Documentazione di Astra Control Automation 22.08

Astra Automation 22.08

NetApp
December 04, 2023

Sommario

Documentazione di Astra Control Automation 22.08

Note di rilascio

A proposito di questa release

La documentazione disponibile in questo sito descrive l'API REST di Astra Control e le relative tecnologie di automazione disponibili con la release di agosto 2022 (22.08) di Astra Control. In particolare, questa release dell'API REST è inclusa con le corrispondenti release 22.08 di Astra Control Center e Astra Control Service.

Per ulteriori informazioni su questa release e sulle release precedenti, consultare le pagine e i siti seguenti:

- "Novità dell'API REST di Astra Control"
- "Risorse REST ed endpoint"
- "Documentazione di Astra Control Center 22.08"
- "Documentazione del servizio Astra Control"
- "Versioni precedenti della documentazione di Astra Automation"

Seguici su Twitter @NetAppDoc. Invia un feedback sulla documentazione diventando un "Collaboratore di GitHub" oppure inviare un'e-mail all'indirizzo doccomments@netapp.com.

Novità dell'API REST di Astra Control

NetApp aggiorna regolarmente l'API REST di Astra Control per offrire nuove funzionalità, miglioramenti e correzioni di bug.

10 agosto 2022 (22.08)

Questa release include un'espansione e un aggiornamento dell'API REST, oltre a funzionalità di sicurezza e amministrazione avanzate.

Risorse Astra nuove e migliorate

Sono stati aggiunti tre nuovi tipi di risorse: **Certificato**, **Gruppo** e **AppMirror**. Inoltre, sono state aggiornate le versioni di diverse risorse esistenti.

Autenticazione LDAP

È possibile configurare Astra Control Center in modo che si integri con un server LDAP per autenticare gli utenti Astra selezionati. Vedere "Configurazione LDAP" per ulteriori informazioni.

Gancio di esecuzione migliorato

Il supporto per gli hook di esecuzione è stato aggiunto alla release Astra Control 21.12. Oltre agli hook di esecuzione pre-snapshot e post-snapshot esistenti, è ora possibile configurare i seguenti tipi di hook di esecuzione con la versione 22.08:

- Pre-backup
- Post-backup

- Post-ripristino

Astra Control ora consente anche di utilizzare lo stesso script per più hook di esecuzione.

Replica dell'applicazione con SnapMirror

È ora possibile replicare le modifiche di dati e applicazioni tra cluster utilizzando la tecnologia NetApp SnapMirror. Questo miglioramento può essere utilizzato per migliorare la business continuity e le funzionalità di recovery.

Informazioni correlate

- "Astra Control Center: Novità"
- "Astra Control Service: Novità"

26 aprile 2022 (22.04)

Questa release include un'espansione e un aggiornamento dell'API REST, oltre a funzionalità di sicurezza e amministrazione avanzate.

Risorse Astra nuove e migliorate

Sono stati aggiunti due nuovi tipi di risorse: **Pacchetto** e **aggiornamento**. Inoltre, le versioni di diverse risorse esistenti sono state aggiornate.

RBAC migliorato con granularità dello spazio dei nomi

Quando si associa un ruolo a un utente associato, è possibile limitare gli spazi dei nomi a cui l'utente ha accesso. Vedere il riferimento **role binding API** e. "Sicurezza RBAC" per ulteriori informazioni.

Rimozione della benna

È possibile rimuovere un bucket quando non è più necessario o non funziona correttamente.

Supporto per Cloud Volumes ONTAP

Cloud Volumes ONTAP è ora supportato come back-end di storage.

Ulteriori miglioramenti del prodotto

Sono disponibili diversi miglioramenti aggiuntivi alle due implementazioni dei prodotti Astra Control, tra cui:

- Ingresso generico per Astra Control Center
- Cluster privato in AKS
- Supporto per Kubernetes 1.22
- Supporto per il portfolio VMware Tanzu

Consulta la pagina **Novità** nei siti di documentazione di Astra Control Center e Astra Control Service.

Informazioni correlate

- "Astra Control Center: Novità"
- "Astra Control Service: Novità"

14 dicembre 2021 (21.12)

Questa release include un'espansione dell'API REST insieme a una modifica alla struttura della documentazione per supportare meglio l'evoluzione di Astra Control attraverso i futuri aggiornamenti delle release.

Documentazione di Astra Automation separata per ogni release di Astra Control

Ogni release di Astra Control include un'API REST distinta che è stata migliorata e adattata alle funzionalità della release specifica. La documentazione per ciascuna release dell'API REST di Astra Control è ora disponibile sul proprio sito Web dedicato insieme al repository di contenuti GitHub associato. Il principale sito di documentazione "Automazione del controllo Astra" contiene sempre la documentazione relativa alla versione più recente. Vedere "Versioni precedenti della documentazione di Astra Control Automation" per informazioni sulle release precedenti.

Espansione dei tipi DI risorse RIMANENTI

Il numero di tipi di risorse REST ha continuato a espandersi con l'enfasi sugli hook di esecuzione e sui backend dello storage. Le nuove risorse includono: Account, gancio di esecuzione, origine hook, override hook di esecuzione, nodo cluster, backend di storage gestito, namespace, dispositivo di storage e nodo di storage. Vedere "Risorse" per ulteriori informazioni.

SDK NetApp Astra Control Python

NetApp Astra Control Python SDK è un pacchetto open source che semplifica lo sviluppo di codice di automazione per il tuo ambiente Astra Control. Il fulcro è l'SDK Astra, che include un insieme di classi per astrarre la complessità delle chiamate API REST. È inoltre disponibile uno script toolkit per eseguire task amministrativi specifici eseguendo il wrapping e l'astrazione delle classi Python. Vedere "SDK NetApp Astra Control Python" per ulteriori informazioni.

5 agosto 2021 (21.08)

Questa release include l'introduzione di un nuovo modello di implementazione Astra e un'importante espansione dell'API REST.

Modello di implementazione di Astra Control Center

Oltre all'offerta di Astra Control Service esistente come servizio di cloud pubblico, questa release include anche il modello di implementazione on-premise di Astra Control Center. Puoi installare Astra Control Center presso la tua sede per gestire il tuo ambiente Kubernetes locale. I due modelli di implementazione di Astra Control condividono la stessa API REST, con piccole differenze indicate nella documentazione.

Espansione dei tipi DI risorse RIMANENTI

Il numero di risorse accessibili tramite l'API REST di Astra Control si è notevolmente ampliato, con molte delle nuove risorse che forniscono una base per l'offerta on-premise di Astra Control Center. Le nuove risorse includono: ASUP, diritto, funzionalità, licenza, impostazione, sottoscrizione, bucket, cloud, cluster, cluster gestito, back-end dello storage e classe di storage. Vedere "Risorse" per ulteriori informazioni.

Endpoint aggiuntivi che supportano un'implementazione Astra

Oltre alle risorse REST estese, sono disponibili diversi altri nuovi endpoint API per supportare un'implementazione di Astra Control.

Supporto di OpenAPI

Gli endpoint OpenAPI forniscono l'accesso al documento JSON OpenAPI corrente e ad altre risorse correlate.

Supporto di OpenMetrics

Gli endpoint OpenMetrics forniscono l'accesso alle metriche degli account attraverso la risorsa OpenMetrics.

15 aprile 2021 (21.04)

Questa versione include le seguenti nuove funzioni e miglioramenti.

Introduzione dell'API REST

L'API REST di Astra Control è disponibile per l'utilizzo con l'offerta di Astra Control Service. È stato creato in base alle tecnologie REST e alle Best practice attuali. L'API fornisce le basi per l'automazione delle implementazioni Astra e include le seguenti funzionalità e vantaggi.

Risorse

Sono disponibili quattordici tipi di risorse REST.

Accesso al token API

L'accesso all'API REST viene fornito tramite un token di accesso API che è possibile generare nell'interfaccia utente web Astra. Il token API fornisce un accesso sicuro all'API.

Supporto per le raccolte

Esiste un insieme completo di parametri di query che possono essere utilizzati per accedere alle raccolte di risorse. Alcune delle operazioni supportate includono il filtraggio, l'ordinamento e l'impaginazione.

Problemi noti

Si consiglia di esaminare tutti i problemi noti relativi alla release corrente relativi all'API REST di Astra Control. I problemi noti identificano i problemi che potrebbero impedire il corretto utilizzo del prodotto.

> Non ci sono nuovi problemi noti con la release 22.08 dell'API REST di Astra Control. I problemi descritti di seguito sono stati rilevati nelle release precedenti e sono ancora applicabili alla release corrente.

Non vengono rilevati tutti i dispositivi di storage in un nodo di storage back-end

Quando si effettua una chiamata API REST per recuperare i dispositivi di storage definiti in un nodo di storage, vengono rilevati solo i dispositivi Astra Data Store. Non tutti i dispositivi vengono restituiti.

Panoramica delle caratteristiche e dei vantaggi

Astra Control Center e Astra Control Service forniscono un'API REST comune a cui è possibile accedere direttamente attraverso un linguaggio di programmazione o un'utility come Curl. Di seguito vengono presentati i principali punti di forza e i vantaggi dell'API.

> Per accedere all'API REST, devi prima accedere all'interfaccia utente web Astra e generare un token API. È necessario includere il token in ogni richiesta API.

Basato sulla tecnologia REST

L'API Astra Control è stata creata utilizzando la tecnologia REST e le Best practice attuali. La tecnologia di base include HTTP, JSON e RBAC.

Supporto per i due modelli di implementazione di Astra Control

Astra Control Service viene utilizzato nell'ambiente di cloud pubblico, mentre Astra Control Center è per le implementazioni on-premise. Esiste un'API REST che supporta entrambi questi modelli di implementazione.

Mappatura chiara tra le risorse degli endpoint REST e il modello a oggetti

Gli endpoint REST esterni utilizzati per accedere alle risorse vengono mappati su un modello a oggetti coerente gestito internamente dal servizio Astra. Il modello a oggetti è progettato utilizzando la modellazione delle relazioni con le entità (ER) che aiuta a definire chiaramente le azioni e le risposte API.

Set completo di parametri di query

L'API REST fornisce un insieme completo di parametri di query che è possibile utilizzare per accedere alle raccolte di risorse. Alcune delle operazioni supportate includono il filtraggio, l'ordinamento e l'impaginazione.

Allineamento con l'interfaccia utente Web di Astra Control

Il design dell'interfaccia utente web Astra è allineato con L'API REST e quindi c'è coerenza tra i due percorsi di accesso e l'esperienza dell'utente.

Solidi dati di debug e determinazione dei problemi

L'API REST di Astra Control offre un'efficace funzionalità di debug e determinazione dei problemi, inclusi eventi di sistema e notifiche degli utenti.

Processi di workflow

Viene fornita una serie di flussi di lavoro per agevolare lo sviluppo del codice di automazione. I flussi di lavoro sono organizzati in due categorie principali: Infrastruttura e gestione.

Base per tecnologie di automazione avanzate

Oltre ad accedere direttamente all'API REST, è possibile utilizzare altre tecnologie di automazione basate sull'API REST.

Parte della documentazione della famiglia Astra

La documentazione di Astra Control Automation fa parte della più ampia documentazione della famiglia Astra. Vedere "Documentazione Astra" per ulteriori informazioni.

Inizia subito

Prima di iniziare

Puoi prepararti rapidamente a iniziare a utilizzare l'API REST di Astra Control seguendo i passaggi riportati di seguito.

Disporre delle credenziali dell'account Astra

Per accedere all'interfaccia utente web Astra e generare un token API, sono necessarie le credenziali Astra. Con Astra Control Center, queste credenziali vengono gestite in locale. Con Astra Control Service è possibile accedere alle credenziali dell'account tramite il servizio **Auth0**.

Acquisire familiarità con i concetti di base di Kubernetes

Dovresti avere familiarità con diversi concetti di base di Kubernetes. Vedere "Concetti di base di Kubernetes" per ulteriori informazioni.

Esaminare i concetti DI REST e l'implementazione

Assicurarsi di rivedere "Implementazione core REST" Per informazioni sui concetti REST e sui dettagli relativi alla progettazione dell'API REST di Astra Control.

Ulteriori informazioni

È necessario conoscere le risorse informative aggiuntive come suggerito in "Risorse aggiuntive".

Ottieni un token API

Per utilizzare l'API REST di Astra Control è necessario ottenere un token API Astra.

Introduzione

Un token API identifica il chiamante di Astra e deve essere incluso in ogni chiamata API REST.

- È possibile generare un token API utilizzando l'interfaccia utente web Astra.
- L'identità dell'utente trasportata con il token è determinata dall'utente che lo crea.
- Il token deve essere incluso in `Authorization` Intestazione della richiesta HTTP.
- Un token non scade mai dopo la sua creazione.
- È possibile revocare un token nell'interfaccia utente web Astra.

Informazioni correlate

- "Revocare un token API"

Creare un token API Astra

La seguente procedura descrive come creare un token API Astra.

