



インフラワークフロー Astra Automation

NetApp
August 11, 2025

This PDF was generated from https://docs.netapp.com/ja-jp/astra-automation/workflows_infra/workflows_infra_before.html on August 11, 2025. Always check docs.netapp.com for the latest.

目次

インフラワークフロー	1
インフラワークフローを使用するための準備	1
一般的な準備	1
ワークフローのカテゴリ	1
ID とアクセス	1
ユーザを一覧表示します	1
ユーザを作成します	3
LDAP の設定	7
LDAP構成を準備	7
AstraでLDAPサーバを使用するように設定する	9
AstraにLDAPエントリを追加	18
LDAPを無効にしてリセットします	25
クラスタ	28
クラスタをリストします	28
クレデンシャルを使用してクラスタを追加する	31
管理対象クラスタをリストします	33
クラスタを管理	34
クラウド	35
クラウドをリストしてください	35
バケット	36
バケットをリストします	36
ストレージ	36
ストレージクラスをリストします	36
ストレージバックエンドをリストします	39
自己管理型クラスタ向けに動的ANFプールを有効化	40

インフラワークフロー

インフラワークフローを使用するための準備

これらのワークフローを使用して、Astra Control Centerの導入で使用するインフラストラクチャを作成および管理できます。多くの場合、このワークフローはAstra Control Serviceでも使用できます。



これらのワークフローは、ネットアップがいつでも拡張および強化できるため、定期的に確認する必要があります。

一般的な準備

いずれかの Astra ワークフローを使用する前に、を確認してください ["ワークフローを使用する準備をします"](#)。

ワークフローのカテゴリ

インフラのワークフローは、さまざまなカテゴリに分類されているため、必要なワークフローを簡単に見つけることができます。

カテゴリ	説明
ID とアクセス	これらのワークフローを使用すると、Astra のアイデンティティとアクセス方法を管理できます。リソースには、ユーザ、クレデンシャル、トークンが含まれます。
LDAP の設定	必要に応じて、選択したユーザーの認証にLDAPを使用するようにAstra Control Centerを設定できます。
クラスタ	管理対象の Kubernetes クラスタを追加すると、クラスタに含まれるアプリケーションを保護およびサポートできるようになります。
クラウド	これらのワークフローは、Astra Control REST APIを使用して利用できるクラウドへのアクセスを提供します。
バケット	これらのワークフローを使用して、バックアップの格納に使用する S3 バケットを作成および管理できます。
ストレージ	これらのワークフローを使用して、ストレージバックエンドとストレージボリュームを追加および管理できます。

ID とアクセス

ユーザを一覧表示します

特定のアストラアカウントに対して定義されているユーザをリストできます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

追加の入力パラメータ

すべての REST API 呼び出しに共通するパラメータに加えて、この手順の curl の例では次のパラメータも使用されます。

パラメータ	を入力します	必須	説明
含める	クエリ	いいえ	必要に応じて、応答で返す値を選択します。

curl の例：すべてのユーザのすべてのデータを返します

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

curlの例:すべてのユーザーの名、姓、およびIDを返す

```
curl --request GET \
--location
"https://astra.netapp.io/accounts/$ACCOUNT_ID/core/v1/users?include=firstName,lastName,id" \
--include \
--header "Accept: */*" \
--header "Authorization: Bearer $API_TOKEN"
```

JSON 出力例

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

ユーザを作成します

特定のクレデンシャルと事前定義されたロールを持つユーザを作成できます。必要に応じて、ユーザのアクセスを特定のネームスペースに制限することもできます。

手順1：ユーザ名を選択する

ワークフローを実行 ["ユーザをリストします"](#) をクリックし、現在使用されていない使用可能な名前を選択します。

手順2：ユーザを作成する

ユーザを作成するには、次のREST API呼び出しを実行します。コールが正常に完了すると、新しいユーザはまだ使用できなくなります。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

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

JSON の入力例

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

JSON 出力例

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

手順3：必要に応じて、許可するネームスペースを選択します。

ワークフローを実行 ["ネームスペースを表示します"](#) アクセスを制限するネームスペースを選択します。

手順4：ユーザをロールにバインドする

次のREST API呼び出しを実行してユーザをロールにバインドします。次の例では、ネームスペースへのアクセスに制限はありません。を参照してください ["ネームスペース単位で強化された RBAC"](#) を参照してください。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

