'
--header 'Authorization: Bearer <API_TOKEN>'
```

Exemplo de resposta JSON

```
{
  "metadata": {
    "creationTimestamp": "2022-07-21T17:44:18Z",
    "modificationTimestamp": "2022-07-21T17:44:18Z",
    "createdBy": "8a02d2b8-a69d-4064-827f-36851b3e1e6e",
    "labels": []
  },
  "type": "application/astra-user",
  "version": "1.2",
  "id": "a7b5e674-a1b1-48f6-9729-6a571426d49f",
  "authProvider": "ldap",
  "authID": "cn=JohnDoe,ou=users,ou=astra,dc=example,dc=com",
  "firstName": "John",
  "lastName": "Doe",
  "companyName": "",
  "email": "john.doe@example.com",
  "postalAddress": {
    "addressCountry": "",
    "addressLocality": "",
    "addressRegion": "",
    "streetAddress1": "",
    "streetAddress2": "",
    "postalCode": ""
  },
  "state": "active",
  "sendWelcomeEmail": "false",
  "isEnabled": "true",
  "isInviteAccepted": "true",
  "enableTimestamp": "2022-07-21T17:44:18Z",
  "lastActTimestamp": ""
}
```


2. Adicione uma vinculação de função para o usuário

Execute a seguinte chamada de API REST para vincular o usuário a uma função específica. Você precisa ter o UUID do usuário criado na etapa anterior.

Método HTTP	Caminho
POST	/Account/_id/core/v1/roleBindings

Exemplo de entrada JSON

```
{
  "type": "application/astra-roleBinding",
  "version": "1.1",
  "accountID": "{account_id}",
  "userID": "a7b5e674-a1b1-48f6-9729-6a571426d49f",
  "role": "member",
  "roleConstraints": ["*"]
}
```

Observe o seguinte sobre os parâmetros de entrada:

- O valor utilizado acima para `roleConstraint` é a única opção disponível para a versão atual do Astra. Ele indica que o usuário não está restrito a um conjunto limitado de namespaces e pode acessá-los todos.

Curl exemplo

```
curl --location -i --request POST --data @JSONinput
'https://astra.example.com/accounts/<ACCOUNT_ID>/core/v1/roleBindings'
--header 'Content-Type: application/astra-roleBinding+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Exemplo de resposta JSON

```
{
  "metadata": {
    "creationTimestamp": "2022-07-21T18:08:24Z",
    "modificationTimestamp": "2022-07-21T18:08:24Z",
    "createdBy": "8a02d2b8-a69d-4064-827f-36851b3e1e6e",
    "labels": []
  },
  "type": "application/astra-roleBinding",
  "principalType": "user",
  "version": "1.1",
  "id": "b02c7e4d-d483-40d1-aaff-e1f900312114",
  "userID": "a7b5e674-a1b1-48f6-9729-6a571426d49f",
  "groupID": "00000000-0000-0000-0000-000000000000",
  "accountID": "d0fdbfa7-be32-4a71-b59d-13d95b42329a",
  "role": "member",
  "roleConstraints": ["*"]
}
```

Observe o seguinte sobre os parâmetros de resposta:

- O valor `user` para o `principalType` campo indica que a vinculação de função foi adicionada para um

usuário (não para um grupo).

Adicione e vincule um grupo

Você pode atribuir uma função a um grupo Astra que é usado após a autenticação LDAP. Isso é apropriado quando há um grande número de usuários e cada um pode ter caraterísticas administrativas semelhantes.

1. Adicionar um grupo

Execute a seguinte chamada de API REST para adicionar um grupo ao Astra e indicar que o LDAP é o provedor de autenticação.

Método HTTP	Caminho
POST	/account//core/v1/groups

Exemplo de entrada JSON

```
{
  "type": "application/astra-group",
  "version": "1.0",
  "name": "Engineering",
  "authProvider": "ldap",
  "authID": "CN=Engineering,OU=groups,OU=astra,DC=example,DC=com"
}
```

Observe o seguinte sobre os parâmetros de entrada:

- São necessários os seguintes parâmetros:
  - `authProvider`
  - `authID`

Curl exemplo

```
curl --location -i --request POST --data @JSONinput
'https://astra.example.com/accounts/<ACCOUNT_ID>/core/v1/groups' --header
'Content-Type: application/astra-group+json' --header 'Accept: */*'
--header 'Authorization: Bearer <API_TOKEN>'
```

Exemplo de resposta JSON

```
{
  "type": "application/astra-group",
  "version": "1.0",
  "id": "8b5b54da-ae53-497a-963d-1fc89990525b",
  "name": "Engineering",
  "authProvider": "ldap",
  "authID": "CN=Engineering,OU=groups,OU=astra,DC=example,DC=com",
  "metadata": {
    "creationTimestamp": "2022-07-21T18:42:52Z",
    "modificationTimestamp": "2022-07-21T18:42:52Z",
    "createdBy": "8a02d2b8-a69d-4064-827f-36851b3e1e6e",
    "labels": []
  }
}
```

2. Adicione uma vinculação de função para o grupo

Execute a seguinte chamada de API REST para vincular o grupo a uma função específica. Você precisa ter o UUID do grupo criado na etapa anterior. Os usuários que são membros do grupo poderão fazer login no Astra após o LDAP executar a autenticação.

Método HTTP	Caminho
POST	/Account/_id/core/v1/roleBindings

Exemplo de entrada JSON

```
{
  "type": "application/astra-roleBinding",
  "version": "1.1",
  "accountID": "{account_id}",
  "groupID": "8b5b54da-ae53-497a-963d-1fc89990525b",
  "role": "viewer",
  "roleConstraints": ["*"]
}
```

Observe o seguinte sobre os parâmetros de entrada:

- O valor utilizado acima para `roleConstraint` é a única opção disponível para a versão atual do Astra. Ele indica que o usuário não está restrito a certos namespaces e pode acessá-los todos.