Prima di iniziare

Hai bisogno di credenziali per un account Astra.

A proposito di questa attività

Questa attività genera un token API nell'interfaccia web Astra. È inoltre necessario recuperare l'ID dell'account necessario anche per effettuare chiamate API.

Fasi

1. Accedi ad Astra utilizzando le credenziali del tuo account.

   Accedere al seguente sito per Astra Control Service: "https://astra.netapp.io"

2. Fare clic sull'icona a forma di figura nella parte superiore destra della pagina e selezionare **API access**.
3. Fare clic su **generate API token** nella pagina e nella finestra popup fare clic su **generate API token**.
4. Fare clic sull'icona per copiare la stringa del token negli Appunti e salvarla nell'editor.
5. Copiare e salvare l'id account disponibile nella stessa pagina.

Al termine

Quando si accede all'API REST di Astra Control tramite Curl o un linguaggio di programmazione, è necessario includere il token del bearer API nell'HTTP `Authorization` intestazione della richiesta.

Ciao mondo

È possibile eseguire un semplice comando curl sulla CLI della workstation per iniziare a utilizzare l'API REST di Astra Control e verificarne la disponibilità.

Prima di iniziare

L'utility Curl deve essere disponibile sulla workstation locale. È inoltre necessario disporre di un token API e dell'identificativo account associato. Vedere "Ottieni un token API" per ulteriori informazioni.

Esempio di arricciamento

Il seguente comando Curl recupera un elenco di utenti Astra. Fornire il <ACCOUNT_ID> e il <API_TOKEN> appropriati, come indicato.

```
curl --location --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/users' --header
'Content-Type: application/json' --header 'Authorization: Bearer
<API_TOKEN>'
```

Esempio di output JSON

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

Preparati a utilizzare i flussi di lavoro

Prima di utilizzarli con un'implementazione live, è necessario conoscere l'organizzazione e il formato dei flussi di lavoro Astra.

Introduzione

Un *workflow* è una sequenza di uno o più passaggi necessari per eseguire un'attività o un obiettivo amministrativo specifico. Ogni fase del flusso di lavoro di Astra Control è una delle seguenti:

- REST API Call (con dettagli come ad esempio CURL e JSON)
- Invocazione di un altro workflow Astra
- Attività correlate a varie attività (come prendere una decisione di progettazione richiesta)

I flussi di lavoro includono i passaggi principali e i parametri necessari per eseguire ogni attività. Forniscono un punto di partenza per personalizzare il tuo ambiente di automazione.

Parametri di input comuni

I parametri di input descritti di seguito sono comuni a tutti gli esempi di curl utilizzati per illustrare una chiamata API REST.

> Poiché questi parametri di input sono universalmente richiesti, non vengono descritti ulteriormente nei singoli flussi di lavoro. Se si utilizzano parametri di input aggiuntivi per un esempio specifico di curl, questi sono descritti nella sezione **parametri di input aggiuntivi**.

Parametri del percorso

Il percorso dell'endpoint utilizzato con ogni chiamata API REST include i seguenti parametri. Vedere anche "Formato URL" per ulteriori informazioni.

ID account

Questo è il valore UUIDv4 che identifica l'account Astra in cui viene eseguita l'operazione API. Vedere "Ottieni un token API" Per ulteriori informazioni su come individuare l'ID account.

Intestazioni delle richieste

A seconda della chiamata API REST, potrebbe essere necessario includere diverse intestazioni di richiesta.

Autorizzazione

Tutte le chiamate API nei flussi di lavoro richiedono un token API per identificare l'utente. È necessario includere il token in `Authorization` intestazione della richiesta. Vedere "Ottieni un token API" Per ulteriori informazioni sulla generazione di un token API.

Tipo di contenuto

Con le richieste HTTP POST e PUT in cui JSON è incluso nel corpo della richiesta, è necessario dichiarare il tipo di supporto in base alla risorsa Astra. Ad esempio, è possibile includere l'intestazione `Content-Type: application/astra-appSnap+json` quando si crea uno snapshot per un'applicazione gestita.

Accettare

È possibile dichiarare il tipo di supporto specifico del contenuto previsto nella risposta in base alla risorsa Astra. Ad esempio, è possibile includere l'intestazione `Accept: application/astra-appBackup+json` quando si elencano i backup per un'applicazione gestita. Tuttavia, per semplicità, i campioni di arricciatura nei flussi di lavoro accettano tutti i tipi di supporto.

Presentazione di token e identificatori

Il token API e gli altri valori ID utilizzati con gli esempi di curl sono opachi e non sono comprensibili. Per migliorare la leggibilità dei campioni, non vengono utilizzati i valori token e ID effettivi. Piuttosto, vengono utilizzate parole chiave riservate più piccole che hanno diversi benefici:

- I campioni Curl e JSON sono più chiari e comprensibili.
- Poiché tutte le parole chiave utilizzano lo stesso formato con parentesi quadre e lettere maiuscole, è possibile identificare rapidamente la posizione e il contenuto da inserire o estrarre.
- Nessun valore viene perso perché i parametri originali non possono essere copiati e utilizzati con un'implementazione effettiva.

Ecco alcune delle parole chiave riservate più comuni utilizzate negli esempi di curl. Questo elenco non è esaustivo e vengono utilizzate parole chiave aggiuntive in base alle necessità. Il loro significato dovrebbe essere ovvio in base al contesto.

Parola chiave	Tipo	Descrizione
<ACCOUNT_ID>	Percorso	Il valore UUIDv4 che identifica l'account in cui viene eseguita l'operazione API.
<API_TOKEN>	Intestazione	Il token del bearer che identifica e autorizza il chiamante.
<APP_ID>	Percorso	Il valore UUIDv4 che identifica l'applicazione per la chiamata API.

Categorie di workflow

Sono disponibili due ampie categorie di flussi di lavoro Astra in base al modello di implementazione. Se si utilizza Astra Control Center, è necessario iniziare con i flussi di lavoro dell'infrastruttura e passare ai flussi di

lavoro di gestione. Quando si utilizza Astra Control Service, in genere è possibile accedere direttamente ai flussi di lavoro di gestione.

> Gli esempi di arricciatura nei flussi di lavoro utilizzano l'URL per Astra Control Service. È necessario modificare l'URL quando si utilizza l'Astra Control Center on-premise in base all'ambiente in uso.

Flussi di lavoro dell'infrastruttura

Questi flussi di lavoro si occupano dell'infrastruttura Astra, inclusi credenziali, bucket e backend dello storage. Sono necessari con Astra Control Center, ma nella maggior parte dei casi possono essere utilizzati anche con Astra Control Service. I flussi di lavoro si concentrano sulle attività necessarie per stabilire e gestire un cluster gestito da Astra.

Workflow di gestione

È possibile utilizzare questi flussi di lavoro dopo aver gestito un cluster. I flussi di lavoro si concentrano sulla protezione delle applicazioni e supportano operazioni come il backup, il ripristino e la clonazione di un'applicazione.

Concetti di base di Kubernetes

Esistono diversi concetti di Kubernetes rilevanti quando si utilizza l'API ASTRA REST.

Oggetti

Gli oggetti mantenuti in un ambiente Kubernetes sono entità persistenti che rappresentano la configurazione del cluster. Questi oggetti descrivono collettivamente lo stato del sistema, incluso il carico di lavoro del cluster.

Spazi dei nomi

Gli spazi dei nomi forniscono una tecnica per isolare le risorse all'interno di un singolo cluster. Questa struttura organizzativa è utile quando si dividono i tipi di lavoro, gli utenti e le risorse. Gli oggetti con un *ambito dello spazio dei nomi* devono essere univoci all'interno dello spazio dei nomi, mentre quelli con un *ambito del cluster* devono essere univoci nell'intero cluster.

Etichette

Le etichette possono essere associate agli oggetti Kubernetes. Descrivono gli attributi che utilizzano coppie chiave-valore e possono imporre un'organizzazione arbitraria sul cluster, che può essere utile a un'organizzazione ma non è al di fuori dell'operazione Kubernetes principale.

Implementazione core REST

Servizi web REST

Representational state Transfer (REST) è uno stile per la creazione di applicazioni web distribuite. Quando viene applicato alla progettazione di un'API di servizi Web, stabilisce un insieme di tecnologie mainstream e Best practice per esporre le risorse basate su server e gestire i loro stati. Mentre REST fornisce una base coerente per lo sviluppo delle applicazioni, i dettagli di ciascuna API possono variare in base alle scelte di progettazione specifiche. Prima di utilizzarla con una distribuzione live, è necessario conoscere le caratteristiche dell'API REST di Astra Control.

Risorse e rappresentazione dello stato

Le risorse sono i componenti di base di un sistema basato su web. Quando si crea un'applicazione di servizi Web REST, le attività di progettazione iniziali includono:

- Identificazione delle risorse di sistema o basate su server

  Ogni sistema utilizza e gestisce le risorse. Una risorsa può essere un file, una transazione di business, un processo o un'entità amministrativa. Una delle prime attività nella progettazione di un'applicazione basata sui servizi web REST è quella di identificare le risorse.

- Definizione degli stati delle risorse e delle operazioni di stato associate

  Le risorse si trovano sempre in un numero limitato di stati. Gli stati, così come le operazioni associate utilizzate per influenzare i cambiamenti di stato, devono essere chiaramente definiti.

Endpoint URI

Ogni risorsa REST deve essere definita e resa disponibile utilizzando uno schema di indirizzamento ben definito. Gli endpoint in cui sono situate e identificate le risorse utilizzano un URI (Uniform Resource Identifier). L'URI fornisce un framework generale per la creazione di un nome univoco per ogni risorsa nella rete. L'URL (Uniform Resource Locator) è un tipo di URI utilizzato con i servizi Web per identificare e accedere alle risorse. Le risorse sono in genere esposte in una struttura gerarchica simile a una directory di file.

Messaggi HTTP

HTTP (Hypertext Transfer Protocol) è il protocollo utilizzato dal client e dal server dei servizi Web per scambiare messaggi di richiesta e risposta relativi alle risorse. Durante la progettazione di un'applicazione di servizi Web, i metodi HTTP vengono mappati alle risorse e alle azioni di gestione dello stato corrispondenti. HTTP è stateless. Pertanto, per associare un insieme di richieste e risposte correlate come parte di una transazione, è necessario includere informazioni aggiuntive nelle intestazioni HTTP portate con i flussi di dati di richiesta e risposta.

Formattazione JSON

Sebbene le informazioni possano essere strutturate e trasferite tra un client e un server di servizi Web in diversi modi, l'opzione più diffusa è JavaScript Object Notation (JSON). JSON è uno standard di settore per la rappresentazione di semplici strutture di dati in testo normale e viene utilizzato per trasferire informazioni di

stato che descrivono le risorse. L'API REST di Astra Control utilizza JSON per formattare i dati trasportati nel corpo di ogni richiesta e risposta HTTP.

Risorse e raccolte

L'API REST di Astra Control fornisce l'accesso alle istanze di risorse e alle raccolte di istanze di risorse.

> Concettualmente, una RISORSA REST* è simile a un **oggetto** come definito con i linguaggi e i sistemi di programmazione orientata agli oggetti (OOP). A volte questi termini vengono utilizzati in modo intercambiabile. In generale, la "risorsa" è preferibile quando viene utilizzata nel contesto dell'API REST esterna, mentre l'oggetto viene utilizzato per i corrispondenti dati dell'istanza stateful memorizzati nel server.

Attributi delle risorse Astra

L'API REST di Astra Control è conforme ai principi di progettazione RESTful. Ogni istanza di risorsa Astra viene creata in base a un tipo di risorsa ben definito. Un insieme di istanze di risorse dello stesso tipo viene definito **insieme**. Le chiamate API agiscono su singole risorse o raccolte di risorse.

Tipi di risorse

I tipi di risorse inclusi nell'API REST di Astra Control hanno le seguenti caratteristiche:

- Ogni tipo di risorsa viene definito utilizzando uno schema (in genere in JSON)
- Ogni schema delle risorse include il tipo e la versione delle risorse
- I tipi di risorse sono univoci a livello globale

Istanze di risorse

Le istanze di risorse disponibili tramite l'API REST di Astra Control hanno le seguenti caratteristiche:

- Le istanze di risorse vengono create in base a un singolo tipo di risorsa
- Il tipo di risorsa viene indicato utilizzando il valore del tipo di supporto
- Le istanze sono composte da dati stateful gestiti dal servizio Astra
- Ogni istanza è accessibile attraverso un URL univoco e di lunga durata
- Nei casi in cui un'istanza di risorsa può avere più di una rappresentazione, è possibile utilizzare diversi tipi di supporto per richiedere la rappresentazione desiderata

Raccolte di risorse

Le raccolte di risorse disponibili tramite l'API REST di Astra Control hanno le seguenti caratteristiche:

- L'insieme di istanze di risorse di un singolo tipo di risorsa è noto come insieme
- Le raccolte di risorse hanno un URL unico e di lunga durata

Identificatori delle istanze

A ogni istanza di risorsa viene assegnato un identificatore al momento della creazione. Questo identificatore è un valore UUIDv4 a 128 bit. I valori UUIDv4 assegnati sono globalmente univoci e immutabili. Dopo aver eseguito una chiamata API che crea una nuova istanza, viene restituito al chiamante un URL con l'id associato in a. `Location` Intestazione della risposta HTTP. È possibile estrarre l'identificatore e utilizzarlo nelle chiamate successive quando si fa riferimento all'istanza della risorsa.

