基礎架構工作流程

Astra Automation

NetApp
August 11, 2025

This PDF was generated from https://docs.netapp.com/zh-tw/astra-automation-2310/workflows_infra/workflows_infra_before.html on August 11, 2025. Always check docs.netapp.com for the latest.

目錄

基礎架構工作流程 1
開始之前 1
一般準備 1
工作流程類別 1
身分識別與存取 1
列出使用者 1
建立使用者 2
LDAP 組態 6
準備 LDAP 組態 6
設定Astra使用LDAP伺服器 8
將LDAP項目新增至Astra 15
停用並重設LDAP 22
叢集 24
列出叢集 24
使用認證新增叢集 27
列出託管叢集 29
管理叢集 29
雲端 30
列出雲端 30
桶 30
列出庫存箱 30
儲存設備 31
列出儲存類別 31
列出儲存後端 33
為自我管理的叢集啟用動態 anf 集區 34

基礎架構工作流程

開始之前

您可以使用這些工作流程來建立及維護Astra Control Center部署所使用的基礎架構。在許多情況下、工作流程也可搭配Astra Control Service使用。

NetApp可隨時擴充及強化這些工作流程、因此您應定期檢閱。

一般準備

使用任何Astra工作流程之前、請務必先檢閱 "準備好使用工作流程"。

工作流程類別

基礎架構工作流程會依不同類別進行組織、以便更容易找到您想要的工作流程。

類別	說明
身分識別與存取	這些工作流程可讓您管理身分識別、以及Astra的存取方式。這些資源包括使用者、認證和權杖。
LDAP 組態	您可以選擇性地將Astra Control Center設定為使用LDAP來驗證選取的使用者。
叢集	您可以新增受管理的Kubernetes叢集、以保護及支援其中所包含的應用程式。
雲端	這些工作流程可透過Astra Control REST API存取可用的雲端。
桶	您可以使用這些工作流程來建立及管理用來儲存備份的S3儲存區。
儲存設備	這些工作流程可讓您新增及維護儲存後端和磁碟區。

身分識別與存取

列出使用者

您可以列出針對特定Astra帳戶所定義的使用者。

1.列出使用者

執行下列REST API呼叫。

HTTP方法	路徑
取得	/Accounts / ｛account_id｝ /核心/ v1 /使用者

其他輸入參數

除了所有REST API呼叫通用的參數之外、此步驟的Curl範例也會使用下列參數。

參數	類型	必要	說明
包括	查詢	否	選擇性地選取您要傳回回應中的值。

Curl範例：傳回所有使用者的所有資料

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

Curl範例：傳回所有使用者的名字、姓氏和**ID**

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/users?include=first
Name,lastName,id' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Json輸出範例

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

建立使用者

您可以建立具有特定認證和預先定義角色的使用者。您也可以選擇性地限制使用者對特定命名空間的存取。

1.選取使用者名稱

執行工作流程 "列出使用者" 並選取目前未使用的可用名稱。

2.建立使用者

執行下列REST API呼叫以建立使用者。成功完成通話之後、新使用者將無法使用。

HTTP方法	路徑
貼文	/Accounts /｛account_id｝/核心/ v1 /使用者

JSONN輸入範例

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

Curl範例

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

Json輸出範例

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

3.選擇性選取允許的命名空間

執行工作流程 "列出命名空間" 然後選取您要限制存取的命名空間。

4.將使用者繫結至角色

執行下列REST API呼叫、將使用者繫結至角色。以下範例對命名空間存取沒有任何限制。請參閱 "增強的RBAC與命名空間精細度" 以取得更多資訊。

HTTP方法	路徑
貼文	/Accounts / ｛account_id｝ /核心/ v1/roleBindings

JSONN輸入範例

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

Curl範例

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/roleBindings'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --data
@JSONinput
```

5. 建立認證

執行下列REST API呼叫、以建立認證並將其與使用者建立關聯。此範例使用以base64值提供的密碼。。 `name` 內容應包含上一步傳回的使用者ID。輸入內容 `change` 也必須以base64編碼、並決定使用者是否必須在第一次登入時變更密碼 (`true` 或 `false`）。

此步驟僅適用於使用本機驗證的Astra Control Center部署。使用LDAP或Astra Control Service部署的Astra Control Center不需要此功能。

HTTP方法	路徑
貼文	/Accounts /｛account_id｝/核心/ v1 /認證

JSONN輸入範例

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

Curl範例

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/credentials'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --data
@JSONinput
```