Curl exemplo

```
curl --location -i --request POST --data @JSONinput
'https://astra.example.com/accounts/<ACCOUNT_ID>/core/v1/roleBindings'
--header 'Content-Type: application/astra-roleBinding+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Exemplo de resposta JSON

```
{
  "metadata": {
    "creationTimestamp": "2022-07-21T18:59:43Z",
    "modificationTimestamp": "2022-07-21T18:59:43Z",
    "createdBy": "527329f2-662c-41c0-ada9-2f428f14c137",
    "labels": []
  },
  "type": "application/astra-roleBinding",
  "principalType": "group",
  "version": "1.1",
  "id": "2f91b06d-315e-41d8-ae18-7df7c08fbb77",
  "userID": "00000000-0000-0000-0000-000000000000",
  "groupID": "8b5b54da-ae53-497a-963d-1fc89990525b",
  "accountID": "d0fdbfa7-be32-4a71-b59d-13d95b42329a",
  "role": "viewer",
  "roleConstraints": ["*"]
}
```

Observe o seguinte sobre os parâmetros de resposta:

- O valor `group` para o `principalType` campo indica que a vinculação de função foi adicionada para um grupo (não para um usuário).

Desativar e repor LDAP

Há duas tarefas administrativas opcionais relacionadas que você pode executar conforme necessário para uma implantação do Astra Control Center. Pode desativar globalmente a autenticação LDAP e repor a configuração LDAP.

Ambas as tarefas de fluxo de trabalho exigem o id para a `astra.account.ldap` configuração Astra. Detalhes sobre como recuperar o ID de configuração estão incluídos em **Configurar o servidor LDAP**. Consulte "Recupere o UUID da configuração LDAP" para obter mais informações.

- "Desativar a autenticação LDAP"
- "Redefina a configuração de autenticação LDAP"

Desativar a autenticação LDAP

Você pode executar a seguinte chamada de API REST para desativar globalmente a autenticação LDAP para uma implantação específica do Astra. A chamada atualiza a `astra.account.ldap` configuração e o `isEnabled` valor é definido como `false`.

Método HTTP	Caminho
COLOQUE	/account/_id/core/v1/settings/ [definições_id]

Exemplo de entrada JSON

```
{
  "type": "application/astra-setting",
  "version": "1.0",
  "desiredConfig": {
    "connectionHost": "myldap.example.com",
    "credentialId": "3bd9c8a7-f5a4-4c44-b778-90a85fc7d154",
    "groupBaseDN": "OU=groups,OU=astra,DC=example,DC=com",
    "isEnabled": "false",
    "port": 686,
    "secureMode": "LDAPS",
    "userBaseDN": "OU=users,OU=astra,DC=example,dc=com",
    "userSearchFilter": "((objectClass=User))",
    "vendor": "Active Directory"
    }
}
```

```
curl --location -i --request PUT --data @JSONinput
'https://astra.example.com/accounts/<ACCOUNT_ID>/core/v1/settings/<SETTING
_ID>' --header 'Content-Type: application/astra-setting+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Se a chamada for bem-sucedida, a `HTTP 204` resposta será retornada. Opcionalmente, você pode recuperar as configurações novamente para confirmar a alteração.

Redefina a configuração de autenticação LDAP

Você pode executar a seguinte chamada de API REST para desconetar o Astra do servidor LDAP e redefinir a configuração LDAP no Astra. A chamada atualiza a `astra.account.ldap` configuração e o valor de `connectionHost` é apagado.

O valor de `isEnabled` também deve ser definido como `false`. Você pode definir esse valor antes de fazer a chamada de redefinição ou como parte de fazer a chamada de redefinição. No segundo caso, `connectionHost` deve ser limpo e `isEnabled` definido como false na mesma chamada de redefinição.

Esta é uma operação disruptiva e você deve prosseguir com cuidado. Elimina todos os utilizadores e grupos LDAP importados. Ele também exclui todos os usuários, grupos e roleBindings relacionados do Astra (tipo LDAP) que você criou no Astra Control Center.

Método HTTP	Caminho
COLOQUE	/account/_id/core/v1/settings/ [definições_id]

Exemplo de entrada JSON

```
{
  "type": "application/astra-setting",
  "version": "1.0",
  "desiredConfig": {
    "connectionHost": "",
    "credentialId": "3bd9c8a7-f5a4-4c44-b778-90a85fc7d154",
    "groupBaseDN": "OU=groups,OU=astra,DC=example,DC=com",
    "isEnabled": "false",
    "port": 686,
    "secureMode": "LDAPS",
    "userBaseDN": "OU=users,OU=astra,DC=example,dc=com",
    "userSearchFilter": "((objectClass=User))",
    "vendor": "Active Directory"
    }
}
```

Observe o seguinte:

- Para alterar o servidor LDAP, você deve desabilitar e redefinir a alteração LDAP `connectHost` para um valor nulo, como mostrado no exemplo acima.

```
curl --location -i --request PUT --data @JSONinput
'https://astra.example.com/accounts/<ACCOUNT_ID>/core/v1/settings/<SETTING
_ID>' --header 'Content-Type: application/astra-setting+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Se a chamada for bem-sucedida, a `HTTP 204` resposta será retornada. Opcionalmente, você pode recuperar a configuração novamente para confirmar a alteração.

Clusters

Liste os clusters

É possível listar os clusters disponíveis em uma nuvem específica.

1. Selecione a nuvem

Execute o fluxo de trabalho "Liste as nuvens" e selecione a nuvem que contém os clusters.

2. Liste os clusters

Execute a seguinte chamada de API REST para listar os clusters em uma nuvem específica.

Método HTTP	Caminho
OBTER	/account//topology/v1/clouds/ /cloud_id/clusters

Exemplo de curl: Retorna todos os dados de todos os clusters

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

Exemplo de saída JSON


```
{
    "items": [
        {
            "type": "application/astra-cluster",
            "version": "1.1",
            "id": "7ce83fba-6aa1-4e0c-a194-26e714f5eb46",
            "name": "openshift-clstr-ol-07",
            "state": "running",
            "stateUnready": [],
            "managedState": "managed",
            "protectionState": "full",
            "protectionStateDetails": [],
            "restoreTargetSupported": "true",
            "snapshotSupported": "true",
            "managedStateUnready": [],
            "managedTimestamp": "2022-11-03T15:50:59Z",
            "inUse": "true",
            "clusterType": "openshift",
            "accHost": "true",
            "clusterVersion": "1.23",
            "clusterVersionString": "v1.23.12+6b34f32",
            "namespaces": [
                "default",
                "kube-node-lease",
                "kube-public",
                "kube-system",
```