L'identificatore di risorsa è la chiave principale utilizzata per le raccolte.

Struttura comune per le risorse Astra

Ogni risorsa Astra Control viene definita utilizzando una struttura comune.

Dati comuni

Ogni risorsa Astra contiene i valori chiave mostrati nella tabella seguente.

Chiave	Descrizione
tipo	Un tipo di risorsa globalmente univoco, noto come **tipo di risorsa**.
versione	Identificatore di versione noto come **versione della risorsa**.
id	Identificatore univoco globale noto come **resource identifier**.
metadati	Oggetto JSON contenente varie informazioni, incluse le etichette utente e di sistema.

Oggetto metadata

L'oggetto JSON di metadati incluso in ogni risorsa Astra contiene i valori chiave mostrati nella tabella seguente.

Chiave	Descrizione
etichette	Array JSON di etichette specificate dal client associate alla risorsa.
CreationTimestamp	Stringa JSON contenente un indicatore data e ora che indica quando è stata creata la risorsa.
ModificationTimestamp	Stringa JSON contenente un timestamp formattato ISO-8601 che indica l'ultima modifica della risorsa.
CreatedBy	Stringa JSON contenente l'identificatore UUIDv4 dell'id utente che ha creato la risorsa. Se la risorsa è stata creata da un componente di sistema interno e non esiste un UUID associato all'entità di creazione, viene utilizzato l'UID **null**.

Stato della risorsa

Risorse selezionate a `state` valore utilizzato per orchestrare le transizioni del ciclo di vita e controllare l'accesso.

Dettagli HTTP

L'API REST di Astra Control utilizza HTTP e i relativi parametri per agire sulle istanze e sugli insiemi di risorse. Di seguito sono presentati i dettagli dell'implementazione HTTP.

Le transazioni API e il modello CRUD

L'API REST di Astra Control implementa un modello transazionale con operazioni ben definite e transizioni di stato.

Transazione API di richiesta e risposta

Ogni chiamata API REST viene eseguita come richiesta HTTP al servizio Astra. Ogni richiesta genera una risposta associata al client. Questa coppia richiesta-risposta può essere considerata una transazione API.

Supporto del modello operativo CRUD

Si accede a ciascuna delle istanze e raccolte di risorse disponibili tramite l'API REST di Astra Control in base al modello **CRUD**. Sono disponibili quattro operazioni, ciascuna delle quali viene mappata a un singolo metodo HTTP. Le operazioni includono:

- Creare
- Leggi
- Aggiornare
- Eliminare

Per alcune risorse Astra, è supportato solo un sottoinsieme di queste operazioni. Esaminare "Riferimento API" Per ulteriori informazioni su una chiamata API specifica.

Metodi HTTP

I metodi HTTP o i verbi supportati dall'API sono presentati nella tabella seguente.

Metodo	CRUD	Descrizione
OTTIENI	Leggi	Recupera le proprietà degli oggetti per un'istanza o una raccolta di risorse. Questa operazione viene considerata un'operazione **list** quando utilizzata con una raccolta.
POST	Creare	Crea una nuova istanza di risorsa in base ai parametri di input. L'URL a lungo termine viene restituito in un `Location` intestazione della risposta.
IN PRIMO PIANO	Aggiornare	Aggiorna un'intera istanza di risorsa con il corpo di richiesta JSON fornito. I valori chiave non modificabili dall'utente vengono conservati.
ELIMINARE	Eliminare	Elimina un'istanza di risorsa esistente.

Intestazioni di richiesta e risposta

La seguente tabella riassume le intestazioni HTTP utilizzate con l'API REST di Astra Control.

Vedere "RFC 7232" e. "RFC 7233" per ulteriori informazioni.

Intestazione	Tipo	Note sull'utilizzo
Accettare	Richiesta	Se il valore è "/" o non viene fornito, `application/json` Viene restituito nell'intestazione di risposta Content-Type. Se il valore è impostato su Astra Resource Media Type, lo stesso tipo di supporto viene restituito nell'intestazione Content-Type.
Autorizzazione	Richiesta	Token bearer con la chiave API per l'utente.
Tipo di contenuto	Risposta	Restituito in base a. `Accept` intestazione della richiesta.
ETAG	Risposta	Incluso con un successo come definito con RFC 7232. Il valore è una rappresentazione esadecimale del valore MD5 per l'intera risorsa JSON.

Intestazione	Tipo	Note sull'utilizzo
IF-Match	Richiesta	Intestazione di richiesta di precondizione implementata come descritto nella sezione 3.1 RFC 7232 e supporto per richieste **PUT**.
IF-modified-since	Richiesta	Intestazione di richiesta di precondizione implementata come descritto nella sezione 3.4 RFC 7232 e supporto per richieste **PUT**.
IF-unmodified-since	Richiesta	Intestazione di richiesta di precondizione implementata come descritto nella sezione 3.4 RFC 7232 e supporto per richieste **PUT**.
Posizione	Risposta	Contiene l'URL completo della risorsa appena creata.

Parametri di query

I seguenti parametri di query sono disponibili per l'utilizzo con le raccolte di risorse. Vedere "Utilizzo delle raccolte" per ulteriori informazioni.

Parametro di query	Descrizione
includi	Contiene i campi che devono essere restituiti durante la lettura di una raccolta.
filtro	Indica i campi che devono corrispondere per la restituzione di una risorsa durante la lettura di una raccolta.
OrderBy	Determina l'ordinamento delle risorse restituite durante la lettura di una raccolta.
limite	Limita il numero massimo di risorse restituite durante la lettura di una raccolta.
saltare	Imposta il numero di risorse da passare e saltare durante la lettura di una raccolta.
conta	Indica se il numero totale di risorse deve essere restituito nell'oggetto metadata.

Codici di stato HTTP

I codici di stato HTTP utilizzati dall'API REST di Astra Control sono descritti di seguito.

> L'API REST di Astra Control utilizza anche lo standard **Problem Details for HTTP API**. Vedere "Diagnostica e supporto" per ulteriori informazioni.

Codice	Significato	Descrizione
200	OK	Indica il successo delle chiamate che non creano una nuova istanza di risorsa.
201	Creato	Un oggetto viene creato correttamente e l'intestazione della risposta di posizione include l'identificatore univoco dell'oggetto.
204	Nessun contenuto	La richiesta è stata completata, anche se non è stato restituito alcun contenuto.
400	Richiesta errata	L'input della richiesta non viene riconosciuto o non è appropriato.
401	Non autorizzato	L'utente non è autorizzato e deve autenticarsi.

Codice	Significato	Descrizione
403	Vietato	Accesso negato a causa di un errore di autorizzazione.
404	Non trovato	La risorsa a cui si fa riferimento nella richiesta non esiste.
409	Conflitto	Tentativo di creazione di un oggetto non riuscito perché l'oggetto esiste già.
500	Errore interno	Si è verificato un errore interno generale nel server.
503	Servizio non disponibile	Il servizio non è pronto a gestire la richiesta per qualche motivo.

Formato URL

La struttura generale dell'URL utilizzato per accedere a un'istanza o a una raccolta di risorse attraverso l'API REST è composta da diversi valori. Questa struttura riflette il modello a oggetti sottostante e la progettazione del sistema.

Account come root

La radice del percorso delle risorse per ogni endpoint REST è l'account Astra. Quindi, tutti i percorsi nell'URL iniziano con `/account/{account_id}` dove `account_id` È il valore UUIDv4 univoco per l'account. Struttura interna questa riflette una progettazione in cui l'accesso a tutte le risorse si basa su un account specifico.

Categoria di risorse degli endpoint

Gli endpoint delle risorse Astra sono suddivisi in tre categorie:

- Core (`/core`)
- Applicazione gestita (`/k8s`)
- Topologia (`/topology`)

Vedere "Risorse" per ulteriori informazioni.

Versione categoria

Ciascuna delle tre categorie di risorse dispone di una versione globale che controlla la versione delle risorse a cui si accede. Per convenzione e definizione, passaggio a una nuova versione principale di una categoria di risorse (ad esempio, da `/v1` a. `/v2`) Introdurrà le ultime modifiche nell'API.

Istanza o raccolta di risorse

È possibile utilizzare una combinazione di tipi di risorse e identificatori nel percorso, in base all'accesso a un'istanza o a una raccolta di risorse.

Esempio

- Percorso delle risorse

  In base alla struttura presentata in precedenza, un percorso tipico verso un endpoint è: `/accounts/{account_id}/core/v1/users`.

- URL completo

L'URL completo per l'endpoint corrispondente è: `https://astra.netapp.io/accounts/{account_id}/core/v1/users`.

Risorse ed endpoint

È possibile accedere alle risorse fornite tramite l'API REST di Astra Control per automatizzare un'implementazione Astra. Ogni risorsa è disponibile attraverso uno o più endpoint. Di seguito viene fornita un'introduzione alle risorse REST che è possibile utilizzare come parte di un'implementazione dell'automazione.

> Il formato del percorso e dell'URL completo utilizzati per accedere alle risorse di Astra Control si basa su diversi valori. Vedere "Formato URL" per ulteriori informazioni. Vedere anche "Riferimento API" Per ulteriori informazioni sull'utilizzo delle risorse e degli endpoint Astra.

Riepilogo delle risorse REST di Astra Control

Gli endpoint delle risorse principali forniti nell'API REST di Astra Control sono organizzati in tre categorie. È possibile accedere a ciascuna risorsa con il set completo di operazioni CRUD (creazione, lettura, aggiornamento, eliminazione), salvo dove indicato.

La colonna **Release** indica la release Astra quando la risorsa è stata introdotta per la prima volta. Questo campo è in grassetto per le risorse aggiunte di recente con la release corrente.

Risorse di base

Gli endpoint principali delle risorse forniscono i servizi di base necessari per stabilire e mantenere l'ambiente di runtime Astra.

Risorsa	Rilasciare	Descrizione
Account	21.12	Le risorse dell'account consentono di gestire i tenant isolati all'interno dell'ambiente di implementazione di Astra Control multi-tenant.
ASUP	21.08	Le risorse ASUP rappresentano i bundle AutoSupport inoltrati al supporto NetApp.
Certificato	**22.08**	Le risorse dei certificati rappresentano i certificati installati utilizzati per l'autenticazione avanzata delle connessioni in uscita.
Credenziale	21.04	Le risorse delle credenziali contengono informazioni relative alla sicurezza che possono essere utilizzate con utenti Astra, cluster, bucket e backend di storage.
Diritto	21.08	Le risorse relative ai diritti rappresentano le funzionalità e le capacità disponibili per un account in base alle licenze e alle sottoscrizioni attive.
Evento	21.04	Le risorse degli eventi rappresentano tutti gli eventi che si verificano nel sistema, incluso il sottoinsieme classificato come notifiche.
Gancio di esecuzione	21.12	Le risorse di esecuzione hook rappresentano script personalizzati che è possibile eseguire prima o dopo l'esecuzione di uno snapshot di un'applicazione gestita.
Funzione	21.08	Le risorse delle funzioni rappresentano le funzioni Astra selezionate che è possibile interrogare per determinare se sono attivate o disattivate nel sistema. L'accesso è limitato alla sola lettura.

Risorsa	Rilasciare	Descrizione
Gruppo	**22.08**	Le risorse del gruppo rappresentano i gruppi Astra e le risorse associate. Nella release corrente sono supportati solo i gruppi LDAP.
Origine gancio	21.12	Le risorse di origine hook rappresentano il codice di origine effettivo utilizzato con un gancio di esecuzione. La separazione del codice sorgente dal controllo di esecuzione offre diversi vantaggi, ad esempio la possibilità di condividere gli script.
Licenza	21.08	Le risorse di licenza rappresentano le licenze disponibili per un account Astra.
Notifica	21.04	Le risorse di notifica rappresentano gli eventi Astra che hanno una destinazione di notifica. L'accesso viene fornito in base all'utente.
Pacchetto	22.04	Le risorse del pacchetto forniscono la registrazione e l'accesso alle definizioni dei pacchetti. I pacchetti software sono composti da vari componenti, tra cui file, immagini e altri elementi.
Binding dei ruoli	21.04	Le risorse di associazione dei ruoli rappresentano le relazioni tra coppie specifiche di utenti e account. Oltre al collegamento tra i due, viene specificato un set di autorizzazioni per ciascuno attraverso un ruolo specifico.
Impostazione	21.08	Le risorse di impostazione rappresentano un insieme di coppie chiave-valore che descrivono una funzionalità per un account Astra specifico.
Iscrizione	21.08	Le risorse di abbonamento rappresentano gli abbonamenti attivi per un account Astra.
Token	21.04	Le risorse token rappresentano i token disponibili per accedere a livello di programmazione all'API REST di Astra Control.
Notifica non letta	21.04	Le risorse di notifica non lette rappresentano le notifiche assegnate a un utente specifico ma non ancora lette.
Eseguire l'upgrade	22.04	Le risorse di aggiornamento forniscono l'accesso ai componenti software e la possibilità di avviare gli aggiornamenti.
Utente	21.04	Le risorse utente rappresentano gli utenti Astra in grado di accedere al sistema in base al proprio ruolo definito.