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

JSON の入力例

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

手順5：クレデンシャルを作成する

次のREST API呼び出しを実行して、クレデンシャルを作成してユーザに関連付けます。この例では、base64 値として提供されるパスワードを使用します。。name プロパティには、前の手順で返されたユーザのIDが含まれている必要があります。入力プロパティ change base64でエンコードされたユーザは、初回ログイン時にパスワードを変更する必要があるかどうかを確認する必要があります (true または false) 。



この手順は、ローカル認証を使用するAstra Control Centerの導入でのみ必要です。LDAPまたはAstra Control Serviceを使用したAstra Control Centerの導入では必要ありません。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

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

JSON の入力例

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

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とおりですが、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）で識別する必要があります。

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、「roleConstraint」でサポートされている値は「*」のみです。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で使用す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、ユーザにgroup1でビューアー・ロールが割り当てられていてgroup2でメンバー・ロールが割り当てられている場合、そのユーザのロールは'member'になります。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(connectionHost)を変更するには、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。

手順1：CA証明書を追加する

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メソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

```
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
```

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フォーマット証明書(PEMエンコード)を含むJSON文字列です。
- 証明書が自己署名されている場合は「isSelfSigned」をtrueに設定する必要がありますデフォルトは「false」です。

JSON 出力例

```
{
  "type": "application/astra-certificate",
  "version": "1.0",
  "id": "a5212e7e-402b-4cff-bba0-63f3c6505199",
  "certUse": "rootCA",
  "cert": "LS0tLS1CRUdJTlBDRVJUSUZJQ0FURS0tLS0tCk1JSUMyVEN",
  "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": []
  }
}
```

手順2：バインドクレデンシャルを追加する

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メソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

```
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
```

JSON の入力例

```
{  
  "name": "ldapBindCredential",  
  "type": "application/astra-credential",  
  "version": "1.1",  
  "keyStore": {  
    "bindDn": "dWlkPWFkbWluLG91PXM5c3RlbQ==",  
    "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」と「password」は、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で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。

JSON 出力例

```
{
  "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。

- 認証情報の「id」は、以降のワークフローの手順で使用されます。

手順3：LDAP設定のUUIDを取得する

次のREST API呼び出しを実行して'Astra Control Center'に含まれている'Astra .account.ldap'設定のUUIDを取得します



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'Astra .account.ldap'を検索することもできます

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

```
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" \
```

JSON 出力例

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

手順4：LDAP設定を更新する

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。以下のURLパスの「<Setting_ID>」の値には、前のAPI呼び出しで取得した「id」値を使用します。



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メソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

```
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
```


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は'true'に設定する必要がありますそうしないとエラーが発生する可能性があります
- 「credentialId」は、前に作成したバインドクレデンシャルのIDです。
- 「secureMode」は、前の手順で設定した内容に基づいて「ldap」または「LDAPS」に設定する必要があります。
- ベンダーとしてサポートされているのは「Active Directory」のみです。

呼び出しが成功すると、HTTP 204の応答が返されます。

手順5：LDAP設定を取得する

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メソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

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

JSON 出力例

```
{  
  "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": {}
}

```

次の表のいずれかの値を持つ応答内の'state'フィールドを探します

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ました。応答の「CurrentConfig」は「desiredConfig」と一致します。
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。

手順1：ユーザを追加する

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メソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

```
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
```

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」のように指定します
- 「authid」
- 「電子メール」
- 「authid」は、LDAPのユーザの識別名（DN）です
- 「email」は、Astraで定義されたすべてのユーザに一意である必要があります

「email」の値が一意でない場合は、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。

JSON 出力例

```
{
  "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-alb1-48f6-9729-6a571426d49f",
  "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": ""
}
```

手順2：ユーザーのロールバインドを追加する

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メソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

```
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
```

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。

- 前述の「roleConstraint」の値は、Astraの現在のリリースで使用する唯一のオプションです。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。

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-elf900312114",
  "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。

- 「principalType」フィールドの値「user」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。

手順1：グループを追加する

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メソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

```
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
```

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」のように指定します
 - 「authid」

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": []  
  }  
}
```

手順2：グループのロールバインドを追加する

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メソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