```
                "metallb-system",
                "mysql",
                "mysql-clone1",
                "mysql-clone2",
                "mysql-clone3",
                "mysql-clone4",
                "netapp-acc-operator",
                "netapp-monitoring",
                "openshift",
                "openshift-apiserver",
                "openshift-apiserver-operator",
                "openshift-authentication",
                "openshift-authentication-operator",
                "openshift-cloud-controller-manager",
                "openshift-cloud-controller-manager-operator",
                "openshift-cloud-credential-operator",
                "openshift-cloud-network-config-controller",
                "openshift-cluster-csi-drivers",
                "openshift-cluster-machine-approver",
                "openshift-cluster-node-tuning-operator",
                "openshift-cluster-samples-operator",
                "openshift-cluster-storage-operator",
                "openshift-cluster-version",
                "openshift-config",
                "openshift-config-managed",
                "openshift-config-operator",
                "openshift-console",
                "openshift-console-operator",
                "openshift-console-user-settings",
                "openshift-controller-manager",
                "openshift-controller-manager-operator",
                "openshift-dns",
                "openshift-dns-operator",
                "openshift-etcd",
                "openshift-etcd-operator",
                "openshift-host-network",
                "openshift-image-registry",
                "openshift-infra",
                "openshift-ingress",
                "openshift-ingress-canary",
                "openshift-ingress-operator",
                "openshift-insights",
                "openshift-kni-infra",
                "openshift-kube-apiserver",
                "openshift-kube-apiserver-operator",
                "openshift-kube-controller-manager",
```

```
                "openshift-kube-controller-manager-operator",
                "openshift-kube-scheduler",
                "openshift-kube-scheduler-operator",
                "openshift-kube-storage-version-migrator",
                "openshift-kube-storage-version-migrator-operator",
                "openshift-machine-api",
                "openshift-machine-config-operator",
                "openshift-marketplace",
                "openshift-monitoring",
                "openshift-multus",
                "openshift-network-diagnostics",
                "openshift-network-operator",
                "openshift-node",
                "openshift-oauth-apiserver",
                "openshift-openstack-infra",
                "openshift-operator-lifecycle-manager",
                "openshift-operators",
                "openshift-ovirt-infra",
                "openshift-sdn",
                "openshift-service-ca",
                "openshift-service-ca-operator",
                "openshift-user-workload-monitoring",
                "openshift-vsphere-infra",
                "pcloud",
                "postgreql",
                "trident"
            ],
            "defaultStorageClass": "4bacbb3c-0727-4f58-b13c-3a2a069baf89",
            "cloudID": "4f1e1086-f415-4451-a051-c7299cd672ff",
            "credentialID": "7ffd7354-b6c2-4efa-8e7b-cf64d5598463",
            "isMultizonal": "false",
            "tridentManagedStateAllowed": [
                "unmanaged"
            ],
            "tridentVersion": "22.10.0",
            "apiServiceID": "98df44dc-2baf-40d5-8826-e198b1b40909",
            "metadata": {
                "labels": [
                    {
                        "name": "astra.netapp.io/labels/read-
only/cloudName",
                        "value": "private"
                    }
                ],
                "creationTimestamp": "2022-11-03T15:50:59Z",
                "modificationTimestamp": "2022-11-04T14:42:32Z",
```

```
                "createdBy": "00000000-0000-0000-0000-000000000000"
            }
        }
    ]
}
```


Listar clusters gerenciados

É possível listar os clusters de Kubernetes gerenciados atualmente pelo Astra.

1. Listar os clusters gerenciados

Execute a seguinte chamada de API REST.

Método HTTP	Caminho
OBTER	/Account/_id/topology/v1/managedclusters

Exemplo de curl: Retorna todos os dados de todos os clusters

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

Nuvens

Liste as nuvens

Você pode listar as nuvens definidas e disponíveis de uma conta específica do Astra.

1. Liste as nuvens

Execute a seguinte chamada de API REST para listar as nuvens.

Método HTTP	Caminho
OBTER	/account/_id/topology/v1/clouds

Curl exemplo: Retorna todos os dados para todas as nuvens

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

Baldes

Liste os buckets

Você pode listar os buckets do S3 definidos para uma conta específica do Astra.

1. Liste os buckets

Execute a seguinte chamada de API REST para listar os buckets.

Método HTTP	Caminho
OBTER	/account/_id/topology/v1/buckets

Curl exemplo: Retorna todos os dados para todos os buckets

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

Armazenamento

Listar classes de armazenamento

Você pode listar as classes de armazenamento disponíveis.

1. Selecione a nuvem

Execute o fluxo de trabalho "Liste as nuvens" e selecione a nuvem na qual você estará trabalhando.

2. Selecione o cluster

Execute o fluxo de trabalho "Liste os clusters" e selecione o cluster.

3. Liste as classes de armazenamento de um cluster específico

Execute a seguinte chamada de API REST para listar as classes de armazenamento de um cluster e nuvem específicos.

Método HTTP	Caminho
OBTER	/Account//topology/v1/clouds/<CLOUD_ID>/clusters/<CLUSTER_ID>/storageClasses

Curl exemplo: Retorna todos os dados para todas as classes de armazenamento

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/clouds/<CLOUD_I
D>/clusters/<CLUSTER_ID>/storageClasses' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>'
```

Exemplo de saída JSON


```
{
    "items": [
        {
            "type": "application/astra-storageClass",
            "version": "1.1",
            "id": "4bacbb3c-0727-4f58-b13c-3a2a069baf89",
            "name": "ontap-basic",
            "provisioner": "csi.trident.netapp.io",
            "available": "eligible",
            "allowVolumeExpansion": "true",
            "reclaimPolicy": "Delete",
            "volumeBindingMode": "Immediate",
            "isDefault": "true",
            "metadata": {
                "createdBy": "system",
                "creationTimestamp": "2022-10-26T05:16:19Z",
                "modificationTimestamp": "2022-10-26T05:16:19Z",
                "labels": []
            }
        },
        {
            "type": "application/astra-storageClass",
            "version": "1.1",
            "id": "150fe657-4a42-47a3-abc6-5dafba3de8bf",
            "name": "thin",
            "provisioner": "kubernetes.io/vsphere-volume",
            "available": "ineligible",
            "reclaimPolicy": "Delete",
            "volumeBindingMode": "Immediate",
            "metadata": {
                "createdBy": "system",
                "creationTimestamp": "2022-10-26T04:46:08Z",
                "modificationTimestamp": "2022-11-04T14:58:19Z",
                "labels": []
            }
        },
        {
```

```
            "type": "application/astra-storageClass",
            "version": "1.1",
            "id": "7c6a5c58-6a0d-4cb6-98a0-8202ad2de74a",
            "name": "thin-csi",
            "provisioner": "csi.vsphere.vmware.com",
            "available": "ineligible",
            "allowVolumeExpansion": "true",
            "reclaimPolicy": "Delete",
            "volumeBindingMode": "WaitForFirstConsumer",
            "metadata": {
                "createdBy": "system",
                "creationTimestamp": "2022-10-26T04:46:17Z",
                "modificationTimestamp": "2022-10-26T04:46:17Z",
                "labels": []
            }
        },
        {
            "type": "application/astra-storageClass",
            "version": "1.1",
            "id": "7010ef09-92a5-4c90-a5e5-3118e02dc9a7",
            "name": "vsim-san",
            "provisioner": "csi.trident.netapp.io",
            "available": "eligible",
            "allowVolumeExpansion": "true",
            "reclaimPolicy": "Delete",
            "volumeBindingMode": "Immediate",
            "metadata": {
                "createdBy": "system",
                "creationTimestamp": "2022-11-03T18:40:03Z",
                "modificationTimestamp": "2022-11-03T18:40:03Z",
                "labels": []
            }
        }
    ]
}
```


Listar backends de armazenamento

Você pode listar os backends de armazenamento disponíveis.