Risorse applicative gestite

Gli endpoint delle risorse applicative gestite forniscono l'accesso alle applicazioni Kubernetes gestite.

Risorsa	Rilasciare	Descrizione
Risorsa applicativa	21.04	Le risorse applicative rappresentano raccolte interne di informazioni di stato necessarie per gestire le applicazioni Astra.
Backup dell'applicazione	21.04	Le risorse di backup delle applicazioni rappresentano i backup delle applicazioni gestite.
Snapshot dell'applicazione	21.04	Le risorse di snapshot delle applicazioni rappresentano snapshot delle applicazioni gestite.

Risorsa	Rilasciare	Descrizione
Override del gancio di esecuzione	21.12	Le risorse di override degli uncini di esecuzione consentono di disattivare gli uncini di esecuzione predefiniti NetApp precaricati per applicazioni specifiche in base alle necessità.
Pianificazione	21.04	Le risorse di pianificazione rappresentano le operazioni di protezione dei dati pianificate per le applicazioni gestite come parte di una policy di protezione dei dati.

Risorse per la topologia

Gli endpoint delle risorse di topologia forniscono l'accesso alle applicazioni non gestite e alle risorse di storage.

Risorsa	Rilasciare	Descrizione
App	21.04	Le risorse applicative rappresentano tutte le applicazioni Kubernetes, incluse quelle non gestite da Astra.
AppMirror	**22.08**	Le risorse di AppMirror rappresentano le risorse di AppMirror da fornire per la gestione delle relazioni di mirroring delle applicazioni.
Bucket	21.08	Le risorse del bucket rappresentano i bucket cloud S3 utilizzati per memorizzare i backup delle applicazioni gestite da Astra.
Cloud	21.08	Le risorse cloud rappresentano i cloud a cui i client Astra possono connettersi per gestire cluster e applicazioni.
Cluster	21.08	Le risorse del cluster rappresentano i cluster Kubernetes non gestiti da Kubernetes.
Nodo del cluster	21.12	Le risorse dei nodi del cluster forniscono una risoluzione aggiuntiva consentendo di accedere ai singoli nodi all'interno di un cluster Kubernetes.
Cluster gestito	21.08	Le risorse del cluster gestito rappresentano i cluster Kubernetes attualmente gestiti da Kubernetes.
Back-end di storage gestito	21.12	Le risorse di back-end dello storage gestito consentono di accedere alle rappresentazioni astratte dei provider di storage back-end. Questi backend di storage possono essere utilizzati dai cluster e dalle applicazioni gestiti.
Namespace	21.12	Le risorse dello spazio dei nomi forniscono l'accesso agli spazi dei nomi utilizzati all'interno di un cluster Kubernetes.
Back-end dello storage	21.08	Le risorse di back-end dello storage rappresentano i provider di servizi di storage che possono essere utilizzati dai cluster e dalle applicazioni gestiti da Astra.
Classe di storage	21.08	Le risorse della classe di storage rappresentano classi o tipi diversi di storage rilevati e disponibili per uno specifico cluster gestito.
Volume	21.04	Le risorse dei volumi rappresentano i volumi di storage Kubernetes associati alle applicazioni gestite.

Risorse ed endpoint aggiuntivi

Esistono diverse risorse aggiuntive e endpoint che è possibile utilizzare per supportare un'implementazione Astra.

> Queste risorse e questi endpoint non sono attualmente inclusi nella documentazione di riferimento dell'API REST di Astra Control.

OpenAPI

Gli endpoint OpenAPI forniscono l'accesso al documento JSON OpenAPI corrente e ad altre risorse correlate.

OpenMetrics

Gli endpoint OpenMetrics forniscono l'accesso alle metriche dell'account attraverso la risorsa OpenMetrics. Il supporto è disponibile con il modello di implementazione di Astra Control Center.

Ulteriori considerazioni sull'utilizzo

Sicurezza RBAC

L'API ASTRA REST supporta il RBAC (role-based access control) per concedere e limitare l'accesso alle funzioni del sistema.

Ruoli Astra

Ogni utente Astra viene assegnato a un singolo ruolo che determina le azioni che possono essere eseguite. I ruoli sono organizzati in una gerarchia come descritto nella tabella seguente.

Ruolo	Descrizione
Proprietario	Dispone di tutte le autorizzazioni del ruolo Admin e può anche eliminare gli account Astra.
Amministratore	Dispone di tutte le autorizzazioni del ruolo membro e può anche invitare gli utenti a unirsi a un account.
Membro	È in grado di gestire completamente l'applicazione Astra e le risorse di calcolo.
Visualizzatore	Limitato solo alla visualizzazione delle risorse.

RBAC migliorato con granularità dello spazio dei nomi

Questa funzionalità è stata introdotta con la versione 22.04 dell'API ASTRA REST.

Quando viene stabilita un'associazione di ruolo per un utente specifico, è possibile applicare un vincolo per limitare gli spazi dei nomi a cui l'utente ha accesso. Questo vincolo può essere definito in diversi modi, come descritto nella tabella seguente. Vedere il parametro `roleContraints` Nell'API di associazione dei ruoli per ulteriori informazioni.

Spazi dei nomi	Descrizione
Tutto	L'utente può accedere a tutti gli spazi dei nomi attraverso il parametro jolly "*"". Questo è il valore predefinito per mantenere la compatibilità con le versioni precedenti.
Nessuno	L'elenco dei vincoli viene specificato anche se è vuoto. Ciò indica che l'utente non può accedere a nessuno spazio dei nomi.
Elenco dei namespace	Viene incluso l'UUID di uno spazio dei nomi che limita l'utente al singolo spazio dei nomi. Per consentire l'accesso a più spazi dei nomi, è possibile utilizzare anche un elenco separato da virgole.
Etichetta	Viene specificata un'etichetta e viene consentito l'accesso a tutti gli spazi dei nomi corrispondenti.

Lavorare con le raccolte

L'API REST di Astra Control offre diversi modi per accedere alle raccolte di risorse attraverso i parametri di query definiti.

Selezione dei valori

È possibile specificare quali coppie chiave-valore devono essere restituite per ogni istanza di risorsa utilizzando `include` parametro. Tutte le istanze vengono restituite nel corpo della risposta.

Filtraggio

Il filtraggio delle risorse di raccolta consente a un utente API di specificare le condizioni che determinano se una risorsa viene restituita nel corpo della risposta. Il `filter` il parametro viene utilizzato per indicare la condizione di filtraggio.

Ordinamento

L'ordinamento delle risorse di raccolta consente a un utente API di specificare l'ordine in cui le risorse vengono restituite nel corpo della risposta. Il `orderBy` il parametro viene utilizzato per indicare la condizione di filtraggio.

Impaginazione

È possibile applicare l'impaginazione limitando il numero di istanze di risorse restituite su una richiesta utilizzando `limit` parametro.

Conta

Se si include il parametro booleano `count` impostare su `true`, il numero di risorse nella matrice restituita per una data risposta è fornito nella sezione dei metadati.

Diagnostica e supporto

Con l'API REST di Astra Control sono disponibili diverse funzionalità di supporto che possono essere utilizzate per la diagnostica e il debug.

Risorse API

Ci sono diverse funzionalità di Astra esposte attraverso le risorse API che forniscono informazioni diagnostiche e supporto.

Tipo	Descrizione
Evento	Attività di sistema registrate come parte dell'elaborazione Astra.
Notifica	Un sottoinsieme di eventi considerati abbastanza importanti da essere presentati all'utente.
Notifica non letta	Le notifiche che devono ancora essere lette o recuperate dall'utente.

Revocare un token API

È possibile revocare un token API all'interfaccia web Astra quando non è più necessario.

Prima di iniziare

Hai bisogno di un account Astra. È inoltre necessario identificare i token che si desidera revocare.

A proposito di questa attività

Una volta revocato, il token risulta immediatamente e permanentemente inutilizzabile.

Fasi

1. Accedi ad Astra utilizzando le credenziali del tuo account.

   Accedere al seguente sito per Astra Control Service: "https://astra.netapp.io"

2. Fare clic sull'icona a forma di figura nella parte superiore destra della pagina e selezionare **API access**.
3. Selezionare il token o i token che si desidera revocare.
4. Nella casella di riepilogo **azioni**, fare clic su **revoca token**.

Flussi di lavoro dell'infrastruttura

Prima di iniziare

È possibile utilizzare questi flussi di lavoro per creare e gestire l'infrastruttura utilizzata con un'implementazione di Astra Control Center. In molti casi, i flussi di lavoro possono essere utilizzati anche con Astra Control Service.

> Questi flussi di lavoro possono essere ampliati e migliorati da NetApp in qualsiasi momento, pertanto è necessario esaminarli periodicamente.

Preparazione generale

Prima di utilizzare uno qualsiasi dei flussi di lavoro Astra, assicurarsi di rivedere "Preparati a utilizzare i flussi di lavoro".

Categorie di workflow

I flussi di lavoro dell'infrastruttura sono organizzati in diverse categorie per facilitare l'individuazione di quello desiderato.

Categoria	Descrizione
Identità e accesso	Questi flussi di lavoro consentono di gestire l'identità e l'accesso ad Astra. Le risorse includono utenti, credenziali e token.
Configurazione LDAP	È possibile configurare Astra Control Center in modo che utilizzi LDAP per autenticare gli utenti selezionati.
Bucket	È possibile utilizzare questi flussi di lavoro per creare e gestire i bucket S3 utilizzati per memorizzare i backup.
Storage	Questi flussi di lavoro consentono di aggiungere e gestire volumi e backend di storage.
Cluster	È possibile aggiungere cluster Kubernetes gestiti che consentono di proteggere e supportare le applicazioni in essi contenute.

Identità e accesso

Elencare gli utenti

È possibile elencare gli utenti definiti per un account Astra specifico.

1. Elencare gli utenti

Eseguire la seguente chiamata API REST.

Metodo HTTP	Percorso
OTTIENI	/account/{account_id}/core/v1/users

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
includi	Query	No	Se si desidera, selezionare i valori che si desidera restituire nella risposta.

Esempio di curl: Restituisce tutti i dati per tutti gli utenti

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

Esempio di curl: Restituisce il nome, il cognome e l'id per tutti gli utenti

```
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>'
```

Esempio di output JSON

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

Configurazione LDAP

Preparazione per la configurazione LDAP

È possibile integrare Astra Control Center con un server LDAP (Lightweight Directory Access Protocol) per eseguire l'autenticazione per gli utenti Astra selezionati. LDAP è un protocollo standard di settore per l'accesso alle informazioni di directory distribuite e una scelta popolare per l'autenticazione aziendale.

Informazioni correlate

- "Roadmap delle specifiche tecniche LDAP"
- "LDAP versione 3"

Panoramica del processo di implementazione

Ad alto livello, è necessario eseguire diversi passaggi per configurare un server LDAP in modo da fornire l'autenticazione agli utenti Astra.

> Sebbene i passaggi presentati di seguito siano in sequenza, in alcuni casi è possibile eseguirli in un ordine diverso. Ad esempio, è possibile definire gli utenti e i gruppi Astra prima di configurare il server LDAP.

1. Revisione "Requisiti e limitazioni" per comprendere le opzioni, i requisiti e le limitazioni.
2. Selezionare un server LDAP e le opzioni di configurazione desiderate (inclusa la protezione).
3. Eseguire il flusso di lavoro "Configurare Astra per l'utilizzo di un server LDAP" Per integrare Astra con il server LDAP.
4. Esaminare gli utenti e i gruppi sul server LDAP per assicurarsi che siano definiti correttamente.
5. Eseguire il flusso di lavoro appropriato in "Aggiungere voci LDAP ad Astra" Identificare gli utenti da autenticare utilizzando LDAP.

Requisiti e limitazioni

Prima di configurare Astra per l'utilizzo di LDAP per l'autenticazione, è necessario esaminare gli elementi essenziali della configurazione di Astra presentati di seguito, incluse le limitazioni e le opzioni di configurazione.

Supportato solo con Astra Control Center

La piattaforma Astra Control offre due modelli di implementazione. L'autenticazione LDAP è supportata solo con le implementazioni di Astra Control Center.

Solo configurazione API REST

La versione corrente di Astra Control Center supporta solo la configurazione dell'autenticazione LDAP utilizzando l'API REST di Astra Control. Un aspetto importante di questa limitazione è che gli utenti LDAP non vengono visualizzati nella scheda Users (utenti) dell'interfaccia web Astra. Sono disponibili tramite l'API REST all'endpoint `../core/v1/users`.

