LDAP の設定

Astra Automation

NetApp
December 01, 2023

目次

LDAP の設定

LDAP構成を準備

必要に応じて、Astra Control CenterをLightweight Directory Access Protocol（LDAP）サーバーと統合して、選択したAstraユーザーの認証を実行できます。LDAPは、分散ディレクトリ情報にアクセスするための業界標準プロトコルであり、エンタープライズ認証に広く使用されています。

関連情報

- "LDAPテクニカル・スペシフィケーション・ロード・マップ"
- "LDAPバージョン3"

実装プロセスの概要

大まかには、Astraユーザに認証を提供するためにLDAPサーバを設定するためには、いくつかの手順を実行する必要があります。

> 以下に示す手順は順序どおりですが、場合によっては別の順序で実行できます。たとえば、LDAPサーバを設定する前に、Astraのユーザとグループを定義できます。

1. レビュー "要件および制限事項" をクリックして、オプション、要件、および制限事項を確認してください。
2. LDAPサーバおよび必要な設定オプション（セキュリティを含む）を選択します。
3. ワークフローを実行 "AstraでLDAPサーバを使用するように設定する" AstraとLDAPサーバを統合する。
4. LDAPサーバでユーザとグループを調べて、適切に定義されていることを確認します。
5. で適切なワークフローを実行します "AstraにLDAPエントリを追加" LDAPを使用して認証するユーザを指定します。

要件および制限事項

認証にLDAPを使用するようにAstraを設定する前に、制限事項や設定オプションなど、ネットアップが提供する基本的な設定を確認しておく必要があります。

Astra Control Centerでのみサポートされます

Astra Controlプラットフォームには、2つの導入モデルがあります。LDAP認証はAstra Control Centerの導入でのみサポートされます。

REST APIまたは**Web**ユーザインターフェイスを使用した設定

現在のリリースのAstra Control Centerでは、Astra Control REST APIとAstra Webユーザインターフェイスの両方を使用したLDAP認証の設定がサポートされています。

LDAPサーバが必要です

Astra認証要求を受け入れて処理するには、LDAPサーバが必要です。MicrosoftのActive Directoryは、現在のAstra Control Centerリリースでサポートされています。

LDAPサーバへのセキュアな接続

AstraでLDAPサーバを設定する場合、必要に応じてセキュアな接続を定義できます。この場合は、LDAPSプロトコルの証明書が必要です。

ユーザまたはグループを設定します

LDAPを使用して認証するユーザを選択する必要があります。これは、個々のユーザまたはユーザのグループを指定することで実行できます。アカウントはLDAPサーバで定義する必要があります。また、認証要求をLDAPに転送できるようにするために、Astra（タイプLDAP）で識別する必要もあります。

ユーザまたはグループをバインドするときのロールの制約

現在リリースされているAstra Control Centerでは、でサポートされている値はのみです `roleConstraint` は「*」です。これは、ユーザがネームスペースの制限に制限されておらず、すべてのネームスペースにアクセスできることを示しています。を参照してください "AstraにLDAPエントリを追加" を参照してください。

LDAPクレデンシャル

LDAPで使用されるクレデンシャルには、ユーザ名（Eメールアドレス）と関連付けられたパスワードが含まれます。

一意のEメールアドレス

Astra Control Center環境でユーザ名として機能するすべての電子メールアドレスは、一意である必要があります。Astraにすでに定義されているEメールアドレスを持つLDAPユーザを追加することはできません。重複するEメールが存在する場合は、最初にAstraから削除する必要があります。を参照してください "ユーザを削除します" 詳細については、Astra Control Centerのドキュメントサイトを参照してください。

必要に応じて、LDAPユーザおよびグループを先に定義します

LDAPユーザとグループは、LDAPにまだ存在していない場合やLDAPサーバが設定されていない場合でも、Astra Control Centerに追加できます。これにより、LDAPサーバを設定する前にユーザとグループを事前に設定できます。