1. Liste os backends

Execute a seguinte chamada de API REST.

Método HTTP	Caminho
OBTER	/Account/_id/topology/v1/storageBackends

Curl exemplo: Retorna todos os dados para todos os backends de armazenamento

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

Exemplo de saída 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"
    }
  ]
}
```

Fluxos de trabalho de gerenciamento

Antes de começar

Use esses workflows como parte da administração das aplicações em um cluster gerenciado do Astra.

Esses fluxos de trabalho podem ser expandidos e aprimorados pelo NetApp a qualquer momento, portanto, você deve revisá-los periodicamente.

Preparação geral

Antes de usar qualquer um dos workflows do Astra, revise "Prepare-se para usar os fluxos de trabalho"o .

Categorias de fluxo de trabalho

Os fluxos de trabalho de gerenciamento são organizados em diferentes categorias para facilitar a localização do que você deseja.

Categoria	Descrição
Controlo de aplicação	Esses fluxos de trabalho permitem que você controle os aplicativos gerenciados e não gerenciados. Você pode listar os aplicativos, bem como criar e remover um aplicativo gerenciado.
Proteção de aplicação	Use esses fluxos de trabalho para proteger suas aplicações gerenciadas por meio de snapshots e backups.
Clonagem e restauração de aplicações	Esse fluxo de trabalho descreve como clonar e restaurar aplicativos gerenciados.
Suporte	Há vários fluxos de trabalho disponíveis para depurar e dar suporte às suas aplicações, bem como o ambiente geral do Kubernetes.

Considerações adicionais

Há várias considerações adicionais ao usar os fluxos de trabalho de gerenciamento.

Clonar uma aplicação

Há algumas coisas a considerar ao clonar um aplicativo. Os parâmetros descritos abaixo fazem parte da entrada JSON.

Identificador do cluster de origem

O valor de `sourceClusterID` sempre identifica o cluster onde o aplicativo original está instalado.

Identificador de cluster

O valor de `clusterID` identifica o cluster onde o novo aplicativo será instalado.

- Ao clonar dentro do mesmo cluster `clusterID` e `sourceClusterID` ter o mesmo valor.
- Ao clonar entre clusters, os dois valores são diferentes e `clusterID` devem ser a ID do cluster de

destino.

Namespaces

O `namespace` valor deve ser diferente do aplicativo de origem original. Além disso, o namespace para o clone não pode existir e o Astra o criará.

Backups e snapshots

Opcionalmente, você pode clonar um aplicativo a partir de um backup ou snapshot existente usando os `backupID` parâmetros ou `snapshotID`. Se você não fornecer um backup ou snapshot, o Astra criará primeiro um backup da aplicação e, em seguida, clonará a partir do backup.

Restaurar uma aplicação

Aqui estão algumas coisas a considerar ao restaurar um aplicativo.

- Restaurar um aplicativo é muito semelhante à operação clone.
- Ao restaurar um aplicativo, você deve fornecer um backup ou um snapshot.

Controlo de aplicações

Liste as aplicações

Você pode listar as aplicações que atualmente são gerenciadas pelo Astra. Você pode fazer isso como parte de encontrar os snapshots ou backups de um aplicativo específico.

1. Liste os aplicativos

Execute a seguinte chamada de API REST.

Método HTTP	Caminho
OBTER	/account//k8s/v2/apps

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
incluir	Consulta	Não	Opcionalmente, selecione os valores que você deseja retornar na resposta.

Curl exemplo: Retorna todos os dados para todos os aplicativos

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Curl exemplo: Retorna o nome, id e estado de todos os aplicativos

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps?include=name,id
,state' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Exemplo de saída JSON


```
{
    "items": [
        [
            "mysql",
            "4ee2b8fa-3696-4f32-8879-399792f477c3",
            "ready"
        ],
        [
            "postgresql",
            "3b984474-e5c9-4b64-97ee-cdeb9bcd212e",
            "ready"
        ],
    ],
    "metadata": {}
}
```


Obtenha uma aplicação

Você pode recuperar todas as variáveis de recurso descrevendo um único aplicativo.

Antes de começar

Você deve ter o ID do aplicativo que deseja recuperar. Se necessário, você pode usar o fluxo de trabalho "Liste as aplicações" para localizar o aplicativo.

1. Obtenha o aplicativo

Execute a seguinte chamada de API REST.

Método HTTP	Caminho
OBTER	/accounts/_id/k8s/v2/apps//app_id

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
id da aplicação	Caminho	Sim	Valor de ID da aplicação a recuperar.

Curl exemplo: Retorna todos os dados para o aplicativo

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps/<APP_ID>'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Gerenciar um aplicativo

Você pode criar uma aplicação gerenciada com base em uma aplicação já conhecida pelo Astra em um namespace específico. Quando uma aplicação é gerenciada ou definida no Astra, você pode protegê-la fazendo backups e snapshots.

1. Selecione o namespace

Execute o fluxo de trabalho "Liste os namespaces" e selecione o namespace.

2. Selecione o cluster

Execute o fluxo de trabalho "Liste os clusters" e selecione o cluster.

3. Gerencie o aplicativo

Execute a seguinte chamada de API REST para gerenciar o aplicativo.

Método HTTP	Caminho
POST	/account//k8s/v2/apps

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
JSON	Corpo	Sim	Fornece os parâmetros necessários para identificar o aplicativo a ser gerenciado. Veja o exemplo abaixo.

Exemplo de entrada JSON

```
{
  "clusterID": "7ce83fba-6aa1-4e0c-a194-26e714f5eb46",
  "name": "subtext",
  "namespaceScopedResources": [{"namespace": "kube-matrix"}],
  "type": "application/astra-app",
  "version": "2.0"
}
```

Curl exemplo: Gerencie um aplicativo

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Content-Type: application/astra-app+json' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>'  --data @JSONinput
```

Desgerenciar um aplicativo

Você pode remover um aplicativo gerenciado quando não for mais necessário. A remoção de um aplicativo gerenciado também exclui as programações associadas.

