Workflows de infraestrutura

Astra Automation

NetApp
January 31, 2025

Índice

Workflows de infraestrutura

Prepare-se para usar os workflows de infraestrutura

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.
Clusters	É possível adicionar clusters gerenciados do Kubernetes que permitem proteger e dar suporte às aplicações que eles contêm.
Nuvens	Esses workflows fornecem acesso às nuvens disponíveis por meio da API REST Astra Control.
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.

Identidade e acesso

Liste os usuários

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

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

Método HTTP	Caminho
OBTER	/accounts/_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 --request GET \
--location "https://astra.netapp.io/accounts/$ACCOUNT_ID/core/v1/users" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

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

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/core/v1/users?include=firstN
ame,lastName,id" \
--include \
--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": {}
}
```

Crie um usuário

Você pode criar um usuário com credenciais específicas e uma função predefinida. Você também pode, opcionalmente, restringir o acesso do usuário a namespaces específicos.

Passo 1: Selecione um nome de usuário

Execute o fluxo de trabalho "Listar usuários" e selecione um nome disponível que não esteja atualmente em uso.

Passo 2: Crie o usuário

Execute a seguinte chamada de API REST para criar um usuário. Após a conclusão bem-sucedida da chamada, o novo usuário ainda não será utilizável.

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo

```
curl --request POST \
--location "https://astra.netapp.io/accounts/$ACCOUNT_ID/core/v1/users" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data  @JSONinput
```

Exemplo de entrada JSON

```
{
  "type" :  "application/astra-user",
  "version" : "1.1",
  "firstName" : "John",
  "lastName" : "West",
  "email" : "jwest@example.com"
}
```

Exemplo de saída JSON


```
{
    "metadata": {
        "creationTimestamp": "2022-11-20T17:23:15Z",
        "modificationTimestamp": "2022-11-20T17:23:15Z",
        "createdBy": "a20e91f3-2c49-443b-b240-615d940ec5f3",
        "labels": []
    },
    "type": "application/astra-user",
    "version": "1.2",
    "id": "d07dac0a-a328-4840-a216-12de16bbd484",
    "authProvider": "local",
    "authID": "jwest@example.com",
    "firstName": "John",
    "lastName": "West",
    "companyName": "",
    "email": "jwest@example.com",
    "postalAddress": {
        "addressCountry": "",
        "addressLocality": "",
        "addressRegion": "",
        "streetAddress1": "",
        "streetAddress2": "",
        "postalCode": ""
    },
    "state": "active",
    "sendWelcomeEmail": "false",
    "isEnabled": "true",
    "isInviteAccepted": "true",
    "enableTimestamp": "2022-11-20T17:23:15Z",
    "lastActTimestamp": ""
}
```


Passo 3: Opcionalmente, selecione os namespaces permitidos

Execute o fluxo de trabalho "Liste os namespaces" e selecione os namespaces aos quais você deseja restringir o acesso.

Passo 4: Vincule o usuário a uma função

Execute a seguinte chamada de API REST para vincular o usuário a uma função. O exemplo abaixo não coloca restrições no acesso ao namespace. Consulte "RBAC aprimorado com granularidade de namespace" para obter mais informações.

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo

```
curl --request POST \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/core/v1/roleBindings" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data  @JSONinput
```

Exemplo de entrada JSON

```
{
  "type" :  "application/astra-roleBinding",
  "version" : "1.1",
  "userID" : "d07dac0a-a328-4840-a216-12de16bbd484",
  "accountID" : "29e1f39f-2bf4-44ba-a191-5b84ef414c95",
  "role" : "viewer",
  "roleConstraints": [ "*" ]
}
```

Passo 5: Crie uma credencial

Execute a seguinte chamada de API REST para criar uma credencial e associá-la ao usuário. Este exemplo usa uma senha que é fornecida como um valor base64. A `name` propriedade deve conter o ID do usuário retornado na etapa anterior. A propriedade de entrada `change` também deve ser codificada em base64 e determina se o usuário deve alterar sua senha no primeiro login (`true` ou `false`).

> Essa etapa só é necessária com implantações do Astra Control Center que usam autenticação local. Não é necessário com implantações Astra Control Center que usam LDAP ou com implantações Astra Control Service.

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo

```
curl --request POST \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/core/v1/credentials" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data  @JSONinput
```

Exemplo de entrada JSON

```
{
  "type" :  "application/astra-credential",
  "version" : "1.1",
  "name" : "d07dac0a-a328-4840-a216-12de16bbd484",
  "keyType" : "passwordHash",
  "keyStore" : {
      "cleartext" : "TmV0QXBwMTIz",
      "change" : "ZmFsc2U="
  },
  "valid" : "true"
}
```

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.

Configuração usando API REST ou interface de usuário da Web

A versão atual do Astra Control Center é compatível com a configuração de autenticação LDAP usando a API REST Astra Control e a interface de usuário web Astra.

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 de API online" 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.

Etapa 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 e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo

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

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`.

Exemplo de saída 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": []
    }
}
```

Etapa 2: Adicione as credenciais de vinculação

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

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo

```
curl --request POST \
--location
"https://astra.example.com/accounts/$ACCOUNT_ID/core/v1/credentials" \
--include \
--header "Content-Type: application/astra-certificate+json" \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data  @JSONinput
```

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.

Exemplo de saída 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.

Etapa 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 e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo

```
curl --request GET \
--location
"https://astra.example.com/accounts/$ACCOUNT_ID/core/v1/settings?filter=na
me%20eq%20'astra.account.ldap'&include=name,id" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
```

Exemplo de saída JSON

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

Etapa 4: Atualize a configuração 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 e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

Método HTTP	Caminho
COLOQUE	/accounts/_id/core/v1/settings//setting_id

Curl exemplo

```
curl --request PUT \
--location
"https://astra.example.com/accounts/$ACCOUNT_ID/core/v1/settings/<SETTING_
ID>" \
--include \
--header "Content-Type: application/astra-setting+json" \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data @JSONinput
```

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.

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

Etapa 5: Recupere a configuração LDAP

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

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

Método HTTP	Caminho
OBTER	/accounts/_id/core/v1/settings//setting_id

Curl exemplo

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

Exemplo de saída 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.

Passo 1: Adicione um usuário

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 e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo

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

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.

Exemplo de saída 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": ""
}
```

Etapa 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 e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo

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

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.

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.

Passo 1: Adicione 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 e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo

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

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`

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": []
  }
}
```

Etapa 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 e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo

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

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.

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 e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

Método HTTP	Caminho
COLOQUE	/accounts/_id/core/v1/settings//setting_id

```
curl --request PUT \
--location
"https://astra.example.com/accounts/$ACCOUNT_ID/core/v1/settings/<SETTING_
ID>" \
--include \
--header "Content-Type: application/astra-setting+json"
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data  @JSONinput
```

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"
    }
}
```

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 e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

Método HTTP	Caminho
COLOQUE	/accounts/_id/core/v1/settings//setting_id

```
curl --request PUT \
--location
"https://astra.example.com/accounts/$ACCOUNT_ID/core/v1/settings/<SETTING_
ID>" \
--include \
--header "Content-Type: application/astra-setting+json"
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
--data  @JSONinput
```

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.
- 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.

Passo 1: Selecione a nuvem

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

Etapa 2: Listar os clusters

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

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

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

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/topology/v1/clouds/<CLOUD_ID
>/clusters" \
--include \
--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"
            }
        }
    ]
}
```

Adicione um cluster usando credenciais

Você pode adicionar um cluster para que ele fique disponível para ser gerenciado pelo Astra. A partir do lançamento do Astra 22,11, você pode adicionar um cluster com Astra Control Center e Astra Control Service.

A adição de um cluster não é necessária ao usar um serviço Kubernetes de um dos principais fornecedores de nuvem (AKS, EKS, GKE).

