LDAP配置

Astra Automation

NetApp
March 08, 2024

目录

LDAP配置

准备LDAP配置

您可以选择将Astra控制中心与轻型目录访问协议(Lightweight Directory Access Protocol、LDAP)服务器集成、以便为选定的Astra用户执行身份验证。LDAP是一种用于访问分布式目录信息的行业标准协议、也是企业身份验证的常见选择。

相关信息

- "LDAP技术规格路线图"
- "LDAP版本3"

实施过程概述

总体而言、要配置LDAP服务器以为Astra用户提供身份验证、需要执行几个步骤。

> 尽管下面介绍的步骤是按顺序执行的、但在某些情况下、您可以按不同顺序执行这些步骤。例如、您可以在配置LDAP服务器之前定义Astra用户和组。

1. 请查看 "要求和限制" 了解选项、要求和限制。
2. 选择LDAP服务器和所需的配置选项(包括安全性)。
3. 执行工作流 "将Astra配置为使用LDAP服务器" 将Astra与LDAP服务器集成。
4. 查看LDAP服务器上的用户和组、确保它们定义正确。
5. 在中执行相应的工作流 "将LDAP条目添加到Astra" 确定要使用LDAP进行身份验证的用户。

要求和限制

在将Astra配置为使用LDAP进行身份验证之前、您应查看下面介绍的Astra配置要点、包括限制和配置选项。

仅支持**Astra**控制中心

Astra Control平台提供了两种部署模式。只有Astra控制中心部署才支持LDAP身份验证。

使用**REST API**或**Web**用户界面进行配置

当前版本的Astra控制中心支持使用Astra Control REST API以及Astra Web用户界面配置LDAP身份验证。

需要**LDAP**服务器

要接受和处理Astra身份验证请求、您必须具有LDAP服务器。当前版本的Astra控制中心支持Microsoft的Active Directory。

与**LDAP**服务器的安全连接

在Astra中配置LDAP服务器时、您可以选择定义安全连接。在这种情况下、LDAPS协议需要证书。

配置用户或组

您需要选择要使用LDAP进行身份验证的用户。为此、您可以确定各个用户或一组用户。必须在LDAP服务器上

定义这些帐户。它们还需要在Astra (类型为LDAP)中进行标识、这样可以将身份验证请求转发到LDAP。

绑定用户或组时的角色限制

在当前版本的Astra控制中心中、是唯一支持的值 `roleConstraint` 为"*"。这表示用户不受限于一组有限的命名空间、并且可以访问所有命名空间。请参见 "将LDAP条目添加到Astra" 有关详细信息 …

LDAP凭据

LDAP使用的凭据包括用户名(电子邮件地址)和关联的密码。

唯一电子邮件地址

在Astra控制中心部署中用作用户名的所有电子邮件地址都必须是唯一的。您不能添加电子邮件地址已定义为Astra的LDAP用户。如果存在重复的电子邮件、您需要先从Astra中将其删除。请参见 "删除用户" 有关详细信息、请访问Astra控制中心文档站点。

也可以先定义**LDAP**用户和组

您可以将LDAP用户和组添加到Astra控制中心、即使它们尚未位于LDAP中或未配置LDAP服务器也是如此。这样、您可以在配置LDAP服务器之前预先配置用户和组。

在多个**LDAP**组中定义的用户

如果某个LDAP用户属于多个LDAP组、并且在Astra中为这些组分配了不同的角色、则用户在进行身份验证时的有效角色将获得最大的特权。例如、如果为用户分配了 `viewer` 角色与group1具有、但具有 `member` 用户在group2中的角色 `member`。这基于Astra使用的层次结构(从高到低):

- 所有者
- 管理员
- 成员
- 查看器

定期同步帐户

Astra大约每60秒就会将其用户和组与LDAP服务器同步一次。因此、如果将用户或组添加到LDAP或从LDAP中删除、则可能需要长达一分钟的时间、才能在Astra中使用该用户或组。

禁用并重置**LDAP**配置

在尝试重置LDAP配置之前、必须先禁用LDAP身份验证。此外、还可以更改LDAP服务器 (`connectionHost`)、则需要同时执行这两个操作。请参见 "禁用并重置LDAP" 有关详细信息 …

REST API参数

LDAP配置工作流会调用REST API来完成特定任务。每个API调用都可以包括输入参数、如提供的示例所示。请参见 "联机API参考" 有关如何查找参考文档的信息。