Antes de começar

Você deve ter o ID do aplicativo que deseja desgerenciar. Se necessário, você pode usar o fluxo de trabalho "Liste as aplicações" para localizar o aplicativo.

Os backups e snapshots do aplicativo não são removidos automaticamente quando são excluídos. Se você não precisar mais dos backups e snapshots, exclua-os antes de remover o aplicativo.

1. Não gerido a aplicação

Execute a seguinte chamada de API REST para remover o aplicativo.

Método HTTP	Caminho
ELIMINAR	/accounts/_id/k8s/v2/apps//app_id

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
id da aplicação	Caminho	Sim	Identifica o aplicativo a ser removido.

Curl exemplo: Remover um aplicativo gerenciado

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps/<APP_ID>'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Proteção da aplicação

Liste os instantâneos

Você pode listar os snapshots que foram obtidos para um aplicativo específico.

Antes de começar

Você deve ter o ID do aplicativo para o qual deseja listar os snapshots. Se necessário, você pode usar o fluxo de trabalho "Liste as aplicações" para localizar o aplicativo.

1. Liste os instantâneos

Execute a seguinte chamada de API REST para listar os snapshots.

Método HTTP	Caminho
OBTER	/Accounts/_id/k8s/v1/apps//app_id/appSnaps

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
id da aplicação	Caminho	Sim	Identifica o aplicativo que possui os snapshots listados.
contar	Consulta	Não	Se `count=true` o número de instantâneos estiver incluído na seção metadados da resposta.

Curl exemplo: Retornar todos os snapshots para o aplicativo

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

Curl exemplo: Retorna todos os instantâneos para o aplicativo e a contagem

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appSna
ps?count=true' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Exemplo de saída JSON


```
{
    "items": [
        {
            "type": "application/astra-appSnap",
            "version": "1.1",
            "id": "1ce34da4-bb0a-4926-b925-4a5d85dda8c2",
            "hookState": "success",
            "metadata": {
                "createdBy": "a530e865-23e8-4e2e-8020-e92c419a3867",
                "creationTimestamp": "2022-10-30T22:44:20Z",
                "modificationTimestamp": "2022-10-30T22:44:20Z",
                "labels": []
            },
            "snapshotAppAsset": "0ebfe3f8-40ed-4bdc-88c4-2144fbda85a0",
            "snapshotCreationTimestamp": "2022-10-30T22:44:33Z",
            "name": "snapshot-david-1",
            "state": "completed",
            "stateUnready": []
        }
    ],
    "metadata": {}
}
```


Liste os backups

Você pode listar os backups que foram criados para um aplicativo específico.

Antes de começar

Você deve ter o ID do aplicativo para o qual deseja listar os backups. Se necessário, você pode usar o fluxo de trabalho "Liste as aplicações" para localizar o aplicativo.

1. Liste os backups

Execute a seguinte chamada de API REST.

Método HTTP	Caminho
OBTER	/Accounts/_id/k8s/v1/apps//app_id/appBackups

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
id da aplicação	Caminho	Sim	Identifica o aplicativo gerenciado que possui os backups listados.

Curl exemplo: Retornar todos os backups para o aplicativo

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

Exemplo de saída JSON

```
{
    "items": [
        {
            "type": "application/astra-appBackup",
            "version": "1.1",
            "id": "8edeb4a4-fd8b-4222-a559-1013145b28fc",
            "name": "backup-david-oct28-1",
            "bucketID": "a443e58f-59bd-4d45-835a-1bc7813f659a",
            "snapshotID": "dfe237cb-57b7-4576-af4d-00ba3a8f2828",
            "state": "completed",
            "stateUnready": [],
            "hookState": "success",
            "totalBytes": 205219132,
            "bytesDone": 205219132,
            "percentDone": 100,
            "metadata": {
                "labels": [
                    {
                        "name": "astra.netapp.io/labels/read-
only/triggerType",
                        "value": "backup"
                    }
                ],
                "creationTimestamp": "2022-10-28T21:58:37Z",
                "modificationTimestamp": "2022-10-28T21:58:55Z",
                "createdBy": "a530e865-23e8-4e2e-8020-e92c419a3867"
            }
        }
    ],
    "metadata": {}
}
```


Crie um instantâneo para um aplicativo

Você pode criar um instantâneo para um aplicativo específico.

Antes de começar

Você deve ter o ID do aplicativo para o qual deseja criar um instantâneo. Se necessário, você pode usar o fluxo de trabalho "Liste as aplicações" para localizar o aplicativo.

1. Criar um instantâneo

Execute a seguinte chamada de API REST.

Método HTTP	Caminho
POST	/Accounts/_id/k8s/v1/apps//app_id/appSnaps

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
id da aplicação	Caminho	Sim	Identifica o aplicativo gerenciado onde o snapshot será criado.
JSON	Corpo	Sim	Fornece os parâmetros para o instantâneo. Veja o exemplo abaixo.

Exemplo de entrada JSON

```
{
  "type": "application/astra-appSnap",
  "version": "1.1",
  "name": "snapshot-david-1"
}
```

Curl exemplo: Crie um instantâneo para o aplicativo

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appSna
ps' --header 'Content-Type: application/astra-appSnap+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --data
@JSONinput
```

Crie uma cópia de segurança para uma aplicação

Você pode criar um backup para um aplicativo específico e usar o backup para restaurar ou clonar o aplicativo.

Antes de começar

Você deve ter o ID do aplicativo que deseja fazer backup. Se necessário, você pode usar o fluxo de trabalho "Liste as aplicações" para localizar o aplicativo.

1. Crie uma cópia de segurança

Execute a seguinte chamada de API REST.

Método HTTP	Caminho
POST	/Accounts/_id/k8s/v1/apps//app_id/appBackups

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
id da aplicação	Caminho	Sim	Identifica o aplicativo onde o backup será criado.
JSON	Corpo	Sim	Fornece os parâmetros para o backup. Veja o exemplo abaixo.

Exemplo de entrada JSON

```
{
  "type": "application/astra-appBackup",
  "version": "1.1",
  "name": "backup-david-1"
}
```

Curl exemplo: Crie um backup para o aplicativo

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appBac
kups' --header 'Content-Type: application/astra-appBackup+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --data
@JSONinput
```

Eliminar um instantâneo

Pode eliminar um instantâneo associado a uma aplicação.

Antes de começar

Você deve ter o seguinte:

- ID do aplicativo que possui o snapshot. Se necessário, você pode usar o fluxo de trabalho "Liste as aplicações" para localizar o aplicativo.
- ID do instantâneo que pretende eliminar. Se necessário, você pode usar o fluxo de trabalho "Liste os instantâneos" para localizar o instantâneo.