Server LDAP richiesto

Per accettare ed elaborare le richieste di autenticazione Astra, è necessario disporre di un server LDAP. Active Directory di Microsoft è supportata con la release corrente di Astra Control Center.

Connessione sicura al server LDAP

Quando si configura il server LDAP in Astra, è possibile definire una connessione sicura. In questo caso è necessario un certificato per il protocollo LDAPS.

Configurare utenti o gruppi

Selezionare gli utenti da autenticare utilizzando LDAP. È possibile eseguire questa operazione identificando i singoli utenti o un gruppo di utenti. Gli account devono essere definiti sul server LDAP. Inoltre, devono essere identificati in Astra (tipo LDAP), che consente di inoltrare le richieste di autenticazione a LDAP.

Vincolo di ruolo quando si lega un utente o un gruppo

Con l'attuale release di Astra Control Center, l'unico valore supportato per `roleConstraint` è "*". Questo indica che l'utente non è limitato a un set limitato di spazi dei nomi e può accedervi tutti. Vedere "Aggiungere voci LDAP ad Astra" per ulteriori informazioni.

Credenziali LDAP

Le credenziali utilizzate da LDAP includono il nome utente (indirizzo e-mail) e la password associata.

Indirizzi e-mail univoci

Tutti gli indirizzi e-mail che fungono da nomi utente in un'implementazione di Astra Control Center devono essere univoci. Non è possibile aggiungere un utente LDAP con un indirizzo e-mail già definito in Astra. Se esiste un'email duplicata, devi prima eliminarla da Astra. Vedere "Rimuovere gli utenti" Per ulteriori informazioni, visitare il sito di documentazione di Astra Control Center.

È possibile definire prima utenti e gruppi LDAP

È possibile aggiungere utenti e gruppi LDAP a Astra Control Center anche se non esistono ancora in LDAP o se il server LDAP non è configurato. Ciò consente di preconfigurare gli utenti e i gruppi prima di configurare il server LDAP.

Un utente definito in più gruppi LDAP

Se un utente LDAP appartiene a più gruppi LDAP e ai gruppi sono stati assegnati ruoli diversi in Astra, il ruolo effettivo dell'utente al momento dell'autenticazione sarà il più privilegiato. Ad esempio, se a un utente è assegnato il `viewer` con il group1, ma ha il `member` ruolo nel group2, il ruolo dell'utente sarebbe `member`. Si basa sulla gerarchia utilizzata da Astra (dal più alto al più basso):

- Proprietario
- Amministratore
- Membro
- Visualizzatore

Sincronizzazione periodica dell'account

Astra sincronizza gli utenti e i gruppi IT con il server LDAP circa ogni 60 secondi. Quindi, se un utente o un gruppo viene aggiunto o rimosso da LDAP, può essere necessario fino a un minuto prima che sia disponibile in Astra.

Disattivazione e ripristino della configurazione LDAP

Prima di tentare di ripristinare la configurazione LDAP, è necessario disattivare l'autenticazione LDAP. Inoltre, per modificare il server LDAP (`connectionHost`), è necessario eseguire entrambe le operazioni. Vedere "Disattivare e ripristinare LDAP" per ulteriori informazioni.

Parametri API REST

I flussi di lavoro di configurazione LDAP effettuano chiamate API REST per eseguire le attività specifiche. Ogni

chiamata API può includere parametri di input come mostrato negli esempi forniti. Vedere "Riferimento API" per informazioni su come individuare la documentazione di riferimento.

Configurare Astra per l'utilizzo di un server LDAP

Selezionare un server LDAP e configurare Astra per utilizzare il server come provider di autenticazione. L'attività di configurazione consiste nei passaggi descritti di seguito. Ogni passaggio include una singola chiamata API REST.

1. Aggiungere un certificato CA

Eseguire la seguente chiamata API REST per aggiungere un certificato CA ad Astra.

> Questo passaggio è facoltativo e necessario solo se si desidera che Astra e LDAP comunichino su un canale sicuro utilizzando LDAPS.

Metodo HTTP	Percorso
POST	/account/{account_id}/core/v1/certificates

Esempio di input JSON

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

Tenere presente quanto segue sui parametri di input:

- `cert` È una stringa JSON contenente un certificato con codifica base64 e formato PKCS-11 (con codifica PEM).
- `isSelfSigned` deve essere impostato su `true` se il certificato è autofirmato. L'impostazione predefinita è `false`.

Esempio di arricciamento

```
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>'
```

Esempio di risposta JSON

```
{
  "type": "application/astra-certificate",
  "version": "1.0",
  "id": "a5212e7e-402b-4cff-bba0-63f3c6505199",
  "certUse": "rootCA",
  "cert": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUMyVEN",
  "cn": "adldap.example.com",
  "expiryTimestamp": "2023-07-08T20:22:07Z",
  "isSelfSigned": "true",
  "trustState": "trusted",
  "trustStateTransitions": [
  {
    "from": "untrusted",
    "to": [
      "trusted",
      "expired"
    ]
  },
  {
    "from": "trusted",
    "to": [
      "untrusted",
      "expired"
    ]
  },
  {
    "from": "expired",
    "to": [
     "untrusted",
     "trusted"
    ]
  }
  ],
  "trustStateDesired": "trusted",
  "trustStateDetails": [],
  "metadata": {
    "creationTimestamp": "2022-07-21T04:16:06Z",
    "modificationTimestamp": "2022-07-21T04:16:06Z",
    "createdBy": "8a02d2b8-a69d-4064-827f-36851b3e1e6e",
    "modifiedBy": "8a02d2b8-a69d-4064-827f-36851b3e1e6e",
    "labels": []
    }
}
```

2. Aggiungere le credenziali di binding

Eseguire la seguente chiamata API REST per aggiungere le credenziali BIND.

Metodo HTTP	Percorso
POST	/account/{account_id}/core/v1/credentials

Esempio di input JSON

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

Tenere presente quanto segue sui parametri di input:

- `bindDn` e. `password` Sono le credenziali bind codificate base64 dell'utente amministratore LDAP in grado di connettersi e cercare nella directory LDAP. `bindDn` È l'indirizzo e-mail dell'utente LDAP.

Esempio di arricciamento

```
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>'
```

Esempio di risposta JSON

```
{
    "type": "application/astra-credential",
    "version": "1.1",
    "id": "3bd9c8a7-f5a4-4c44-b778-90a85fc7d154",
    "name": "ldapBindCredential",
    "metadata": {
        "creationTimestamp": "2022-07-21T06:53:11Z",
        "modificationTimestamp": "2022-07-21T06:53:11Z",
        "createdBy": "527329f2-662c-41c0-ada9-2f428f14c137"
    }
}
```

Osservare i seguenti parametri di risposta:

- Il `id` della credenziale viene utilizzata nelle fasi successive del flusso di lavoro.

3. Recuperare l'UUID dell'impostazione LDAP

Eseguire la seguente chiamata API REST per recuperare l'UUID di `astra.account.ldap` Impostazione inclusa in Astra Control Center.

> Nell'esempio riportato di seguito viene utilizzato un parametro di query per filtrare la raccolta delle impostazioni. È invece possibile rimuovere il filtro per ottenere tutte le impostazioni e quindi cercare `astra.account.ldap`.

Metodo HTTP	Percorso
OTTIENI	/account/{account_id}/core/v1/settings

Esempio di arricciamento

```
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>'
```

Esempio di risposta JSON

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

4. Aggiornare l'impostazione LDAP

Eseguire la seguente chiamata API REST per aggiornare l'impostazione LDAP e completare la configurazione. Utilizzare `id` Valore della chiamata API precedente per `<SETTING_ID>` Valore nel percorso URL riportato di seguito.

> È possibile inviare una richiesta GET per l'impostazione specifica prima di visualizzare configSchema. In questo modo verranno fornite ulteriori informazioni sui campi obbligatori della configurazione.

Metodo HTTP	Percorso
IN PRIMO PIANO	/account/{account_id}/core/v1/settings/{setting_id}

Esempio di input JSON

```
{
  "type": "application/astra-setting",
  "version": "1.0",
  "desiredConfig": {
    "connectionHost": "myldap.example.com",
    "credentialId": "3bd9c8a7-f5a4-4c44-b778-90a85fc7d154",
    "groupBaseDN": "OU=groups,OU=astra,DC=example,DC=com",
    "isEnabled": "true",
    "port": 686,
    "secureMode": "LDAPS",
    "userBaseDN": "OU=users,OU=astra,DC=example,dc=com",
    "userSearchFilter": "((objectClass=User))",
    "vendor": "Active Directory"
    }
}
```

Tenere presente quanto segue sui parametri di input:

- `isEnabled` deve essere impostato su `true` oppure si potrebbe verificare un errore.
- `credentialId` è l'id della credenziale bind creata in precedenza.
- `secureMode` deve essere impostato su `LDAP` oppure `LDAPS` in base alla configurazione del passaggio precedente.
- Solo "Active Directory" è supportato come vendor.

Esempio di arricciamento

```
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>'
```

Se la chiamata ha esito positivo, viene restituita la risposta HTTP 204.

5. Recuperare l'impostazione LDAP

È possibile eseguire la seguente chiamata API REST per recuperare le impostazioni LDAP e confermare l'aggiornamento.

Metodo HTTP	Percorso
OTTIENI	/account/{account_id}/core/v1/settings/{setting_id}

Esempio di arricciamento

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

Esempio di risposta JSON


```
{
  "items": [
  {
    "type": "application/astra-setting",
    "version": "1.0",
    "metadata": {
      "creationTimestamp": "2022-06-17T21:16:31Z",
      "modificationTimestamp": "2022-07-21T07:12:20Z",
      "labels": [],
      "createdBy": "system",
      "modifiedBy": "00000000-0000-0000-0000-000000000000"
    },
    "id": "12072b56-e939-45ec-974d-2dd83b7815df",
    "name": "astra.account.ldap",
    "desiredConfig": {
      "connectionHost": "10.193.61.88",
      "credentialId": "3bd9c8a7-f5a4-4c44-b778-90a85fc7d154",
      "groupBaseDN": "ou=groups,ou=astra,dc=example,dc=com",
      "isEnabled": "true",
      "port": 686,
      "secureMode": "LDAPS",
      "userBaseDN": "ou=users,ou=astra,dc=example,dc=com",
      "userSearchFilter": "((objectClass=User))",
      "vendor": "Active Directory"
    },
    "currentConfig": {
      "connectionHost": "10.193.160.209",
      "credentialId": "3bd9c8a7-f5a4-4c44-b778-90a85fc7d154",
      "groupBaseDN": "ou=groups,ou=astra,dc=example,dc=com",
      "isEnabled": "true",
      "port": 686,
      "secureMode": "LDAPS",
      "userBaseDN": "ou=users,ou=astra,dc=example,dc=com",
```

```
      "userSearchFilter": "((objectClass=User))",
      "vendor": "Active Directory"
    },
    "configSchema": {
      "$schema": "http://json-schema.org/draft-07/schema#",
      "title": "astra.account.ldap",
      "type": "object",
      "properties": {
        "connectionHost": {
          "type": "string",
          "description": "The hostname or IP address of your LDAP server."
        },
        "credentialId": {
          "type": "string",
          "description": "The credential ID for LDAP account."
        },
        "groupBaseDN": {
          "type": "string",
          "description": "The base DN of the tree used to start the group
search. The system searches the subtree from the specified location."
        },
        "groupSearchCustomFilter": {
          "type": "string",
          "description": "Type of search that controls the default group
search filter used."
        },
        "isEnabled": {
          "type": "string",
          "description": "This property determines if this setting is
enabled or not."
        },
        "port": {
          "type": "integer",
          "description": "The port on which the LDAP server is running."
        },
        "secureMode": {
          "type": "string",
          "description": "The secure mode LDAPS or LDAP."
        },
        "userBaseDN": {
          "type": "string",
          "description": "The base DN of the tree used to start the user
search. The system searches the subtree from the specified location."
        },
        "userSearchFilter": {
          "type": "string",
```

```
          "description": "The filter used to search for users according a
search criteria."
        },
        "vendor": {
          "type": "string",
          "description": "The LDAP provider you are using.",
          "enum": ["Active Directory"]
        }
      },
      "additionalProperties": false,
      "required": [
        "connectionHost",
        "secureMode",
        "credentialId",
        "userBaseDN",
        "userSearchFilter",
        "groupBaseDN",
        "vendor",
        "isEnabled"
      ]
      },
      "state": "valid",
    }
  ],
  "metadata": {}
}
```


Individuare il `state` nella risposta che avrà uno dei valori nella tabella seguente.

Stato	Descrizione
in sospeso	Il processo di configurazione è ancora attivo e non ancora completato.
valido	La configurazione è stata completata correttamente e. `currentConfig` nella risposta corrisponde `desiredConfig`.
errore	Il processo di configurazione LDAP non è riuscito.