LDAP 組態

準備 LDAP 組態

您可以選擇性地將Astra Control Center與輕量型目錄存取傳輸協定（LDAP）伺服器整合、以便為選取的Astra使用者執行驗證。LDAP是存取分散式目錄資訊的業界標準傳輸協定、也是企業驗證的熱門選擇。

相關資訊

- "LDAP技術規格藍圖"
- "LDAP版本3"

實作程序總覽

在高層級上、您需要執行幾個步驟來設定LDAP伺服器、以便為Astra使用者提供驗證。

> 雖然下列步驟依序顯示、但在某些情況下、您可以依照不同的順序執行這些步驟。例如、您可以先定義Astra使用者和群組、再設定LDAP伺服器。

1. 檢閱 "要求與限制" 以瞭解選項、需求及限制。
2. 選取LDAP伺服器和所需的組態選項（包括安全性）。
3. 執行工作流程 "設定Astra使用LDAP伺服器" 將Astra與LDAP伺服器整合。
4. 檢閱LDAP伺服器上的使用者和群組、確定其定義正確。
5. 在中執行適當的工作流程 "將LDAP項目新增至Astra" 識別要使用LDAP驗證的使用者。

要求與限制

您應該先檢閱下列Astra組態基本要點、包括限制和組態選項、再將Astra設定為使用LDAP進行驗證。

僅**Astra**控制中心支援

Astra Control平台提供兩種部署模式。LDAP驗證僅支援Astra Control Center部署。

使用**REST API**或**Web**使用者介面進行組態設定

目前版本的Astra Control Center支援使用Astra Control REST API和Astra網路使用者介面來組態LDAP驗證。

需要**LDAP**伺服器

您必須擁有LDAP伺服器、才能接受及處理Astra驗證要求。Microsoft的Active Directory受目前Astra Control

Center版本支援。

安全連線至**LDAP**伺服器

在Astra中設定LDAP伺服器時、您可以選擇性地定義安全連線。在此情況下、LDAPS傳輸協定需要憑證。

設定使用者或群組

您需要選取要使用LDAP驗證的使用者。您可以透過識別個別使用者或使用者群組來執行此作業。帳戶必須在LDAP伺服器上定義。也需要在Astra（類型LDAP）中識別這些驗證要求、以便將驗證要求轉送到LDAP。

連結使用者或群組時的角色限制

目前推出的Astra Control Center是唯一支援的值 `roleConstraint` 為「*」。這表示使用者不受限於一組有限的命名空間、而且可以存取所有命名空間。請參閱 "將LDAP項目新增至Astra" 以取得更多資訊。

LDAP認證

LDAP使用的認證資料包括使用者名稱（電子郵件地址）和相關密碼。

獨特的電子郵件地址

Astra Control Center部署中所有以使用者名稱身分使用的電子郵件地址都必須是唯一的。您無法使用已定義為Astra的電子郵件地址新增LDAP使用者。如果存在重複的電子郵件、您必須先從Astra刪除。請參閱 "移除使用者" 如需詳細資訊、請參閱Astra Control Center文件網站。

（可選）先定義**LDAP**使用者和群組

您可以將LDAP使用者和群組新增至Astra Control Center、即使LDAP中尚未存在或LDAP伺服器尚未設定亦然。這可讓您在設定LDAP伺服器之前預先設定使用者和群組。

在多個**LDAP**群組中定義的使用者

如果LDAP使用者屬於多個LDAP群組、且已在Astra中指派不同的角色、則使用者在驗證時的有效角色將是最具權限的角色。例如、如果指派給使用者 `viewer` 角色與群組1、但具有 `member` 角色在群組2中、使用者的角色是 `member`。這是根據Astra（最高至最低）所使用的階層架構而定：