複数のLDAPグループに定義されたユーザ

あるLDAPユーザが複数のLDAPグループに属していて、グループにAstraで別々のロールが割り当てられている場合、認証時にそのユーザの有効なロールが最も特権的に使用されます。たとえば、ユーザがに割り当てられている場合などです `viewer` group1のロールですが、にはがあります `member` group2での役割は、ユーザーの役割です `member`。これは、Astraが使用する階層に基づいています（最高から最低まで）。

- オーナー
- 管理
- メンバー
- ビューアー（ Viewer ）

定期的なアカウント同期

Astraは、ユーザーとグループを約60秒ごとにLDAPサーバーと同期します。したがって、ユーザまたはグループがLDAPに追加またはLDAPから削除されると、Astraで利用可能になるまでに最大1分かかる場合があります。

LDAP設定の無効化とリセット

LDAP設定をリセットする前に、LDAP認証を無効にする必要があります。また、LDAPサーバを変更します(`connectionHost`）をクリックし、両方の操作を実行する必要があります。を参照してください "LDAPを無効にしてリセットします" を参照してください。

REST APIパラメータ

LDAPの設定ワークフローは、特定のタスクを実行するためにREST API呼び出しを実行します。各API呼び出しについて、表示されるサンプルに示すような入力パラメータを含めることができます。を参照してください "オンラインのAPIリファレンス" 参照ドキュメントの検索方法については、を参照してください。

AstraでLDAPサーバを使用するように設定する

LDAPサーバを選択し、認証プロバイダとしてサーバを使用するようにAstraを設定する必要があります。設定タスクは、以下に説明する手順で構成されています。各手順には、単一のREST API呼び出しが含まれています。

1. CA証明書を追加します

次のREST API呼び出しを実行して、AstraにCA証明書を追加します。

この手順は省略可能で、LDAPSを使用してセキュアなチャネル経由で送信する場合にのみ必要です。

HTTP メソッド	パス
投稿（ Post ）	/accounts / {account_id} /core/v1/certificates

JSON の入力例

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

入力パラメータについては、次の点に注意してください。

- `cert` は、Base64でエンコードされたPKCS-11形式証明書（PEMでエンコードされた証明書）を含むJSON文字列です。
- `isSelfSigned` をに設定する必要があります `true` 証明書が自己署名証明書の場合。デフォルトはです `false`。

カールの例

```
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応答例

```
{
  "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.バインドクレデンシャルを追加

バインドクレデンシャルを追加するには、次のREST API呼び出しを実行します。

HTTP メソッド	パス
投稿（ Post ）	/accounts / {account_id} /core/v1/credentials

JSON の入力例

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

入力パラメータについては、次の点に注意してください。

- `bindDn` および `password` LDAPディレクトリに接続して検索できる、LDAP管理ユーザのbase64エンコードされたバインドクレデンシャルです。 `bindDn` は、LDAPユーザのEメールアドレスです。

カールの例

```
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応答例

```
{
    "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"
    }
}
```

次の応答パラメータに注意してください。

- 。 `id` のクレデンシャルは、以降のワークフローの手順で使用します。

3. LDAP設定のUUIDを取得します

次のREST API呼び出しを実行して、のUUIDを取得します `astra.account.ldap` Astra Control Centerに付属している設定。

> 次のcurlの例では、クエリパラメータを使用してsettingsコレクションをフィルタリングしています。代わりに、フィルタを削除してすべての設定を取得し、を検索できます `astra.account.ldap`。

HTTP メソッド	パス
取得	/accounts / {account_id} /core/v1/settings

カールの例