将Astra配置为使用LDAP服务器

您需要选择LDAP服务器并将Astra配置为使用该服务器作为身份验证提供程序。配置任务包括以下步骤。每个步骤都包括一个REST API调用。

1.添加**CA**证书

执行以下REST API调用、将CA证书添加到Astra。

此步骤是可选的、只有当您希望Astra和LDAP通过使用LDAPS的安全通道进行通信时、才需要执行此步骤。

HTTP 方法	路径
发布	/accouns/ {account_id} /core/v1/certificates

JSON 输入示例

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

有关输入参数、请注意以下事项：

- `cert` 是一个JSON字符串、其中包含base64编码的PKCS-11格式证书(PEM编码)。
- `isSelfSigned` 应设置为 `true` 证书是自签名的。默认值为 `false`。

curl 示例

```
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>'
```

JSON响应示例

```
{
  "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.添加绑定凭据

执行以下REST API调用以添加绑定凭据。

HTTP 方法	路径
发布	/accouns/ {account_id} /core/v1/credentials

JSON 输入示例

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

有关输入参数、请注意以下事项：

- `bindDn` 和 `password` 是LDAP管理员用户的base64编码绑定凭据、可以连接和搜索LDAP目录。 `bindDn` 是LDAP用户的电子邮件地址。

curl 示例

```
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>'
```

JSON响应示例

```
{
    "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"
    }
}
```

请注意以下响应参数：

- 。 `id` 的凭据将在后续工作流步骤中使用。

3.检索LDAP设置的UUID

执行以下REST API调用以检索的UUID `astra.account.ldap` Astra控制中心附带的设置。

> 以下cURL示例使用查询参数筛选设置收集。您可以改为删除筛选器以获取所有设置、然后搜索 `astra.account.ldap`。

HTTP 方法	路径
获取	/accouns/ {account_id} /core/v1/settings

curl 示例

```
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>'
```

JSON响应示例

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

4.更新LDAP设置

执行以下REST API调用以更新LDAP设置并完成配置。使用 `id` 上一次API调用中的值 `<SETTING_ID>` 以下URL路径中的值。

> 您可以先对特定设置的GET请求进行问题描述 处理、以查看configSchema。此操作将提供有关配置中所需字段的详细信息。

HTTP 方法	路径
PUT	/accouns/ {account_id} /core/v1/settings/ {setting_id}

JSON 输入示例

```
{
  "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"
    }
}
```

有关输入参数、请注意以下事项：

- `isEnabled` 应设置为 `true` 或者可能发生错误。
- `credentialId` 是先前创建的绑定凭据的ID。
- `secureMode` 应设置为 `LDAP` 或 `LDAPS` 根据上一步中的配置。

- 仅支持使用"Active Directory"作为供应商。

curl 示例

```
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>'
```

如果调用成功、则返回HTTP 204响应。

5.检索LDAP设置

您可以选择执行以下REST API调用来检索LDAP设置并确认更新。

HTTP 方法	路径
获取	/accouns/ {account_id} /core/v1/settings/ {setting_id}

curl 示例

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

JSON响应示例


```
{
  "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": {}
}
```

找到 `state` 字段中的值、该值将包含下表中的一个值。

State	Description
待定	配置过程仍处于活动状态、尚未完成。
valid	已成功完成配置、然后 `currentConfig` 在响应中匹配 `desiredConfig`。
error	LDAP配置过程失败。

将LDAP条目添加到Astra

将LDAP配置为Astra控制中心的身份验证提供程序后、您可以选择Astra将使用LDAP凭据进行身份验证的LDAP用户。每个用户都必须在Astra中具有一个角色、然后才能通过Astra Control REST API访问Astra。

您可以通过两种方式配置Astra来分配角色。选择适合您的环境的选项。

- "添加并绑定单个用户"
- "添加和绑定组"

LDAP凭据以用户名作为电子邮件地址以及关联的LDAP密码的形式提供。

添加并绑定单个用户

您可以为每个Astra用户分配一个角色、该角色将在LDAP身份验证后使用。如果用户数量较少、并且每个用户的管理特征可能不同、则这种做法是合适的。

1.添加用户

执行以下REST API调用、将用户添加到Astra并指示LDAP是身份验证提供程序。

HTTP 方法	路径
发布	/accouns/ {account_id} /core/v1/users

JSON 输入示例