1. Eliminar o instantâneo

Execute a seguinte chamada de API REST.

Método HTTP	Caminho
ELIMINAR	/Accounts/_id/k8s/v1/apps//app_id/appSnaps//appSnap_id

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
id da aplicação	Caminho	Sim	Identifica o aplicativo gerenciado que possui o snapshot.
id do instantâneo	Caminho	Sim	Identifica o instantâneo a ser eliminado.

Curl exemplo: Exclua um único snapshot para o aplicativo

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appSna
ps/<SNAPSHOT_ID>' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Eliminar uma cópia de segurança

Pode eliminar uma cópia de segurança associada a uma aplicação.

Antes de começar

Você deve ter o seguinte:

- ID do aplicativo que possui o backup. Se necessário, você pode usar o fluxo de trabalho "Liste as aplicações" para localizar o aplicativo.
- ID da cópia de segurança que pretende eliminar. Se necessário, você pode usar o fluxo de trabalho "Liste os backups" para localizar o instantâneo.

1. Eliminar a cópia de segurança

Execute a seguinte chamada de API REST.

> Você pode forçar a exclusão de um backup com falha usando o cabeçalho de solicitação opcional, conforme descrito abaixo.

Método HTTP	Caminho
ELIMINAR	/Accounts/ k8s/v1/apps//app_id/appBackups/ appBackup_id

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
id da aplicação	Caminho	Sim	Identifica o aplicativo gerenciado que possui o backup.
id de cópia de segurança	Caminho	Sim	Identifica o backup a ser excluído.
forçar a eliminação	Colhedor	Não	Usado para forçar a exclusão de um backup com falha.

Curl exemplo: Exclua um único backup para o aplicativo

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appBac
kups/<BACKUP_ID>' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Curl exemplo: Exclua um único backup para o aplicativo com a opção forçar

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appBac
kups/<BACKUP_ID>' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>' --header 'Force-Delete: true'
```

Clonar e restaurar um aplicativo

Clonar uma aplicação

É possível criar uma nova aplicação clonando uma aplicação existente.

Antes de começar

Observe o seguinte sobre este fluxo de trabalho:

- Não é utilizado um backup ou instantâneo da aplicação
- A operação de clone é executada no mesmo cluster
- O novo aplicativo é colocado em um namespace diferente

> Para clonar um aplicativo para um cluster diferente, você precisa atualizar o `clusterId` parâmetro na entrada JSON conforme apropriado para o seu ambiente.

1. Selecione o aplicativo para clonar

Execute o fluxo de trabalho "Liste as aplicações" e selecione a aplicação que deseja clonar. Vários dos valores

de recursos são necessários para a chamada REST usada para clonar o aplicativo.

2. Clonar a aplicação

Execute a seguinte chamada de API REST para clonar o aplicativo.

Método HTTP	Caminho
POST	/account//k8s/v2/apps

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
JSON	Corpo	Sim	Fornece os parâmetros para o aplicativo clonado. Veja o exemplo abaixo.

Exemplo de entrada JSON

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "name": "mysql-clone",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "mysql-ns",
  "sourceAppID": "e591ee59-ea90-4a9f-8e6c-d2b6e8647096"
}
```

Curl exemplo: Clonar um aplicativo

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Content-Type: application/astra-app+json' --header '*/*' --header
'Authorization: Bearer <API_TOKEN>' --data @JSONinput
```

Clonar um aplicativo a partir de um snapshot

Crie uma nova aplicação clonando-a a partir de um snapshot.

Antes de começar

Observe o seguinte sobre este fluxo de trabalho:

- É utilizado um instantâneo da aplicação
- A operação de clone é executada no mesmo cluster

Para clonar um aplicativo para um cluster diferente, você precisa atualizar o `clusterId` parâmetro na entrada JSON conforme apropriado para o seu ambiente.

1. Selecione o aplicativo para clonar

Execute o fluxo de trabalho "Liste as aplicações" e selecione a aplicação que deseja clonar. Vários dos valores de recursos são necessários para a chamada REST usada para clonar o aplicativo.

2. Selecione o instantâneo a utilizar

Execute o fluxo de trabalho "Liste os instantâneos" e selecione instantâneo que deseja usar.

3. Clonar a aplicação

Execute a seguinte chamada de API REST.

Método HTTP	Caminho
POST	/account//k8s/v2/apps

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
JSON	Corpo	Sim	Fornece os parâmetros para o aplicativo clonado. Veja o exemplo abaixo.

Exemplo de entrada JSON

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "name": "mysql-clone2",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "mysql",
  "snapshotID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"
}
```

Curl exemplo: Clonar um aplicativo a partir de um snapshot

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Content-Type: application/astra-app+json' --header '*/*' --header
'Authorization: Bearer <API_TOKEN>' --data @JSONinput
```

Clonar um aplicativo a partir de um backup

Crie uma nova aplicação clonando-a a partir de um backup.

Antes de começar

Observe o seguinte sobre este fluxo de trabalho:

- Um backup de aplicativo é usado
- A operação de clone é executada no mesmo cluster

> Para clonar um aplicativo para um cluster diferente, você precisa atualizar o `clusterId` parâmetro na entrada JSON conforme apropriado para o seu ambiente.

1. Selecione o aplicativo para clonar

Execute o fluxo de trabalho "Liste as aplicações" e selecione a aplicação que deseja clonar. Vários dos valores de recursos são necessários para a chamada REST usada para clonar o aplicativo.

2. Selecione a cópia de segurança a utilizar

Execute o fluxo de trabalho "Liste os backups" e selecione cópia de segurança que pretende utilizar.

3. Clonar a aplicação

Execute a seguinte chamada de API REST.

Método HTTP	Caminho
POST	/account//k8s/v2/qpps

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
JSON	Corpo	Sim	Fornece os parâmetros para o aplicativo clonado. Veja o exemplo abaixo.

Exemplo de entrada JSON

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "name": "mysql-clone3",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "mysql",
  "backupID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"
}
```

Curl exemplo: Clonar um aplicativo a partir de um backup

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Content-Type: application/astra-app+json' --header '*/*' --header
'Authorization: Bearer <API_TOKEN>' --data @JSONinput
```

Restaurar uma aplicação a partir de uma cópia de segurança

Você pode restaurar um aplicativo criando um novo aplicativo a partir de um backup.

1. Selecione a aplicação a restaurar

Execute o fluxo de trabalho "Liste as aplicações" e selecione a aplicação que deseja clonar. Vários dos valores de recursos são necessários para a chamada REST usada para restaurar o aplicativo.

2. Selecione a cópia de segurança a utilizar

Execute o fluxo de trabalho "Liste os backups" e selecione cópia de segurança que pretende utilizar.

3. Restaure a aplicação

Execute a seguinte chamada de API REST. Você deve fornecer o ID para um backup (como mostrado abaixo) ou snapshot.

Método HTTP	Caminho
COLOQUE	/account/_id/k8s/v2/apps//app_id

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
JSON	Corpo	Sim	Fornece os parâmetros para o aplicativo clonado. Veja o exemplo abaixo.

Exemplo de entrada JSON

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "backupID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"
}
```

Curl exemplo: Restaure um aplicativo no lugar a partir de um backup

```
curl --location -i --request PUT
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps/<APP_ID>'
--header 'Content-Type: application/astra-app+json' --header '*/*'
--header 'ForceUpdate: true' --header 'Authorization: Bearer <API_TOKEN>'
--data @JSONinput
```

Namespaces

Liste os namespaces

Você pode listar os namespaces disponíveis.