Aggiungere voci LDAP ad Astra

Una volta configurato LDAP come provider di autenticazione per Astra Control Center, è possibile selezionare gli utenti LDAP che Astra eseguirà l'autenticazione utilizzando le credenziali LDAP. Ogni utente deve avere un ruolo in Astra prima di poter accedere ad Astra attraverso l'API REST di Astra Control.

Esistono due modi per configurare Astra per assegnare i ruoli. Scegliere quello più adatto al proprio ambiente.

- "Aggiungere e associare un singolo utente"

- "Aggiungere e associare un gruppo"

> Le credenziali LDAP sono sotto forma di nome utente come indirizzo e-mail e password LDAP associata.

Aggiungere e associare un singolo utente

È possibile assegnare un ruolo a ciascun utente Astra utilizzato dopo l'autenticazione LDAP. Ciò è appropriato quando vi è un numero limitato di utenti e ciascuno potrebbe avere caratteristiche amministrative diverse.

1. Aggiungere un utente

Eseguire la seguente chiamata API REST per aggiungere un utente ad Astra e indicare che LDAP è il provider di autenticazione.

Metodo HTTP	Percorso
POST	/account/{account_id}/core/v1/users

Esempio di input JSON

```
{
  "type" :  "application/astra-user",
  "version" : "1.1",
  "authID" : "cn=JohnDoe,ou=users,ou=astra,dc=example,dc=com",
  "authProvider" : "ldap",
  "firstName" : "John",
  "lastName" : "Doe",
  "email" : "john.doe@example.com"
}
```

Tenere presente quanto segue sui parametri di input:

- Sono necessari i seguenti parametri:
  - `authProvider`
  - `authID`
  - `email`
- `authID` È il nome distinto (DN) dell'utente in LDAP
- `email` Deve essere univoco per tutti gli utenti definiti in Astra

Se il `email` Il valore non è univoco, si verifica un errore e nella risposta viene restituito un codice di stato 409 HTTP.

Esempio di arricciamento

```
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>'
```

Esempio di risposta JSON


```
{
  "metadata": {
    "creationTimestamp": "2022-07-21T17:44:18Z",
    "modificationTimestamp": "2022-07-21T17:44:18Z",
    "createdBy": "8a02d2b8-a69d-4064-827f-36851b3e1e6e",
    "labels": []
  },
  "type": "application/astra-user",
  "version": "1.2",
  "id": "a7b5e674-a1b1-48f6-9729-6a571426d49f",
  "authProvider": "ldap",
  "authID": "cn=JohnDoe,ou=users,ou=astra,dc=example,dc=com",
  "firstName": "John",
  "lastName": "Doe",
  "companyName": "",
  "email": "john.doe@example.com",
  "postalAddress": {
    "addressCountry": "",
    "addressLocality": "",
    "addressRegion": "",
    "streetAddress1": "",
    "streetAddress2": "",
    "postalCode": ""
  },
  "state": "active",
  "sendWelcomeEmail": "false",
  "isEnabled": "true",
  "isInviteAccepted": "true",
  "enableTimestamp": "2022-07-21T17:44:18Z",
  "lastActTimestamp": ""
}
```


2. Aggiungere un’associazione di ruolo per l’utente

Eseguire la seguente chiamata API REST per associare l’utente a un ruolo specifico. È necessario creare l’UUID dell’utente nel passaggio precedente.

Metodo HTTP	Percorso
POST	/Account/{account_id}/core/v1/roleBindings

Esempio di input JSON

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

Tenere presente quanto segue sui parametri di input:

- Il valore utilizzato in precedenza per `roleConstraint` È l'unica opzione disponibile per la release corrente di Astra. Indica che l'utente non è limitato a un set limitato di spazi dei nomi e può accedervi tutti.

Esempio di arricciamento

```
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>'
```

Esempio di risposta JSON

```
{
  "metadata": {
    "creationTimestamp": "2022-07-21T18:08:24Z",
    "modificationTimestamp": "2022-07-21T18:08:24Z",
    "createdBy": "8a02d2b8-a69d-4064-827f-36851b3e1e6e",
    "labels": []
  },
  "type": "application/astra-roleBinding",
  "principalType": "user",
  "version": "1.1",
  "id": "b02c7e4d-d483-40d1-aaff-e1f900312114",
  "userID": "a7b5e674-a1b1-48f6-9729-6a571426d49f",
  "groupID": "00000000-0000-0000-0000-000000000000",
  "accountID": "d0fdbfa7-be32-4a71-b59d-13d95b42329a",
  "role": "member",
  "roleConstraints": ["*"]
}
```

Tenere presente quanto segue in merito ai parametri di risposta:

- Il valore `user` per `principalType` il campo indica l'aggiunta dell'associazione di ruoli per un utente (non un gruppo).

Aggiungere e associare un gruppo

È possibile assegnare un ruolo a un gruppo Astra che viene utilizzato dopo l'autenticazione LDAP. Ciò è appropriato quando vi è un numero elevato di utenti e ciascuno potrebbe avere caratteristiche amministrative simili.

1. Aggiungere un gruppo

Eseguire la seguente chiamata API REST per aggiungere un gruppo ad Astra e indicare che LDAP è il provider di autenticazione.

Metodo HTTP	Percorso
POST	/account/{account_id}/core/v1/groups

Esempio di input JSON

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

Tenere presente quanto segue sui parametri di input:

- Sono necessari i seguenti parametri:
  - `authProvider`
  - `authID`

Esempio di arricciamento

```
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>'
```

Esempio di risposta JSON

```
{
  "type": "application/astra-group",
  "version": "1.0",
  "id": "8b5b54da-ae53-497a-963d-1fc89990525b",
  "name": "Engineering",
  "authProvider": "ldap",
  "authID": "CN=Engineering,OU=groups,OU=astra,DC=example,DC=com",
  "metadata": {
    "creationTimestamp": "2022-07-21T18:42:52Z",
    "modificationTimestamp": "2022-07-21T18:42:52Z",
    "createdBy": "8a02d2b8-a69d-4064-827f-36851b3e1e6e",
    "labels": []
  }
}
```

2. Aggiungere un’associazione di ruolo per il gruppo

Eseguire la seguente chiamata API REST per associare il gruppo a un ruolo specifico. È necessario creare l’UUID del gruppo nel passaggio precedente. Gli utenti che sono membri del gruppo potranno accedere ad Astra dopo che LDAP ha eseguito l’autenticazione.

Metodo HTTP	Percorso
POST	/Account/{account_id}/core/v1/roleBindings

Esempio di input JSON

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

Tenere presente quanto segue sui parametri di input:

- Il valore utilizzato in precedenza per `roleConstraint` È l'unica opzione disponibile per la release corrente di Astra. Indica che l'utente non è limitato a determinati spazi dei nomi e può accedervi tutti.

Esempio di arricciamento

```
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>'
```

Esempio di risposta JSON

```
{
  "metadata": {
    "creationTimestamp": "2022-07-21T18:59:43Z",
    "modificationTimestamp": "2022-07-21T18:59:43Z",
    "createdBy": "527329f2-662c-41c0-ada9-2f428f14c137",
    "labels": []
  },
  "type": "application/astra-roleBinding",
  "principalType": "group",
  "version": "1.1",
  "id": "2f91b06d-315e-41d8-ae18-7df7c08fbb77",
  "userID": "00000000-0000-0000-0000-000000000000",
  "groupID": "8b5b54da-ae53-497a-963d-1fc89990525b",
  "accountID": "d0fdbfa7-be32-4a71-b59d-13d95b42329a",
  "role": "viewer",
  "roleConstraints": ["*"]
}
```

Tenere presente quanto segue in merito ai parametri di risposta:

- Il valore `group` per `principalType` il campo indica l'aggiunta dell'associazione di ruoli per un gruppo

(non per un utente).

Disattivare e ripristinare LDAP

Sono disponibili due attività amministrative opzionali, sebbene correlate, che è possibile eseguire in base alle necessità per un'implementazione di Astra Control Center. È possibile disattivare globalmente l'autenticazione LDAP e ripristinare la configurazione LDAP.

Entrambe le attività del flusso di lavoro richiedono l'id per `astra.account.ldap` Impostazione Astra. I dettagli su come recuperare l'id impostazione sono inclusi in **Configurazione del server LDAP**. Vedere "Recuperare l'UUID dell'impostazione LDAP" per ulteriori informazioni.

- "Disattiva autenticazione LDAP"
- "Ripristinare la configurazione di autenticazione LDAP"

Disattiva autenticazione LDAP

È possibile eseguire la seguente chiamata REST API per disattivare globalmente l'autenticazione LDAP per una specifica implementazione Astra. La chiamata aggiorna `astra.account.ldap` e il `isEnabled` il valore è impostato su `false`.

Metodo HTTP	Percorso
IN PRIMO PIANO	/account/{account_id}/core/v1/settings/{setting_id}

Esempio di input JSON

```
{
  "type": "application/astra-setting",
  "version": "1.0",
  "desiredConfig": {
    "connectionHost": "myldap.example.com",
    "credentialId": "3bd9c8a7-f5a4-4c44-b778-90a85fc7d154",
    "groupBaseDN": "OU=groups,OU=astra,DC=example,DC=com",
    "isEnabled": "false",
    "port": 686,
    "secureMode": "LDAPS",
    "userBaseDN": "OU=users,OU=astra,DC=example,dc=com",
    "userSearchFilter": "((objectClass=User))",
    "vendor": "Active Directory"
    }
}
```

```
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>'
```

Se la chiamata ha esito positivo, il `HTTP 204` la risposta viene restituita. È possibile recuperare di nuovo le impostazioni di configurazione per confermare la modifica.

Ripristinare la configurazione di autenticazione LDAP

È possibile eseguire la seguente chiamata REST API per disconnettere Astra dal server LDAP e reimpostare la configurazione LDAP in Astra. La chiamata aggiorna `astra.account.ldap` e il valore di `connectionHost` è deselezionato.

Il valore di `isEnabled` deve anche essere impostato su `false`. È possibile impostare questo valore prima di effettuare la chiamata di ripristino o come parte della chiamata di ripristino. Nel secondo caso, `connectionHost` devono essere cancellati e. `isEnabled` impostare su false per la stessa chiamata di ripristino.

> Si tratta di un'operazione di interruzione e si consiglia di procedere con cautela. Elimina tutti gli utenti e i gruppi LDAP importati. Inoltre, elimina tutti gli utenti, i gruppi e le associazioni di roleBinding Astra (tipo LDAP) creati in Astra Control Center.

Metodo HTTP	Percorso
IN PRIMO PIANO	/account/{account_id}/core/v1/settings/{setting_id}

Esempio di input JSON

```
{
  "type": "application/astra-setting",
  "version": "1.0",
  "desiredConfig": {
    "connectionHost": "",
    "credentialId": "3bd9c8a7-f5a4-4c44-b778-90a85fc7d154",
    "groupBaseDN": "OU=groups,OU=astra,DC=example,DC=com",
    "isEnabled": "false",
    "port": 686,
    "secureMode": "LDAPS",
    "userBaseDN": "OU=users,OU=astra,DC=example,dc=com",
    "userSearchFilter": "((objectClass=User))",
    "vendor": "Active Directory"
    }
}
```

Tenere presente quanto segue:

- Per modificare il server LDAP, è necessario disattivare e reimpostare LDAP Changing `connectHost` su un valore nullo come mostrato nell'esempio precedente.

```
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>'
```

Se la chiamata ha esito positivo, il `HTTP 204` la risposta viene restituita. È possibile recuperare nuovamente la configurazione per confermare la modifica.

Cluster

Elencare i cluster

È possibile elencare i cluster disponibili in un cloud specifico.

1. Selezionare il cloud

Eseguire il flusso di lavoro "Elencare i cloud" e seleziona il cloud che contiene i cluster.

2. Elencare i cluster

Eseguire la seguente chiamata API REST per elencare i cluster in un cloud specifico.