Passo 1: Obtenha o arquivo kubeconfig

Você precisa obter uma cópia do arquivo **kubconfig** do administrador ou serviço do Kubernetes.

Passo 2: Prepare o arquivo kubeconfig

Antes de usar o arquivo **kubeconfig**, você deve executar as seguintes operações:

1. Converter arquivo do formato YAML para JSON:

   Se você receber o arquivo kubeconfig formatado como YAML, você precisará convertê-lo para JSON.

2. Codificar JSON em base64:

   Você deve codificar o arquivo JSON em base64.

Exemplo

Aqui está um exemplo de conversão do arquivo kubeconfig de YAML para JSON e depois codificá-lo em base64:

```
yq -o=json ~/.kube/config | base64
```

Passo 3: Selecione a nuvem

Execute o fluxo de trabalho "Liste as nuvens" e selecione a nuvem em que o cluster será adicionado.

A única nuvem que você pode selecionar é a nuvem **privada**.

Passo 4: Crie uma credencial

Execute a seguinte chamada de API REST para criar uma credencial usando o arquivo kubeconfig.

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo

```
curl --request POST \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/core/v1/credentials" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data  @JSONinput
```

Exemplo de entrada JSON

```
{
  "type" :  "application/astra-credential",
  "version" : "1.1",
  "name" : "Cloud One",
  "keyType" : "kubeconfig",
  "keyStore" : {
    "base64": encoded_kubeconfig
  },
  "valid" : "true"
}
```

Passo 5: Adicione o cluster

Execute a seguinte chamada de API REST para adicionar o cluster à nuvem. O valor do `credentialID` campo de entrada é obtido a partir da chamada API REST na etapa anterior.

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

Método HTTP	Caminho
POST	/accounts//topology/v1/clouds/ /cloud_id/clusters

Curl exemplo

```
curl --request POST \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/topology/v1/clouds/<CLOUD_ID
>/clusters" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data  @JSONinput
```

Exemplo de entrada JSON

```
{
  "type" :  "application/astra-cluster",
  "version" : "1.1",
  "credentialID": credential_id
}
```

Listar clusters gerenciados

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

Execute a seguinte chamada de API REST.

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

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

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/topology/v1/managedClusters"
\
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

Gerenciar um cluster

É possível gerenciar um cluster do Kubernetes para que a proteção de dados possa ser executada.

Passo 1: Selecione o cluster a gerir

Execute o fluxo de trabalho "Listar clusters" e selecione o cluster desejado. A propriedade `managedState` do cluster deve ser `unmanaged`.

Passo 2: Opcionalmente, selecione a classe de armazenamento

Opcionalmente, execute o fluxo de trabalho "Listar classes de armazenamento" e selecione a classe de armazenamento desejada.

> Se você não fornecer uma classe de armazenamento na chamada para gerenciar o cluster, sua classe de armazenamento padrão será usada.

Etapa 3: Gerencie o cluster

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

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

Método HTTP	Caminho
POST	/Accounts/_id/topology/v1/managedclusters

Curl exemplo

```
curl --request POST \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/topology/v1/managedClusters"
\
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data  @JSONinput
```

Exemplo de entrada JSON

```
{
  "type": "application/astra-managedCluster",
  "version": "1.0",
  "id": "d0fdf455-4330-476d-bb5d-4d109714e07d"
}
```

Nuvens

Liste as nuvens

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

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

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo: Retorna todos os dados para todas as nuvens

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/topology/v1/clouds" \
--include \
--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.

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

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo: Retorna todos os dados para todos os buckets

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/topology/v1/buckets" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

Armazenamento

Listar classes de armazenamento

Você pode listar as classes de armazenamento disponíveis.

Passo 1: Selecione a nuvem

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

Passo 2: Selecione o cluster

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

Etapa 3: Listar 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 e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

Método HTTP	Caminho
OBTER	/Accounts/(account_id)/topology/v1/clouds/<CLOUD_ID>/clusters/<CLUSTER_ID>/storageClasses

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

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/topology/v1/clouds/<CLOUD_ID
>/clusters/<CLUSTER_ID>/storageClasses" \
--include \
--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.