- 擁有者
- 管理
- 成員
- 檢視者

定期帳戶同步

Astra大約每60秒將IT的使用者和群組與LDAP伺服器同步一次。因此、如果將使用者或群組新增至LDAP或從LDAP移除、可能需要一分鐘的時間才能在Astra中使用。

停用及重設**LDAP**組態

在嘗試重設LDAP組態之前、您必須先停用LDAP驗證。此外、也可變更LDAP伺服器 (`connectionHost`）、您必須同時執行這兩項作業。請參閱 "停用並重設LDAP" 以取得更多資訊。

REST API參數

LDAP組態工作流程會呼叫REST API來完成特定工作。每個API呼叫都可以包含輸入參數、如所提供的範例所示。請參閱 "線上API參考" 以取得如何找到參考文件的相關資訊。

設定Astra使用LDAP伺服器

您需要選取LDAP伺服器、並設定Astra以使用伺服器做為驗證供應商。組態工作包含下列步驟。每個步驟都包含單一REST API呼叫。

1.新增CA憑證

執行下列REST API呼叫、將CA憑證新增至Astra。

此步驟為選用步驟、只有當您希望Astra和LDAP透過使用LDAPS的安全通道進行通訊時才需要執行。

HTTP方法	路徑
貼文	/Accounts / {account_id} /核心/ v1/certificates

JSONN輸入範例

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

請注意下列關於輸入參數的資訊：

- `cert` 是Json字串、其中包含一個已編碼的64位元的PKCS-11格式化憑證（PES編碼）。
- `isSelfSigned` 應設為 `true` 如果憑證是自我簽署的。預設值為 `false`。

Curl範例

```
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>'
```

JSONN回應範例

```
{
  "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.新增連結認證資料

執行下列REST API呼叫以新增繫結認證。

HTTP方法	路徑
貼文	/Accounts / {account_id} /核心/ v1 /認證

JSONN輸入範例

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

請注意下列關於輸入參數的資訊：

- `bindDn` 和 `password` 是LDAP管理使用者的基礎64編碼繫結認證、可連線及搜尋LDAP目錄。 `bindDn` 為LDAP使用者的電子郵件地址。

Curl範例

```
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>'
```

JSONN回應範例

```
{
    "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"
    }
}
```

請注意下列回應參數：

- 。 `id` 的認證資料會用於後續的工作流程步驟。

3.擷取LDAP設定的UUID

執行下列REST API呼叫、以擷取的UUID `astra.account.ldap` Astra Control Center隨附的設定。

> 下列Curl範例使用查詢參數來篩選設定集合。您可以移除篩選條件、以取得所有設定、然後搜尋 `astra.account.ldap`。

HTTP方法	路徑
取得	/Accounts / {account_id} /核心/ v1/settings

Curl範例

```
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>'
```

JSONN回應範例

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

4.更新LDAP設定

執行下列REST API呼叫、以更新LDAP設定並完成組態。使用 `id` 先前API呼叫的值 `<SETTING_ID>` URL路徑中的值。

> 您可以先發出特定設定的Get要求、以查看configSchema。這將提供組態中必要欄位的詳細資訊。

HTTP方法	路徑
放入	/Accounts / {account_id} /核心/ v1/settings / {setting_id}

JSONN輸入範例

```
{
  "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"
    }
}
```

請注意下列關於輸入參數的資訊：

- `isEnabled` 應設為 `true` 或可能發生錯誤。
- `credentialId` 是先前建立的連結認證資料ID。
- `secureMode` 應設為 `LDAP` 或 `LDAPS` 根據您在先前步驟中的組態。
- 廠商僅支援「Active Directory」。

Curl範例

```
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>'
```

如果通話成功、則會傳回HTTP 204回應。

5.擷取LDAP設定

您可以選擇性地執行下列REST API呼叫、以擷取LDAP設定並確認更新。

HTTP方法	路徑
取得	/Accounts / {account_id} /核心/ v1/settings / {setting_id}

Curl範例

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

JSONN回應範例