```
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
```

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。

- 前述の「roleConstraint」の値は、Astraの現在のリリースで利用できる唯一のオプションです。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。

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。

- ・「ライセンスタイプ」フィールドの値「group」は、（ユーザではなく）グループにロールバインディングが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'Astrer.account.Idap'Astra設定のIDが必要です設定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。コールは'Astra .account.Idap'設定を更新し'isEnabled'値は'false'に設定されます

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
PUT	/accounts / {account_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
```

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"
  }
}
```

コールが成功すると'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。コールは'Astra .account.ldap'の設定と'connectionHost'の値を更新します

IsEnabledの値も'false'に設定する必要がありますこの値は、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番目の場合、「connectionHost」はクリアし、「IsEnabled」は同じリセットコールでfalseに設定する必要があります。



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、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メソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
PUT	/accounts / {account_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
```

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'前述の例に示すように'LDAPを無効にしてから再設定して'connecthost'をnull値に変更する必要があります
- コールが成功すると'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。

クラスタ

クラスタをリストします

特定のクラウドで使用可能なクラスタをリストできます。

ステップ1：クラウドを選択する

ワークフローを実行 "[クラウドをリストしてください](#)" をクリックし、クラスタを含むクラウドを選択します。

手順2：クラスタをリストする

特定のクラウドにあるクラスタを表示するには、次のREST API呼び出しを実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/accounts / {account_id} /topology/v1 /雲/ {cloud_id} / clusters

curl の例：すべてのクラスタのすべてのデータを返します

```
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"
```

JSON 出力例

```
{  
  "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-vmware-infra",
    "pcloud",
    "postgresql",
    "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"
    }
  }
]
}

```

クレデンシャルを使用してクラスタを追加する

クラスタを追加して、Astraで管理できるようにすることができます。Astra 22.11リリース以降、Astra Control CenterとAstra Control Serviceの両方でクラスタを追加できるようになりました。



主要なクラウドプロバイダ（AKS、EKS、GKE）のいずれかからKubernetesサービスを使用する場合、クラスタの追加は必要ありません。

手順1：kubecfgファイルを取得する

Kubernetesの管理者またはサービスから* kubconfig *ファイルのコピーを取得する必要があります。

手順2：kubecfgファイルを準備する

*kubecfg *ファイルを使用する前に、次の操作を実行する必要があります。

1. ファイルをYAML形式からJSON形式に変換します。

YAML形式のkubecfgファイルを受信した場合は、JSONに変換する必要があります。

2. base64でJSONをエンコードします。

base64のJSONファイルをエンコードする必要があります。

例

次に、kubecfgファイルをYAMLからJSONに変換し、base64でエンコードする例を示します。

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

ステップ3：クラウドを選択する

ワークフローを実行 "[クラウドをリストしてください](#)" をクリックし、クラスタを追加するクラウドを選択します。



選択できるクラウドは*プライベート*クラウドのみです。

手順4：クレデンシャルを作成する

次のREST API呼び出しを実行して、kubeconfigファイルを使用してクレデンシャルを作成します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

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

JSON の入力例

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

手順5：クラスタを追加する

次のREST API呼び出しを実行してクラスタをクラウドに追加します。の値 `credentialID` 入力フィールドは、前の手順でREST API呼び出しから取得されます。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（Post）	/accounts / {account_id} /topology/v1 /雲/ {cloud_id} / clusters

カールの例

```
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
```

JSON の入力例

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

管理対象クラスタをリストします

Astra が現在管理している Kubernetes クラスタをリストできます。

次の REST API 呼び出しを実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/accounts / {account_id} /topology/v1 / managedClusters

curl の例：すべてのクラスタのすべてのデータを返します

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

クラスタを管理

Kubernetesクラスタを管理して、データ保護を実行できます。

手順1：管理するクラスタを選択する

ワークフローを実行 "[クラスタをリストします](#)" をクリックし、目的のクラスタを選択します。プロパティ managedState クラスタのがである必要があります unmanaged。

手順2：必要に応じて、ストレージクラスを選択します。

必要に応じてワークフローを実行します "[ストレージクラスをリストします](#)" をクリックし、目的のストレージクラスを選択します。



クラスタを管理するためのストレージクラスを指定しなかった場合は、デフォルトのストレージクラスが使用されます。

手順3：クラスタを管理する

クラスタを管理するために次のREST API呼び出しを実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
投稿（Post）	/accounts / {account_id} /topology/v1 / managedClusters