```
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応答例

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

4. LDAP設定を更新します

次のREST API呼び出しを実行してLDAP設定を更新し、設定を完了します。を使用します `id` の前回のAPI呼び出しで取得された値 `<SETTING_ID>` 次のURLパスの値。

configSchemaを最初に表示するには、特定の設定に対するGET要求を問題 に送信します。これにより、構成内の必須フィールドの詳細が表示されます。

HTTP メソッド	パス
PUT	/accounts / {account_id} /core/v1/settings/ {setting_id}

JSON の入力例

```
{
  "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"
    }
}
```

入力パラメータについては、次の点に注意してください。

- `isEnabled` をに設定する必要があります `true` エラーが発生する可能性があります。
- `credentialId` は、前の手順で作成したバインドクレデンシャルのIDです。

- `secureMode` をに設定する必要があります `LDAP` または `LDAPS` 前の手順の構成に基づいて計算します。
- ベンダーとしてサポートされているのは「Active Directory」のみです。

カールの例

```
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>'
```

呼び出しが成功すると、HTTP 204の応答が返されます。

5. LDAP設定を取得します

必要に応じて、次のREST API呼び出しを実行し、LDAP設定を取得して更新を確認することができます。

HTTP メソッド	パス
取得	/accounts / {account_id} /core/v1/settings/ {setting_id}

カールの例

```
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応答例

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

を探します `state` 次の表のいずれかの値を持つ応答内のフィールド。

状態	説明
保留中です	設定プロセスはまだアクティブで、まだ完了していません。
有効	構成は正常に完了し `currentConfig` と応答が一致しています `desiredConfig`。
エラー	LDAP設定プロセスに失敗しました。

AstraにLDAPエントリを追加

Astra Control Centerの認証プロバイダとしてLDAPを設定したら、AstraがLDAPクレデンシャルを使用して認証するLDAPユーザを選択できます。各ユーザがAstraでAstra Control REST APIを使用してAstraにアクセスするには、Astraでロールを持つ必要があります。

Astraを設定してロールを割り当てるには、2つの方法があります。環境に適したものを選択してください。

- "個々のユーザを追加してバインドします"
- "グループを追加してバインドします"

LDAPクレデンシャルは、ユーザ名の形式でEメールアドレスおよび関連付けられたLDAPパスワードです。

個々のユーザを追加してバインドします

LDAP認証後に使用する各Astraユーザにロールを割り当てることができます。これは、ユーザの数が少なく、それぞれの管理特性が異なる場合に適しています。

1.ユーザを追加します

次のREST API呼び出しを実行してAstraにユーザを追加し、LDAPが認証プロバイダであることを示します。

HTTP メソッド	パス
投稿（ Post ）	/accounts / {account_id} /core/v1/users

JSON の入力例

```
{
  "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"
}
```

入力パラメータについては、次の点に注意してください。

- 次のパラメータが必要です。
  - `authProvider`
  - `authID`
  - `email`
- `authID` は、LDAPでのユーザの識別名（DN）です
- `email` Astraで定義されているすべてのユーザに一意である必要があります

状況に応じて `email` 値が一意ではありません。エラーが発生し、応答に409 HTTPステータスコードが返されます。

カールの例

```
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応答例

```
{
  "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.ユーザのロールバインディングを追加します

次のREST API呼び出しを実行して、ユーザを特定のロールにバインドします。前の手順で作成したユーザのUUIDを用意する必要があります。

HTTP メソッド	パス
投稿（ Post ）	/accounts/{account_id}/core/v1/roleBindings

JSON の入力例

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

入力パラメータについては、次の点に注意してください。

- の上で使用された値 `roleConstraint` は、現在のリリースのAstraでのみ使用できます。ユーザがネームスペースの制限付きのセットに制限されておらず、すべてのネームスペースにアクセスできることを示しています。

カールの例