```
{
  "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` 回應中的欄位、其值如下表所示。

州/省	說明
擱置中	組態程序仍在作用中、尚未完成。
有效	組態已成功完成且 `currentConfig` 在回應中相符 `desiredConfig`。
錯誤	LDAP組態程序失敗。

將**LDAP**項目新增至**Astra**

將LDAP設定為Astra Control Center的驗證提供者之後、您可以選取Astra將使用LDAP認證進行驗證的LDAP使用者。每位使用者必須在Astra中扮演角色、才能透過Astra Control REST API存取Astra。

您可以使用兩種方式設定Astra來指派角色。選擇適合您環境的產品。

- "新增及連結個別使用者"
- "新增及繫結群組"

LDAP認證資料的形式為使用者名稱、電子郵件地址及相關的LDAP密碼。

新增及連結個別使用者

您可以指派角色給LDAP驗證後所使用的每個Astra使用者。當使用者人數不多、而且每個使用者可能具有不同的管理特性時、這是適當的做法。

1.新增使用者

執行下列REST API呼叫、將使用者新增至Astra、並指出LDAP為驗證提供者。

HTTP方法	路徑
貼文	/Accounts / ｛account_id｝ /核心/ v1 /使用者

JSONN輸入範例

```
{
  "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"
}
```

請注意下列關於輸入參數的資訊：

- 需要下列參數：
  - `authProvider`
  - `authID`
  - `email`
- `authID` 是LDAP中使用者的辨別名稱（DN）
- `email` 對於Astra中定義的所有使用者而言、必須是唯一的

如果是 `email` 值並非唯一、會發生錯誤、並傳回回應中的409 HTTP狀態代碼。

Curl範例

```
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>'
```

JSONN回應範例

```
{
  "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.新增使用者的角色繫結

執行下列REST API呼叫、將使用者繫結至特定角色。您必須擁有上一步建立的使用者UUID。

HTTP方法	路徑
貼文	/Accounts / {account_id} /核心/ v1/roleBindings

JSONN輸入範例

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

請注意下列關於輸入參數的資訊：

- 上述使用的值 `roleConstraint` 是目前Astra版本唯一可用的選項。這表示使用者不受限於一組有限的命名空間、而且可以全部存取。

Curl範例

```
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>'
```

JSONN回應範例

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

請注意下列關於回應參數的資訊：

- 價值 `user` 適用於 `principalType` 欄位表示已為使用者（而非群組）新增角色繫結。

新增及繫結群組

您可以將角色指派給Astra群組、此群組在LDAP驗證之後使用。當使用者數量眾多、而且每個使用者可能具有類似的管理特性時、這是適當的做法。

1.新增群組

執行下列REST API呼叫、將群組新增至Astra、並指出LDAP為驗證提供者。

HTTP方法	路徑
貼文	/Accounts / {account_id} /核心/ v1/Groups

JSONN輸入範例

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

請注意下列關於輸入參數的資訊：

- 需要下列參數：
  - authProvider
  - authID

Curl範例

```
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>'
```

JSONN回應範例

```
{
  "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.新增群組的角色繫結

執行下列REST API呼叫、將群組繫結至特定角色。您必須擁有上一步建立的群組UUID。在LDAP執行驗證之後、屬於群組成員的使用者將能夠登入Astra。

HTTP方法	路徑
貼文	/Accounts / {account_id} /核心/ v1/roleBindings

JSONN輸入範例

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

請注意下列關於輸入參數的資訊：

- 上述使用的值 `roleConstraint` 是目前Astra版本唯一可用的選項。這表示使用者不受限於特定命名空間、而且可以全部存取。

Curl範例

```
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>'
```

JSONN回應範例

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

請注意下列關於回應參數的資訊：

- 價值 `group` 適用於 `principalType` 欄位表示已新增群組的角色繫結（非使用者）。

停用並重設LDAP

您可以視需要執行兩項選用的相關管理工作、以進行Astra Control Center部署。您可以全域停用LDAP驗證並重設LDAP組態。

這兩個工作流程工作都需要的ID `astra.account.ldap` Astra環境。有關如何擷取設定ID的詳細資訊、請參閱*設定LDAP伺服器*。請參閱 "擷取LDAP設定的UUID" 以取得更多資訊。