1. Liste os namespaces

Execute a seguinte chamada de API REST para listar os namespaces.

Método HTTP	Caminho
OBTER	/account/_id/topology/v1/namespaces

Curl exemplo: Retorna todos os dados para todos os namespaces

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

Curl exemplo: Retorna nome, estado e ID de cluster para todos os namespaces

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/namespaces?incl
ude=name,namespaceState,clusterID' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>'
```

Exemplo de saída JSON


```
{
    "items": [
        [
            "default",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "kube-node-lease",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "kube-public",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "kube-system",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "mysql",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "mysql-clone1",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "netapp-acc-operator",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
```

```
        [
            "openshift",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "trident",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ]
    ],
    "metadata": {}
}
```

Suporte

Liste as notificações

Você pode listar as notificações de uma conta Astra específica. Você pode fazer isso como parte do monitoramento da atividade do sistema ou depuração de um problema.

1. Liste as notificações

Execute a seguinte chamada de API REST.

Método HTTP	Caminho
OBTER	/account/_id/core/v1/notificações

Parâmetros de entrada adicionais

Além dos parâmetros comuns com todas as chamadas de API REST, os seguintes parâmetros também são usados nos exemplos curl para esta etapa.

Parâmetro	Tipo	Obrigatório	Descrição
filtro	Consulta	Não	Opcionalmente, filtre as notificações que você deseja retornar na resposta.
incluir	Consulta	Não	Opcionalmente, selecione os valores que você deseja retornar na resposta.

Curl exemplo: Retornar todas as notificações

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

Curl exemplo: Retorne a descrição para notificações com gravidade de aviso

```
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>'
```

Exemplo de saída 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 uma aplicação com falha

Talvez você não consiga remover um aplicativo gerenciado se ele tiver um backup ou snapshot em um estado com falha. Neste caso, você pode remover manualmente o aplicativo usando o fluxo de trabalho descrito abaixo.

1. Selecione a aplicação a eliminar

Execute o fluxo de trabalho "Liste as aplicações" e selecione a aplicação que pretende remover.

2. Liste os backups existentes para o aplicativo

Execute o fluxo de "Liste os backups"trabalho .

3. Exclua todos os backups

Exclua todos os backups de aplicativos executando o fluxo de trabalho "Eliminar uma cópia de segurança" para cada backup na lista.

4. Liste os instantâneos existentes para o aplicativo

Execute o fluxo de "Liste os instantâneos"trabalho .

5. Eliminar todos os instantâneos

Execute o fluxo de trabalho "Eliminar um instantâneo"de cada instantâneo na lista.

6. Remova a aplicação

Execute o fluxo de trabalho "Desgerenciar um aplicativo" para remover a aplicação.

Usando Python

O NetApp já está disponível

O NetApp é um pacote de código aberto que você pode usar para automatizar uma implantação do Astra Control. O pacote também é um recurso valioso para aprender sobre a API REST do Astra Control, talvez como parte da criação de sua própria plataforma de automação.

Para simplificar, o SDK do NetApp será referido como o **SDK** ao longo do restante desta página.

Duas ferramentas de software relacionadas

O SDK inclui duas ferramentas diferentes, embora relacionadas, que operam em diferentes níveis de abstração ao acessar a API REST do Astra Control.

SDK do Astra

O Astra SDK fornece a funcionalidade principal da plataforma. Ele inclui um conjunto de classes Python que abstraem as chamadas de API REST subjacentes. As classes dão suporte a ações administrativas em vários recursos do Astra Control, incluindo aplicações, backups, snapshots e clusters.

O Astra SDK é uma parte do pacote e é fornecido em um único `astraSDK.py` arquivo. Você pode importar esse arquivo para o seu ambiente e usar as classes diretamente.

O **NetApp é o nome de todo o pacote. O *Astra SDK** refere-se às classes Python principais no único arquivo `astraSDK.py`.

Script do Toolkit

Além do arquivo Astra SDK, o `toolkit.py` script também está disponível. Este script opera em um nível mais alto de abstração, fornecendo acesso a ações administrativas discretas definidas internamente como funções Python. O script importa o Astra SDK e faz chamadas para as classes conforme necessário.

Como aceder

Você pode acessar o SDK das seguintes maneiras.

Pacote Python

O SDK está disponível em "Índice do Pacote Python" sob o nome **actoolkit**. O pacote recebe um número de versão e continuará a ser atualizado conforme necessário. Você deve usar o utilitário de gerenciamento de pacotes **PIP** para instalar o pacote em seu ambiente.

Uma vez instalado, as classes astraSDK.py podem ser utilizadas colocando `import astraSDK` em seus scripts. Além disso, `actoolkit` pode ser invocado diretamente no seu prompt de comando e é equivalente a `toolkit.py` (`actoolkit list clusters` é igual `./toolkit.py list clusters`a).

Consulte "PyPI: NetApp é um dos nossos selecionados Jogos de Plataforma" para obter mais informações.

Código-fonte do GitHub

O código-fonte do SDK também está disponível no GitHub. O repositório inclui o seguinte:

- `astraSDK.py` (Astra SDK com classes Python)
- `toolkit.py` (script baseado em funções de nível superior)
- Requisitos e instruções de instalação detalhadas
- Scripts de instalação
- Documentação adicional

Você pode clonar o "GitHub: NetApp/NetApp-astra-toolkits" repositório para o seu ambiente local.

Instalação e requisitos básicos

Existem várias opções e requisitos a considerar como parte da instalação do pacote e preparação para usá-lo.

Resumo das opções de instalação

Você pode instalar o SDK de uma das seguintes maneiras:

- Use a imagem preparada "Docker: Kits de ferramentas NetApp/astra", que tem todas as dependências necessárias instaladas, incluindo `actoolkit`
- Use o PIP para instalar o `actoolkit` pacote do PyPI em seu ambiente Python
- Clone o repositório do GitHub e copie/modifique os dois arquivos Python principais para que eles estejam acessíveis ao seu código de cliente Python

Consulte as páginas PyPI e GitHub para obter mais informações.

Requisitos para o ambiente Astra Control

Seja usando diretamente as classes Python no Astra SDK ou as funções no `toolkit.py` script, você estará acessando a API REST em uma implantação do Astra Control. Por causa disso, você precisará de uma conta Astra juntamente com um token de API. Consulte "Antes de começar" e as outras páginas na seção **Introdução** desta documentação para obter mais informações.

Requisitos para o SDK Python do NetApp

O SDK tem vários pré-requisitos relacionados ao ambiente Python local. Por exemplo, você deve usar Python 3,8 ou posterior. Além disso, existem vários pacotes Python que são necessários. Consulte a página do repositório do GitHub ou a página do pacote PyPI para obter mais informações.

Resumo dos recursos úteis

Aqui estão alguns dos recursos que você precisará para começar.

- "PyPI: NetApp é um dos nossos selecionados Jogos de Plataforma"
- "GitHub: NetApp/NetApp-astra-toolkits"
- "Docker: Kits de ferramentas NetApp/astra"

Python nativo

Antes de começar

Python é uma linguagem de desenvolvimento popular para automação de data center. Antes de usar os recursos nativos do Python juntamente com vários pacotes comuns, você precisa preparar o ambiente e os arquivos de entrada necessários.

Além de acessar a API REST do Astra Control diretamente usando Python, o NetApp também fornece um pacote de kit de ferramentas que abstrai a API e remove algumas das complexidades. Consulte "O NetApp já está disponível" para obter mais informações.

Prepare o ambiente

Os requisitos básicos de configuração para executar os scripts Python são descritos abaixo.

Python 3

Você precisa ter a versão mais recente do Python 3 instalada.

Bibliotecas adicionais

As bibliotecas **requests** e **urllib3** devem ser instaladas. Você pode usar o pip ou outra ferramenta de gerenciamento Python conforme apropriado para o seu ambiente.

Acesso à rede

A estação de trabalho onde os scripts são executados deve ter acesso à rede e ser capaz de alcançar o Astra Control. Ao usar o Astra Control Service, você deve estar conetado à Internet e ser capaz de se conetar ao serviço em `https://astra.netapp.io`.

Informações de identidade

Você precisa de uma conta Astra válida com o identificador de conta e token de API. Consulte "Obtenha um token de API" para obter mais informações.

Crie os arquivos de entrada JSON

Os scripts Python dependem de informações de configuração contidas em arquivos de entrada JSON. Os arquivos de amostra são fornecidos abaixo.

Você precisa atualizar as amostras conforme apropriado para o seu ambiente.

Informações de identidade

O arquivo a seguir contém o token de API e a conta Astra. Você precisa passar esse arquivo para scripts Python usando o `-i` parâmetro CLI (ou `--identity`).

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

Liste as aplicações

Você pode usar o script a seguir para listar os aplicativos da sua conta Astra.

"Antes de começar"Consulte para obter um exemplo do arquivo de entrada JSON necessário.