```
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応答例

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

応答パラメータについては、次の点に注意してください。

- 値 `user` をクリックします `principalType` フィールドは、グループではなくユーザにロールバインディングが追加されたことを示します。

グループを追加してバインドします

LDAP認証後に使用するAstraグループにロールを割り当てることができます。これは、ユーザが多数あり、それぞれに類似した管理特性がある場合に適しています。

1.グループを追加します

次のREST API呼び出しを実行してAstraにグループを追加し、LDAPが認証プロバイダであることを示します。

HTTP メソッド	パス
投稿（ Post ）	/accounts / {account_id} /core/v1/groups

JSON の入力例

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

入力パラメータについては、次の点に注意してください。

- 次のパラメータが必要です。
  - `authProvider`
  - `authID`

カールの例

```
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応答例

```
{
  "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.グループのロールバインドを追加します

次のREST API呼び出しを実行して、グループを特定のロールにバインドします。前の手順で作成したグループのUUIDが必要です。LDAPが認証を実行すると、グループのメンバーであるユーザはAstraにサインインできるようになります。

HTTP メソッド	パス
投稿（ Post ）	/accounts/{account_id}/core/v1/roleBindings

JSON の入力例

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

入力パラメータについては、次の点に注意してください。

- の上で使用された値 `roleConstraint` は、現在のリリースのAstraでのみ使用できます。ユーザが特定のネームスペースに制限されておらず、すべてのネームスペースにアクセスできることを示しています。

カールの例

```
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応答例

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

応答パラメータについては、次の点に注意してください。

- 値 `group` をクリックします `principalType` フィールドは、ロールバインディングが（ユーザではなく）グループに追加されたことを示します。

LDAPを無効にしてリセットします

必要に応じてAstra Control Centerの導入で実行できる、関連する管理タスクには2つのオプションがあります。LDAP認証をグローバルに無効にし、LDAP設定をリセットできます。

どちらのワークフロータスクにも、のIDが必要です `astra.account.ldap` アストラのセッティング。設定IDの取得方法の詳細については、「LDAPサーバの設定」*を参照してください。を参照してください "LDAP設定のUUIDを取得します" を参照してください。

- "LDAP認証を無効にします"
- "LDAP認証設定をリセットします"

LDAP認証を無効にします

次のREST API呼び出しを実行して、特定のアストラ環境に対してLDAP認証をグローバルに無効にすることができます。コールによってが更新されます `astra.account.ldap` の設定と `isEnabled` 値はに設定されます `false`。

HTTP メソッド	パス
PUT	/accounts / {account_id} /core/v1/settings/ {setting_id}

JSON の入力例

```
{
  "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>'
```

コールが成功した場合は、を参照してください `HTTP 204` 応答が返されます。必要に応じて、設定をもう一度取得して変更を確認することもできます。

LDAP認証設定をリセットします

次のREST API呼び出しを実行して、AstraをLDAPサーバから切断し、AstraでLDAP設定をリセットできます。コールによってが更新されます `astra.account.ldap` の設定と値 `connectionHost` がクリアされます。

の値 `isEnabled` もに設定する必要があります `false`。この値は、リセットコールの前に設定することも、リセットコールの一部として設定することもできます。2つ目のケースでは、 `connectionHost` とをクリアする必要があります `isEnabled` 同じリセットコールでfalseに設定します。

これはシステムの停止を伴う処理なので、注意してください。インポートされたLDAPユーザおよびグループがすべて削除されます。また、Astra Control Centerで作成した関連するAstraユーザー、グループ、役割バインディング(LDAPタイプ)もすべて削除されます。

HTTP メソッド	パス
PUT	/accounts / {account_id} /core/v1/settings/ {setting_id}

JSON の入力例

```
{
  "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"
    }
}
```

次の点に注意してください。

- LDAPサーバを変更するには、LDAPの変更を無効にしてリセットする必要があります `connectHost` 上の例に示すように、をNULL値に設定します。

```
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>'
```

コールが成功した場合は、を参照してください `HTTP 204` 応答が返されます。必要に応じて、設定を再取得して変更を確認することもできます。

著作権に関する情報

商標に関する情報