カールの例

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

JSON の入力例

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

クラウド

クラウドをリストしてください

特定のアストラアカウントで定義され、使用可能なクラウドをリストできます。

以下のREST API呼び出しを実行してクラウドを表示します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/accounts / {account_id} /topology/v1 /雲

curlの例：すべてのクラウドのすべてのデータを返します

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

バケット

バケットをリストします

特定のアストラアカウント用に定義された S3 バケットをリストできます。

次のREST API呼び出しを実行してバケットを表示します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/accounts / {account_id} /topology/v1 /バケット

curl の例：すべてのバケットのすべてのデータを返します

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

ストレージ

ストレージクラスをリストします

使用可能なストレージクラスをリストできます。

ステップ1：クラウドを選択する

ワークフローを実行 "[クラウドをリストしてください](#)" 作業するクラウドを選択します。

手順2：クラスタを選択する

ワークフローを実行 "[クラスタをリストします](#)" をクリックし、クラスタを選択します。

手順3：特定のクラスタのストレージクラスを表示する

次のREST API呼び出しを実行して、特定のクラスタとクラウドのストレージクラスを一覧表示します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/accounts / {account_id} /topology/v1 /雲/<CLOUD_ID> /<CLUSTER_ID> / storageClasses

curlの例：すべてのストレージクラスのすべてのデータを返します

```
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"
```

JSON 出力例

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

```


ストレージバックエンドをリストします

使用可能なストレージバックエンドを表示できます。

次の REST API 呼び出しを実行します。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

HTTP メソッド	パス
取得	/accounts / {account_id} /topology/v1/storageBackends

cURL の例：すべてのストレージバックエンドのすべてのデータを返します

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

```
{
  "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"
    }
  ]
}
```

自己管理型クラスタ向けに動的ANFプールを有効化

ANFストレージバックエンドを備えたプライベートオンプレミスクラスタで管理対象アプリケーションをバックアップする場合は、動的なANFプール機能を有効にする必要があります。これには、容量プールを拡張および縮小するときに使用するサブスクリプションIDを指定します。



動的ANFプールは、Azure NetApp Files (ANF) ストレージバックエンドを使用するAstraが管理するアプリケーションの機能です。これらのアプリケーションをバックアップすると、永続ボリュームが属する容量プールがAstraによって自動的に1.5倍に拡張および縮小されます。これにより、追加の恒久的な料金を支払うことなく、バックアップ用の十分なスペースが確保されます。を参照してください ["Azureアプリケーションのバックアップ"](#) を参照してください。

手順1：AzureサブスクリプションIDを追加する

次の REST API 呼び出しを実行します。



サブスクリプションIDやサービスプリンシパルのbase64値など、環境に応じてJSON入力例を更新する必要があります。

HTTPメソッドとエンドポイント

このREST API呼び出しでは、次のメソッドとエンドポイントを使用します。

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

カールの例

```
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
```

JSON の入力例

```
{
  "keyStore": {
    "privKey": "SGkh",
    "pubKey": "UGhpcyCpcyBhbiBleGFtcGxlLg==",
    "base64":
    "fwogICAgJmFwcElkIjogIjY4ZmSiODFiLTY0YWYtNDdjNC04ZjUzLWE2NDdlZTUzMGZkZCIsc
    iAgICAiZGlzcGxheU5hbWUiOiAic3AtYXN0cmEtZGV2LXFhIiwKICAgICJuYW11IjogImh0dHA
    6Ly9zcC1hc3RyYS1kZXYtcWEiLAogICAgInBhc3N3b3JkIjogIlllLQThRfk9IVVJkZWZYM0pST
    WJlLnpuUeFBleVE0UnNwTG9DcUJjazAiLAogICAgInRlbnFudCI6ICIwMTFjZGY2Yy03NTEyLTQ
    3MDUtyjIOZS03NzIxYWZkOGNhMzciLAogICAgInN1YnNjcmlwdGlvbklkIjogImIyMDAxNTVmL
    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"
      }
    ]
  }
}
```

手順2：必要に応じてバケットを追加する

必要に応じて、管理対象アプリケーションにバケットを追加する必要があります。

ステップ3：管理対象アプリのバックアップを作成する

ワークフローを実行 ["アプリケーションのバックアップを作成する"](#)。元の永続ボリュームが存在する容量プールは、自動的に拡張および縮小されます。

手順4：イベントログを確認する

アクティビティイベントはバックアップ中に記録されます。ワークフローを実行 ["通知を一覧表示します"](#) をクリックしてメッセージを表示します。

著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。