Execute a seguinte chamada de API REST.

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

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

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/topology/v1/storageBackends"
\
--include \
--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"
    }
  ]
}
```


Habilite pools de ANF dinâmicos para clusters autogerenciados

Ao fazer backup de uma aplicação gerenciada em um cluster privado no local que tenha um back-end de storage do ANF, você precisa ativar o recurso Dynamic ANF Pools. Isso é feito fornecendo um ID de assinatura para usar ao expandir e contratar os pools de capacidade.

Os pools de ANF dinâmicos são recursos das aplicações gerenciadas pelo Astra que usam um back-end de storage do Azure NetApp Files (ANF). Ao fazer backup dessas aplicações, o Astra expande e contrai automaticamente os pools de capacidade aos quais os volumes persistentes pertencem por um fator de 1,5. Isso garante que haja espaço suficiente para o backup sem incorrer em uma carga permanente adicional. Consulte "Backups de aplicativos do Azure" para obter mais informações.

Etapa 1: Adicione o identificador de assinatura do Azure

Execute a seguinte chamada de API REST.

Você precisa atualizar o exemplo de entrada JSON conforme apropriado para o seu ambiente, incluindo o ID de assinatura e o valor base64 para o responsável pelo serviço.

Método HTTP e endpoint

Essa chamada de API REST usa o método e o endpoint a seguir.

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

Curl exemplo

```
curl --request POST \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/core/v1/credentials" \
--include \
--header "Content-Type: application/astra-credential+json"
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN" \
--data  @JSONinput
```

Exemplo de entrada JSON


```
{
  "keyStore": {
    "privKey": "SGkh",
    "pubKey": "UGhpcyCpcyBhbiBleGFtcGxlLg==",
    "base64":
"fwogICAgJmFwcElkIjogIjY4ZmSiODFiLTY0YWYtNDdjNC04ZjUzLWE2NDdlZTUzMGZkZCIsC
iAgICAiZGlzcGxheU5hbWUiOiAic3AtYXN0cmEtZGV2LXFhIiwKICAgICJuYW1lIjogImh0dHA
6Ly9zcC1hc3RyYS1kZXYtcWEiLAogICAgInBhc3N3b3JkIjogIllLQThRfk9IVVJkZWZYM0pST
WJlLnpUeFBleVE0UnNwTG9DcUJjazAiLAogICAgInRlbmFudCI6ICIwMTFjZGY2Yy03NTEyLTQ
3MDUtYjI0ZS03NzIxYWZkOGNhMzciLAogICAgInN1YnNjcmlwdGlvbklkIjogImIyMDAxNTVmL
TAwMWEtNDNiZS04N2JlLTNlZGRlODNhY2VmNCIKfQ=="
  },
  "name": "myCert",
  "type": "application/astra-credential",
  "version": "1.1",
  "metadata": {
    "labels": [
      {
        "name": "astra.netapp.io/labels/read-only/credType",
        "value": "service-account"
      },
      {
        "name": "astra.netapp.io/labels/read-only/cloudName",
        "value": "OCP"
      },
      {
        "name": "astra.netapp.io/labels/read-only/azure/subscriptionID",
        "value": "b212156f-001a-43be-87be-3edde83acef5"
      }
    ]
  }
}
```


Passo 2: Adicione um balde, se necessário

Você deve adicionar um bucket ao aplicativo gerenciado, se necessário.

Passo 3: Faça um backup do aplicativo gerenciado

Execute o fluxo de "Crie uma cópia de segurança para uma aplicação"trabalho . O pool de capacidade onde o volume persistente original está presente irá expandir e diminuir automaticamente.

Etapa 4: Revise o log de eventos

Os eventos de atividade são registados durante a cópia de segurança. Execute o fluxo de trabalho "Liste as notificações" para exibir as mensagens.

Informações sobre direitos autorais

Informações sobre marcas comerciais