Metodo HTTP	Percorso
OTTIENI	/account/{account_id}/topology/v1/cloud/{cloud_id}/cluster

Esempio di curl: Restituisce tutti i dati per tutti i cluster

```
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>'
```

Esempio di output JSON

```
{
    "items": [
        {
            "type": "application/astra-cluster",
            "version": "1.1",
            "id": "7ce83fba-6aa1-4e0c-a194-26e714f5eb46",
            "name": "openshift-clstr-ol-07",
            "state": "running",
```

```
            "stateUnready": [],
            "managedState": "managed",
            "protectionState": "full",
            "protectionStateDetails": [],
            "restoreTargetSupported": "true",
            "snapshotSupported": "true",
            "managedStateUnready": [],
            "managedTimestamp": "2022-11-03T15:50:59Z",
            "inUse": "true",
            "clusterType": "openshift",
            "accHost": "true",
            "clusterVersion": "1.23",
            "clusterVersionString": "v1.23.12+6b34f32",
            "namespaces": [
                "default",
                "kube-node-lease",
                "kube-public",
                "kube-system",
                "metallb-system",
                "mysql",
                "mysql-clone1",
                "mysql-clone2",
                "mysql-clone3",
                "mysql-clone4",
                "netapp-acc-operator",
                "netapp-monitoring",
                "openshift",
                "openshift-apiserver",
                "openshift-apiserver-operator",
                "openshift-authentication",
                "openshift-authentication-operator",
                "openshift-cloud-controller-manager",
                "openshift-cloud-controller-manager-operator",
                "openshift-cloud-credential-operator",
                "openshift-cloud-network-config-controller",
                "openshift-cluster-csi-drivers",
                "openshift-cluster-machine-approver",
                "openshift-cluster-node-tuning-operator",
                "openshift-cluster-samples-operator",
                "openshift-cluster-storage-operator",
                "openshift-cluster-version",
                "openshift-config",
                "openshift-config-managed",
                "openshift-config-operator",
                "openshift-console",
                "openshift-console-operator",
```

```
                "openshift-console-user-settings",
                "openshift-controller-manager",
                "openshift-controller-manager-operator",
                "openshift-dns",
                "openshift-dns-operator",
                "openshift-etcd",
                "openshift-etcd-operator",
                "openshift-host-network",
                "openshift-image-registry",
                "openshift-infra",
                "openshift-ingress",
                "openshift-ingress-canary",
                "openshift-ingress-operator",
                "openshift-insights",
                "openshift-kni-infra",
                "openshift-kube-apiserver",
                "openshift-kube-apiserver-operator",
                "openshift-kube-controller-manager",
                "openshift-kube-controller-manager-operator",
                "openshift-kube-scheduler",
                "openshift-kube-scheduler-operator",
                "openshift-kube-storage-version-migrator",
                "openshift-kube-storage-version-migrator-operator",
                "openshift-machine-api",
                "openshift-machine-config-operator",
                "openshift-marketplace",
                "openshift-monitoring",
                "openshift-multus",
                "openshift-network-diagnostics",
                "openshift-network-operator",
                "openshift-node",
                "openshift-oauth-apiserver",
                "openshift-openstack-infra",
                "openshift-operator-lifecycle-manager",
                "openshift-operators",
                "openshift-ovirt-infra",
                "openshift-sdn",
                "openshift-service-ca",
                "openshift-service-ca-operator",
                "openshift-user-workload-monitoring",
                "openshift-vsphere-infra",
                "pcloud",
                "postgreql",
                "trident"
            ],
            "defaultStorageClass": "4bacbb3c-0727-4f58-b13c-3a2a069baf89",
```

```
            "cloudID": "4f1e1086-f415-4451-a051-c7299cd672ff",
            "credentialID": "7ffd7354-b6c2-4efa-8e7b-cf64d5598463",
            "isMultizonal": "false",
            "tridentManagedStateAllowed": [
                "unmanaged"
            ],
            "tridentVersion": "22.10.0",
            "apiServiceID": "98df44dc-2baf-40d5-8826-e198b1b40909",
            "metadata": {
                "labels": [
                    {
                        "name": "astra.netapp.io/labels/read-
only/cloudName",
                        "value": "private"
                    }
                ],
                "creationTimestamp": "2022-11-03T15:50:59Z",
                "modificationTimestamp": "2022-11-04T14:42:32Z",
                "createdBy": "00000000-0000-0000-0000-000000000000"
            }
        }
    ]
}
```


Elencare i cluster gestiti

Puoi elencare i cluster Kubernetes attualmente gestiti da Astra.

1. Elencare i cluster gestiti

Eseguire la seguente chiamata API REST.

Metodo HTTP	Percorso
OTTIENI	/Account/{account_id}/topology/v1/managedClusters

Esempio di curl: Restituisce tutti i dati per tutti i cluster

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

Cloud

Elencare i cloud

Puoi elencare i cloud definiti e disponibili per un account Astra specifico.

1. Elencare i cloud

Eseguire la seguente chiamata API REST per elencare i cloud.

Metodo HTTP	Percorso
OTTIENI	/account/{account_id}/topology/v1/cloud

Esempio di curl: Restituisce tutti i dati per tutti i cloud

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

Bucket

Elencare i bucket

È possibile elencare i bucket S3 definiti per un account Astra specifico.

1. Elencare i bucket

Eseguire la seguente chiamata API REST per elencare i bucket.

Metodo HTTP	Percorso
OTTIENI	/account/{account_id}/topology/v1/bucket

Esempio di curl: Restituisce tutti i dati per tutti i bucket

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

Storage

Elencare le classi di storage

È possibile elencare le classi di storage disponibili.

1. Selezionare il cloud

Eseguire il flusso di lavoro "Elencare i cloud" e seleziona il cloud in cui lavorerai.

2. Selezionare il cluster

Eseguire il flusso di lavoro "Elencare i cluster" e selezionare il cluster.

3. Elencare le classi di storage per un cluster specifico

Eseguire la seguente chiamata API REST per elencare le classi di storage per un cluster e un cloud specifici.

Metodo HTTP	Percorso
OTTIENI	/Account/{account_id}/topology/v1/cloud/<CLOUD_ID>/Clusters/<CLUSTER_ID>/storageClasses

Esempio di curl: Restituisce tutti i dati per tutte le classi di storage

```
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>'
```

Esempio di output JSON


```
{
    "items": [
        {
            "type": "application/astra-storageClass",
            "version": "1.1",
            "id": "4bacbb3c-0727-4f58-b13c-3a2a069baf89",
            "name": "ontap-basic",
            "provisioner": "csi.trident.netapp.io",
            "available": "eligible",
            "allowVolumeExpansion": "true",
            "reclaimPolicy": "Delete",
            "volumeBindingMode": "Immediate",
            "isDefault": "true",
            "metadata": {
                "createdBy": "system",
                "creationTimestamp": "2022-10-26T05:16:19Z",
                "modificationTimestamp": "2022-10-26T05:16:19Z",
                "labels": []
            }
        },
        {
            "type": "application/astra-storageClass",
            "version": "1.1",
            "id": "150fe657-4a42-47a3-abc6-5dafba3de8bf",
            "name": "thin",
```

```
            "provisioner": "kubernetes.io/vsphere-volume",
            "available": "ineligible",
            "reclaimPolicy": "Delete",
            "volumeBindingMode": "Immediate",
            "metadata": {
                "createdBy": "system",
                "creationTimestamp": "2022-10-26T04:46:08Z",
                "modificationTimestamp": "2022-11-04T14:58:19Z",
                "labels": []
            }
        },
        {
            "type": "application/astra-storageClass",
            "version": "1.1",
            "id": "7c6a5c58-6a0d-4cb6-98a0-8202ad2de74a",
            "name": "thin-csi",
            "provisioner": "csi.vsphere.vmware.com",
            "available": "ineligible",
            "allowVolumeExpansion": "true",
            "reclaimPolicy": "Delete",
            "volumeBindingMode": "WaitForFirstConsumer",
            "metadata": {
                "createdBy": "system",
                "creationTimestamp": "2022-10-26T04:46:17Z",
                "modificationTimestamp": "2022-10-26T04:46:17Z",
                "labels": []
            }
        },
        {
            "type": "application/astra-storageClass",
            "version": "1.1",
            "id": "7010ef09-92a5-4c90-a5e5-3118e02dc9a7",
            "name": "vsim-san",
            "provisioner": "csi.trident.netapp.io",
            "available": "eligible",
            "allowVolumeExpansion": "true",
            "reclaimPolicy": "Delete",
            "volumeBindingMode": "Immediate",
            "metadata": {
                "createdBy": "system",
                "creationTimestamp": "2022-11-03T18:40:03Z",
                "modificationTimestamp": "2022-11-03T18:40:03Z",
                "labels": []
            }
        }
    ]
```

```
}
```

Elenca i backend di storage

È possibile elencare i backend di storage disponibili.

1. Elencare i backend

Eseguire la seguente chiamata API REST.

Metodo HTTP	Percorso
OTTIENI	/Account/{account_id}/topology/v1/storageBackends

Esempio di curl: Restituisce tutti i dati per tutti i backend di storage

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

Esempio di output JSON

```
{
  "items": [
    {
      "backendCredentialsName": "10.191.77.177",
      "backendName": "myinchunhcluster-1",
      "backendType": "ONTAP",
      "backendVersion": "9.8.0",
      "configVersion": "Not applicable",
      "health": "Not applicable",
      "id": "46467c16-1585-4b71-8e7f-f0bc5ff9da15",
      "location": "nalab2",
      "metadata": {
        "createdBy": "4c483a7e-207b-4f9a-87b7-799a4629d7c8",
        "creationTimestamp": "2021-07-30T14:26:19Z",
        "modificationTimestamp": "2021-07-30T14:26:19Z"
      },
      "ontap": {
        "backendManagementIP": "10.191.77.177",
        "managementIPs": [
          "10.191.77.177",
          "10.191.77.179"
        ]
      },
      "protectionPolicy": "Not applicable",
      "region": "Not applicable",
      "state": "Running",
      "stateUnready": [],
      "type": "application/astra-storageBackend",
      "version": "1.0",
      "zone": "Not applicable"
    }
  ]
}
```

Workflow di gestione

Prima di iniziare

È possibile utilizzare questi flussi di lavoro come parte dell'amministrazione delle applicazioni in un cluster gestito da Astra.

> Questi flussi di lavoro possono essere ampliati e migliorati da NetApp in qualsiasi momento, pertanto è necessario esaminarli periodicamente.

Preparazione generale

Prima di utilizzare uno qualsiasi dei flussi di lavoro Astra, assicurarsi di rivedere "Preparati a utilizzare i flussi di lavoro".

Categorie di workflow

I flussi di lavoro di gestione sono organizzati in diverse categorie per facilitare l'individuazione di quello desiderato.

Categoria	Descrizione
Controllo delle applicazioni	Questi flussi di lavoro consentono di controllare le applicazioni gestite e non gestite. È possibile elencare le applicazioni, nonché creare e rimuovere un'applicazione gestita.
Protezione dell'applicazione	È possibile utilizzare questi flussi di lavoro per proteggere le applicazioni gestite attraverso snapshot e backup.
Clonare e ripristinare le applicazioni	Questo flusso di lavoro descrive come clonare e ripristinare le applicazioni gestite.
Supporto	Sono disponibili diversi flussi di lavoro per il debug e il supporto delle applicazioni, oltre all'ambiente Kubernetes generale.

Considerazioni aggiuntive

Quando si utilizzano i flussi di lavoro di gestione, è necessario considerare alcune considerazioni aggiuntive.

Clonare un'applicazione

Quando si clonano un'applicazione, è necessario prendere in considerazione alcuni aspetti. I parametri descritti di seguito fanno parte dell'input JSON.

Identificatore del cluster di origine

Il valore di `sourceClusterID` identifica sempre il cluster in cui è installata l'applicazione originale.

Identificatore del cluster

Il valore di `clusterID` identifica il cluster in cui verrà installata la nuova applicazione.

- Durante la clonazione all'interno dello stesso cluster, `clusterID` e. `sourceClusterID` hanno lo stesso valore.

- Quando si esegue la clonazione tra cluster, i due valori sono diversi e. `clusterID` Deve essere l'ID del cluster di destinazione.

Spazi dei nomi

Il `namespace` il valore deve essere diverso dall'applicazione di origine. Inoltre, lo spazio dei nomi per il clone non può esistere e Astra lo crea.

Backup e snapshot

È possibile clonare un'applicazione da un backup o da uno snapshot esistente utilizzando `backupID` oppure `snapshotID` parametri. Se non si fornisce un backup o uno snapshot, Astra crea prima un backup dell'applicazione e poi clonerà dal backup.

Ripristino di un'applicazione

Di seguito sono riportati alcuni aspetti da considerare durante il ripristino di un'applicazione.

- Il ripristino di un'applicazione è molto simile all'operazione di clonazione.
- Durante il ripristino di un'applicazione, è necessario fornire un backup o uno snapshot.

Controllo dell'app

Elencare le applicazioni

È possibile elencare le applicazioni attualmente gestite da Astra. È possibile eseguire questa operazione nell'ambito della ricerca di snapshot o backup per un'applicazione specifica.

1. Elencare le applicazioni

Eseguire la seguente chiamata API REST.

Metodo HTTP	Percorso
OTTIENI	/account/{account_id}/k8s/v2/apps

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
includi	Query	No	Se si desidera, selezionare i valori che si desidera restituire nella risposta.

Esempio di curl: Restituisce tutti i dati per tutte le applicazioni

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Esempio di curl: Restituisce il nome, l'id e lo stato per tutte le applicazioni

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps?include=name,id
,state' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Esempio di output JSON

```
{
    "items": [
        [
            "mysql",
            "4ee2b8fa-3696-4f32-8879-399792f477c3",
            "ready"
        ],
        [
            "postgresql",
            "3b984474-e5c9-4b64-97ee-cdeb9bcd212e",
            "ready"
        ],
    ],
    "metadata": {}
}
```

Scarica un'app

È possibile recuperare tutte le variabili delle risorse che descrivono una singola applicazione.