```
{
  "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"
}
```

有关输入参数、请注意以下事项：

- 需要以下参数：
  - `authProvider`
  - `authID`
  - `email`
- `authID` 是LDAP中用户的可分辨名称(DN)
- `email` 对于在Astra中定义的所有用户、必须是唯一的

如果 `email` 值不唯一、发生错误、响应中返回409个HTTP状态代码。

curl 示例

```
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>'
```

JSON响应示例

```
{
  "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.为用户添加角色绑定

执行以下REST API调用以将用户绑定到特定角色。您需要具有上一步中创建的用户UUID。

HTTP 方法	路径
发布	/accouns/ {account_id} /core/v1/roleBindings

JSON 输入示例

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

有关输入参数、请注意以下事项：

- 上述用于的值 `roleConstraint` 是当前版本的Astra唯一可用的选项。它表示用户不受限于一组有限的命名空间、并且可以访问所有这些命名空间。

curl 示例

```
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>'
```

JSON响应示例

```
{
  "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": ["*"]
}
```

请注意以下有关响应参数的信息：

- 值 `user` 。 `principalType` 字段指示已为用户(而不是组)添加角色绑定。

添加和绑定组

您可以为Astra组分配一个角色、该角色将在LDAP身份验证后使用。如果用户数量很多、并且每个用户都可能具有类似的管理特征、则这种做法是合适的。

1.添加组

执行以下REST API调用、将组添加到Astra并指示LDAP是身份验证提供程序。

HTTP 方法	路径
发布	/accouns/ {account_id} /core/v1/groups

JSON 输入示例

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

有关输入参数、请注意以下事项：

- 需要以下参数：
  - `authProvider`
  - `authID`

curl 示例

```
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>'
```

JSON响应示例

```
{
  "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.为组添加角色绑定

执行以下REST API调用以将组绑定到特定角色。您需要具有上一步中创建的组的UUID。在LDAP执行身份验证后、属于组成员的用户将能够登录到Astra。

HTTP 方法	路径
发布	/accouns/ {account_id} /core/v1/roleBindings

JSON 输入示例

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

有关输入参数、请注意以下事项：

- 上述用于的值 `roleConstraint` 是当前版本的Astra唯一可用的选项。它表示用户不受特定命名空间的限制、并且可以访问所有命名空间。

curl 示例

```
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>'
```

JSON响应示例

```
{
  "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": ["*"]
}
```

请注意以下有关响应参数的信息：

- 值 `group` 。 `principalType` 字段指示已为组(而不是用户)添加角色绑定。

禁用并重置LDAP

您可以根据需要为Astra控制中心部署执行两项可选的相关管理任务。您可以全局禁用LDAP身份验证并重置LDAP配置。

这两个工作流任务都需要的ID `astra.account.ldap` Astra设置。有关如何检索设置ID的详细信息、请参见*配置LDAP服务器*。请参见 "检索LDAP设置的UUID" 有关详细信息 …

- "禁用LDAP身份验证"
- "重置LDAP身份验证配置"

禁用**LDAP**身份验证

您可以执行以下REST API调用来全局禁用特定Astra部署的LDAP身份验证。此调用将更新 `astra.account.ldap` 设置和 `isEnabled` 值设置为 `false`。

HTTP 方法	路径
PUT	/accouns/ {account_id} /core/v1/settings/ {setting_id}

JSON 输入示例

```
{
  "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>'
```

如果调用成功、则会显示 `HTTP 204` 返回响应。您可以选择再次检索配置设置以确认更改。

重置LDAP身份验证配置

您可以执行以下REST API调用以断开Astra与LDAP服务器的连接、并在Astra中重置LDAP配置。此调用将更新 `astra.account.ldap` 设置和值 `connectionHost` 已清除。

的值 `isEnabled` 此外、还必须设置为 `false`。您可以在进行重置调用之前设置此值、也可以在进行重置调用时设置此值。在第二种情况下、 `connectionHost` 应清除和 `isEnabled` 在同一重置调用中设置为false。

> 此操作会造成系统中断、您应谨慎操作。它会删除所有已导入的LDAP用户和组。它还会删除您在Astra控制中心中创建的所有相关Astra用户、组和roleBindings (LDAP类型)。

HTTP 方法	路径
PUT	/accouns/ {account_id} /core/v1/settings/ {setting_id}

JSON 输入示例

```
{
  "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"
    }
}
```

请注意以下事项：

- 要更改LDAP服务器、必须同时禁用和重置LDAP更改 `connectHost` 设置为空值、如上例所示。

```
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>'
```

如果调用成功、则会显示 `HTTP 204` 返回响应。您也可以选择重新检索配置以确认更改。