```
#!/usr/bin/env python3
##-----------------------------------------------------------------------------
------
#
# Usage: python3 list_man_apps.py -i identity_file.json
#
# (C) Copyright 2022 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 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/v2/apps"

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

Referência da API

Você pode acessar os detalhes das chamadas da API REST do Astra Control, incluindo os métodos HTTP, parâmetros de entrada e respostas. Essa referência completa é útil ao desenvolver aplicativos de automação usando a API REST.

A documentação de referência da API REST é fornecida atualmente com o Astra Control e está disponível on-line.

Antes de começar

Você precisa de uma conta para Astra Control Center ou Astra Control Service.

Passos

1. Faça login no Astra usando suas credenciais de conta.

   Acesse o seguinte site do Astra Control Service: "https://astra.netapp.io"

2. Clique no ícone de figura no canto superior direito da página e selecione **Acesso à API**.
3. Na parte superior da página, clique no URL exibido em **Documentação da API**.
4. Forneça as credenciais da sua conta novamente, se solicitado.

Recursos adicionais

Há recursos adicionais que você pode acessar para obter ajuda e encontrar mais informações sobre os serviços e suporte em nuvem da NetApp, bem como conceitos GERAIS DE REST e nuvem.

Astra

- "Documentação do Astra Control Center 22,08"

  Documentação da versão atual do software Astra Control Center implantado nas instalações do cliente.

- "Documentação do Astra Control Service"

  Documentação da versão atual do software Astra Control Service disponível na nuvem pública.

- "Documentação do Astra Trident"

  Documentação da versão atual do software Astra Trident, um orquestrador de storage de código aberto mantido pela NetApp.

- "Documentação da família Astra"

  Local central para acessar toda a documentação do Astra para implantações locais e de nuvem pública.

Recursos de nuvem da NetApp

- "NetApp BlueXP"

  Local central para as soluções de nuvem da NetApp.

- "Console central da nuvem NetApp"

  Console de serviço NetApp Cloud Central com login.

- "Suporte à NetApp"

  Acesse ferramentas de solução de problemas, documentação e assistência técnica.

Conceitos DE REST e nuvem

- PhD "dissertação" por Roy Fielding

  Esta publicação introduziu e estabeleceu o modelo de desenvolvimento de aplicações REST.

- "Auth0"

  Este é o serviço de plataforma de autenticação e autorização usado pelo serviço Astra para acesso à Web.

- "Editor RFC"

Fonte autorizada para padrões da Web e da Internet mantida como uma coleção de documentos RFC numerados de forma única.

Versões anteriores da documentação do Astra Control Automation

Você pode acessar a documentação de automação de versões anteriores do Astra Control nos links abaixo.

- "Documentação do Astra Control Automation 22,04"
- "Documentação do Astra Control Automation 21,12"
- "Documentação do Astra Control Automation 21,08"

Avisos legais

Avisos legais fornecem acesso a declarações de direitos autorais, marcas registradas, patentes e muito mais.

Direitos de autor

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

Marcas comerciais

NetApp, o logotipo DA NetApp e as marcas listadas na página de marcas comerciais da NetApp são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respetivos proprietários.

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

Patentes

Uma lista atual de patentes de propriedade da NetApp pode ser encontrada em:

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

Política de privacidade

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

Licença de API Astra Control

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

Informações sobre direitos autorais

Informações sobre marcas comerciais