- "停用 LDAP 驗證"
- "重設LDAP驗證組態"

停用 LDAP 驗證

您可以執行下列REST API呼叫、以全域停用特定Astra部署的LDAP驗證。通話會更新 `astra.account.ldap` 設定和 `isEnabled` 值設為 `false`。

HTTP方法	路徑
放入	/Accounts / {account_id} /核心/ v1/settings / {setting_id}

JSONN輸入範例

```
{
  "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>'
```

如果通話成功、則為 `HTTP 204` 傳回回應。您可以選擇再次擷取組態設定以確認變更。

重設LDAP驗證組態

您可以執行下列REST API呼叫、以中斷Astra與LDAP伺服器的連線、並在Astra中重設LDAP組態。通話會更新 `astra.account.ldap` 設定和值 `connectionHost` 已清除。

的價值 `isEnabled` 也必須設定為 `false`。您可以在進行重設通話之前或在進行重設通話時設定此值。在第二種情況下、 `connectionHost` 應清除及 `isEnabled` 在相同的重設通話中設為假。

> 這是一項顛覆性的作業、您應該謹慎進行。它會刪除所有匯入的LDAP使用者和群組。它也會刪除您在Astra Control Center中建立的所有相關Astra使用者、群組和角色繫結（LDAP類型）。

HTTP方法	路徑
放入	/Accounts / {account_id} /核心/ v1/settings / {setting_id}

JSONN輸入範例

```
{
  "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"
    }
}
```

請注意下列事項：

- 若要變更LDAP伺服器、您必須停用並重設LDAP變更 `connectHost` 至null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>'
```

如果通話成功、則為 `HTTP 204` 傳回回應。您可以選擇再次擷取組態以確認變更。

叢集

列出叢集

您可以在特定雲端中列出可用的叢集。

1.選取雲端

執行工作流程 "列出雲端" 然後選取包含叢集的雲端。

2. 列出叢集

執行下列REST API呼叫、以列出特定雲端中的叢集。

HTTP方法	路徑
取得	/Accounts / {account_id} /拓撲/ v1/v雲端/ {雲端_id} /叢集

Curl範例：傳回所有叢集的所有資料

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

Json輸出範例


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

使用認證新增叢集

您可以新增叢集、以便由Astra管理。從Astra 22.11版開始、您可以使用Astra Control Center和Astra Control Service來新增叢集。

使用其中一家主要雲端供應商（包括：KS、EKS、GKE）提供的Kubernetes服務時、不需要新增叢集。

1.取得Kbeconfig檔案

您需要向Kubernetes系統管理員或服務取得* kubconfig*檔案的複本。

2.準備Kbeconfig檔案

在使用* kubeconfig *檔案之前、您應該先執行下列作業：

將檔案從Yaml格式轉換為Json

如果您收到格式為Yaml的Kbeconfig檔案、則必須將其轉換為Json。

在base64中編碼Json

您必須在base64中編碼Json檔案。

範例

以下是將 Kebeconfig 檔案從 YAML 轉換為 JSON 、然後在 base64 中編碼的範例：

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

3.選擇雲端

執行工作流程 "列出雲端" 然後選取要新增叢集的雲端。

您唯一可以選擇的雲端是*私有*雲端。

4.建立認證資料

執行下列REST API呼叫、以使用Kbeconfig檔案建立認證。

HTTP方法	路徑
貼文	/Accounts / ｛account_id｝/核心/ v1 /認證

JSONN輸入範例

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

Curl範例

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/credentials'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --data
@JSONinput
```

5.新增叢集

執行下列REST API呼叫、將叢集新增至雲端。的值 `credentialID` 輸入欄位可從上一步的REST API呼叫取得。

HTTP方法	路徑
貼文	/Accounts / ｛account_id｝/拓撲/ v1/v雲端/ ｛雲端_id｝/叢集

JSONN輸入範例

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

Curl範例

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

列出託管叢集

您可以列出目前由Astra管理的Kubernetes叢集。

1.列出託管叢集

執行下列REST API呼叫。

HTTP方法	路徑
取得	/Accounts / {account_id} /拓撲/ v1/managedClusters

Curl範例：傳回所有叢集的所有資料

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

管理叢集

您可以管理Kubernetes叢集、以便執行資料保護。

1.選取要管理的叢集

執行工作流程 "列出叢集" 然後選取所需的叢集。屬性 `managedState` 叢集必須是 `unmanaged`。

2.（可選）選擇儲存類別

（可選）執行工作流程 "列出儲存類別" 並選擇所需的儲存類別。

如果您未在通話中提供儲存類別來管理叢集、將會使用您的預設儲存類別。

3.管理叢集

執行下列REST API呼叫來管理叢集。

HTTP方法	路徑
貼文	/Accounts / {account_id} /拓撲/ v1/managedClusters

JSONN輸入範例

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

Curl範例

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

雲端

列出雲端

您可以列出已定義的雲端、以及可用的特定Astra帳戶。

1.列出雲端

執行下列REST API呼叫、以列出雲端。

HTTP方法	路徑
取得	/Accounts / ｛account_id｝/拓撲/ v1/v雲端

Curl範例：傳回所有雲端的所有資料

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

桶

列出庫存箱

您可以列出針對特定Astra帳戶所定義的S3儲存區。

1.列出庫存箱

執行下列REST API呼叫、以列出儲存區。

HTTP方法	路徑
取得	/Accounts / ｛account_id｝/拓撲/ v1/buckets

Curl範例：傳回所有儲存區的所有資料

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

儲存設備

列出儲存類別

您可以列出可用的儲存類別。

1.選取雲端

執行工作流程 "列出雲端" 並選擇您要使用的雲端。

2.選取叢集

執行工作流程 "列出叢集" 然後選取叢集。

3.列出特定叢集的儲存類別

執行下列REST API呼叫、列出特定叢集和雲端的儲存類別。

HTTP方法	路徑
取得	/accounts/{account_id}/topology / v1/ves/setf/setles/wes/sets/wese/storageClasses <CLUSTER_ID> <CLOUD_ID>

Curl範例：傳回所有儲存類別的所有資料

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

Json輸出範例

```

```

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


列出儲存後端

您可以列出可用的儲存後端。

1.列出後端

執行下列REST API呼叫。

HTTP方法	路徑
取得	/accounts/{account_id}/topology / v1/storageBackends

Curl範例：傳回所有儲存後端的所有資料

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

Json輸出範例


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


為自我管理的叢集啟用動態 **anf** 集區

在具有 ANF 儲存後端的私有內部部署叢集中備份受管理應用程式時、您必須啟用動態 ANF 集區功能。這是透過提供訂閱 ID 來完成、以便在擴充和承包容量資源池時使用。

動態 anf 資源池是 Astra 託管應用程式的功能、使用 Azure NetApp Files （ anf ）儲存後端。在備份這些應用程式時、 Astra 會自動擴充並收縮持續磁碟區所屬的容量集區 1.5 倍。如此可確保有足夠的空間進行備份、而不會產生額外的永久性費用。請參閱 "Azure 應用程式備份" 以取得更多資訊。

1. 新增 **Azure** 訂閱識別碼

執行下列REST API呼叫。

您需要視環境而更新 JSON 輸入範例、包括服務主體的訂閱 ID 和 base64 值。

HTTP方法	路徑
貼文	/Accounts / {account_id} /核心/ v1 /認證

JSONN輸入範例

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

Curl範例

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

2. 視需要新增貯體

如有需要、您應將儲存庫新增至託管應用程式。

3. 備份託管應用程式

執行工作流程 "為應用程式建立備份"。原始持續磁碟區所在的容量集區將會自動擴充和縮小。

4. 檢閱事件記錄

活動事件會在備份期間記錄。執行工作流程 "列出通知" 以檢視訊息。

版權資訊

Copyright © 2025 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。 美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 http://www.netapp.com/TM 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。