Prima di iniziare

È necessario disporre dell'ID dell'applicazione che si desidera recuperare. Se necessario, è possibile utilizzare il flusso di lavoro "Elencare le applicazioni" per individuare l'applicazione.

1. Scarica l'applicazione

Eseguire la seguente chiamata API REST.

Metodo HTTP	Percorso
OTTIENI	/accounts/{account_id}/k8s/v2/apps/{app_id}

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
id app	Percorso	Sì	Valore ID dell'applicazione da recuperare.

Esempio di curl: Restituisce tutti i dati per l'applicazione

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps/<APP_ID>'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Gestire un'applicazione

È possibile creare un'applicazione gestita in base a un'applicazione già nota ad Astra in uno spazio dei nomi specifico. Quando un'applicazione viene gestita o definita in Astra, è possibile proteggerla eseguendo backup e snapshot.

1. Selezionare lo spazio dei nomi

Eseguire il flusso di lavoro "Elencare gli spazi dei nomi" e selezionare lo spazio dei nomi.

2. Selezionare il cluster

Eseguire il flusso di lavoro "Elencare i cluster" e selezionare il cluster.

3. Gestire l'applicazione

Eseguire la seguente chiamata API REST per gestire l'applicazione.

Metodo HTTP	Percorso
POST	/account/{account_id}/k8s/v2/apps

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
JSON	Corpo	Sì	Fornisce i parametri necessari per identificare l'applicazione da gestire. Vedere l'esempio riportato di seguito.

Esempio di input JSON

```
{
  "clusterID": "7ce83fba-6aa1-4e0c-a194-26e714f5eb46",
  "name": "subtext",
  "namespaceScopedResources": [{"namespace": "kube-matrix"}],
  "type": "application/astra-app",
  "version": "2.0"
}
```

Esempio di curl: Gestire un'applicazione

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Content-Type: application/astra-app+json' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>'  --data @JSONinput
```

Annullare la gestione di un'applicazione

Puoi rimuovere un'applicazione gestita quando non è più necessaria. La rimozione di un'applicazione gestita elimina anche le pianificazioni associate.

Prima di iniziare

Devi disporre dell'ID dell'applicazione che desideri annullare la gestione. Se necessario, è possibile utilizzare il flusso di lavoro "Elencare le applicazioni" per individuare l'applicazione.

I backup e le snapshot dell'applicazione non vengono rimossi automaticamente quando vengono eliminati. Se non sono più necessari backup e snapshot, è necessario eliminarli prima di rimuovere l'applicazione.

1. Applicazione non gestita

Eseguire la seguente chiamata API REST per rimuovere l'applicazione.

Metodo HTTP	Percorso
ELIMINARE	/accounts/{account_id}/k8s/v2/apps/{app_id}

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
id app	Percorso	Sì	Identifica l'applicazione da rimuovere.

Esempio di curl: Rimuovere un'applicazione gestita

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps/<APP_ID>'
--header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Protezione delle applicazioni

Elencare le istantanee

È possibile elencare le istantanee acquisite per un'applicazione specifica.

Prima di iniziare

È necessario disporre dell'ID dell'applicazione per la quale si desidera elencare le snapshot. Se necessario, è possibile utilizzare il flusso di lavoro "Elencare le applicazioni" per individuare l'applicazione.

1. Elencare le istantanee

Eseguire la seguente chiamata API REST per elencare le snapshot.

Metodo HTTP	Percorso
OTTIENI	/Accounts/{account_id}/k8s/v1/apps/{app_id}/appSnap

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
id app	Percorso	Sì	Identifica l'applicazione proprietaria delle istantanee elencate.
conta	Query	No	Se `count=true` il numero di snapshot è incluso nella sezione dei metadati della risposta.

Esempio di curl: Restituire tutte le snapshot per l'applicazione

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appSna
ps' --header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Esempio di curl: Restituisce tutte le snapshot per l'applicazione e il numero

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appSna
ps?count=true' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Esempio di output JSON


```
{
    "items": [
        {
            "type": "application/astra-appSnap",
            "version": "1.1",
            "id": "1ce34da4-bb0a-4926-b925-4a5d85dda8c2",
            "hookState": "success",
            "metadata": {
                "createdBy": "a530e865-23e8-4e2e-8020-e92c419a3867",
                "creationTimestamp": "2022-10-30T22:44:20Z",
                "modificationTimestamp": "2022-10-30T22:44:20Z",
                "labels": []
            },
            "snapshotAppAsset": "0ebfe3f8-40ed-4bdc-88c4-2144fbda85a0",
            "snapshotCreationTimestamp": "2022-10-30T22:44:33Z",
            "name": "snapshot-david-1",
            "state": "completed",
            "stateUnready": []
        }
    ],
    "metadata": {}
}
```


Elencare i backup

È possibile elencare i backup creati per un'applicazione specifica.

Prima di iniziare

È necessario disporre dell'ID dell'applicazione per cui si desidera elencare i backup. Se necessario, è possibile utilizzare il flusso di lavoro "Elencare le applicazioni" per individuare l'applicazione.

1. Elencare i backup

Eseguire la seguente chiamata API REST.

Metodo HTTP	Percorso
OTTIENI	/Accounts/{account_id}/k8s/v1/apps/{app_id}/appBackups

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
id app	Percorso	Sì	Identifica l'applicazione gestita proprietaria dei backup elencati.

Esempio di curl: Restituire tutti i backup per l'applicazione

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appBac
kups' --header 'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>'
```

Esempio di output JSON

```
{
    "items": [
        {
            "type": "application/astra-appBackup",
            "version": "1.1",
            "id": "8edeb4a4-fd8b-4222-a559-1013145b28fc",
            "name": "backup-david-oct28-1",
            "bucketID": "a443e58f-59bd-4d45-835a-1bc7813f659a",
            "snapshotID": "dfe237cb-57b7-4576-af4d-00ba3a8f2828",
            "state": "completed",
            "stateUnready": [],
            "hookState": "success",
            "totalBytes": 205219132,
            "bytesDone": 205219132,
            "percentDone": 100,
            "metadata": {
                "labels": [
                    {
                        "name": "astra.netapp.io/labels/read-
only/triggerType",
                        "value": "backup"
                    }
                ],
                "creationTimestamp": "2022-10-28T21:58:37Z",
                "modificationTimestamp": "2022-10-28T21:58:55Z",
                "createdBy": "a530e865-23e8-4e2e-8020-e92c419a3867"
            }
        }
    ],
    "metadata": {}
}
```


Creare un’istantanea per un’applicazione

È possibile creare uno snapshot per un’applicazione specifica.

Prima di iniziare

È necessario disporre dell’ID dell’applicazione per la quale si desidera creare uno snapshot. Se necessario, è possibile utilizzare il flusso di lavoro "Elencare le applicazioni" per individuare l’applicazione.

1. Creare un’istantanea

Eseguire la seguente chiamata API REST.

Metodo HTTP	Percorso
POST	/Accounts/{account_id}/k8s/v1/apps/{app_id}/appSnap

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
id app	Percorso	Sì	Identifica l'applicazione gestita in cui verrà creata l'istantanea.
JSON	Corpo	Sì	Fornisce i parametri per lo snapshot. Vedere l'esempio riportato di seguito.

Esempio di input JSON

```
{
  "type": "application/astra-appSnap",
  "version": "1.1",
  "name": "snapshot-david-1"
}
```

Esempio di curl: Creare un'istantanea per l'applicazione

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appSna
ps' --header 'Content-Type: application/astra-appSnap+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --data
@JSONinput
```

Creare un backup per un'applicazione

È possibile creare un backup per un'applicazione specifica e utilizzarlo per ripristinare o clonare l'applicazione.

Prima di iniziare

Devi disporre dell'ID dell'applicazione di cui desideri eseguire il backup. Se necessario, è possibile utilizzare il flusso di lavoro "Elencare le applicazioni" per individuare l'applicazione.

1. Creare un backup

Eseguire la seguente chiamata API REST.

Metodo HTTP	Percorso
POST	/Accounts/{account_id}/k8s/v1/apps/{app_id}/appBackups

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
id app	Percorso	Sì	Identifica l'applicazione in cui verrà creato il backup.
JSON	Corpo	Sì	Fornisce i parametri per il backup. Vedere l'esempio riportato di seguito.

Esempio di input JSON

```
{
  "type": "application/astra-appBackup",
  "version": "1.1",
  "name": "backup-david-1"
}
```

Esempio di curl: Creare un backup per l'applicazione

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appBac
kups' --header 'Content-Type: application/astra-appBackup+json' --header
'Accept: */*' --header 'Authorization: Bearer <API_TOKEN>' --data
@JSONinput
```

Eliminare uno snapshot

È possibile eliminare uno snapshot associato a un'applicazione.

Prima di iniziare

È necessario disporre di quanto segue:

- ID dell'applicazione proprietaria dello snapshot. Se necessario, è possibile utilizzare il flusso di lavoro "Elencare le applicazioni" per individuare l'applicazione.
- ID dello snapshot che si desidera eliminare. Se necessario, è possibile utilizzare il flusso di lavoro "Elencare le istantanee" per individuare lo snapshot.

1. Eliminare l'istantanea

Eseguire la seguente chiamata API REST.

Metodo HTTP	Percorso
ELIMINARE	/Accounts/{account_id}/k8s/v1/apps/{app_id}/appSnap/{appSnap_id}

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
id app	Percorso	Sì	Identifica l'applicazione gestita proprietaria dello snapshot.
id snapshot	Percorso	Sì	Identifica lo snapshot da eliminare.

Esempio di curl: Eliminare una singola istantanea per l'applicazione

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appSna
ps/<SNAPSHOT_ID>' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Eliminare un backup

È possibile eliminare un backup associato a un'applicazione.

Prima di iniziare

È necessario disporre di quanto segue:

- ID dell'applicazione proprietaria del backup. Se necessario, è possibile utilizzare il flusso di lavoro "Elencare le applicazioni" per individuare l'applicazione.
- ID del backup che si desidera eliminare. Se necessario, è possibile utilizzare il flusso di lavoro "Elencare i backup" per individuare lo snapshot.

1. Eliminare il backup

Eseguire la seguente chiamata API REST.

> È possibile forzare l'eliminazione di un backup non riuscito utilizzando l'intestazione della richiesta opzionale come descritto di seguito.

Metodo HTTP	Percorso
ELIMINARE	/Accounts/{account_id}/k8s/v1/apps/{app_id}/appBackups/{appBackup_id}

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
id app	Percorso	Sì	Identifica l'applicazione gestita proprietaria del backup.
id backup	Percorso	Sì	Identifica il backup da eliminare.
forza eliminazione	Intestazione	No	Utilizzato per forzare l'eliminazione di un backup non riuscito.

Esempio di curl: Eliminare un singolo backup per l'applicazione

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appBac
kups/<BACKUP_ID>' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>'
```

Esempio di curl: Eliminare un singolo backup per l'applicazione con l'opzione force

```
curl --location -i --request DELETE
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v1/apps/<APP_ID>/appBac
kups/<BACKUP_ID>' --header 'Accept: */*' --header 'Authorization: Bearer
<API_TOKEN>' --header 'Force-Delete: true'
```

Clonare e ripristinare un'applicazione

Clonare un'applicazione

È possibile creare una nuova applicazione clonando un'applicazione esistente.

Prima di iniziare

Tenere presente quanto segue a proposito di questo flusso di lavoro:

- Non viene utilizzato un backup o uno snapshot dell'applicazione
- L'operazione di cloni viene eseguita all'interno dello stesso cluster
- La nuova applicazione viene inserita in uno spazio dei nomi diverso

> Per clonare un'applicazione in un cluster diverso, è necessario aggiornare `clusterId` Nell'input JSON appropriato per il proprio ambiente.

1. Selezionare l'applicazione da clonare

Eseguire il flusso di lavoro "Elencare le applicazioni" e selezionare l'applicazione che si desidera clonare. Per la chiamata DI PAUSA utilizzata per clonare l'applicazione sono necessari diversi valori delle risorse.

2. Clonare l'applicazione

Eseguire la seguente chiamata API REST per clonare l'applicazione.

Metodo HTTP	Percorso
POST	/account/{account_id}/k8s/v2/apps

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
JSON	Corpo	Sì	Fornisce i parametri per l'applicazione clonata. Vedere l'esempio riportato di seguito.

Esempio di input JSON

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "name": "mysql-clone",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "mysql-ns",
  "sourceAppID": "e591ee59-ea90-4a9f-8e6c-d2b6e8647096"
}
```

Esempio di curl: Clonare un'applicazione

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Content-Type: application/astra-app+json' --header '*/*' --header
'Authorization: Bearer <API_TOKEN>' --data @JSONinput
```

Clonare un'applicazione da uno snapshot

È possibile creare una nuova applicazione clonandola da uno snapshot.

Prima di iniziare

Tenere presente quanto segue a proposito di questo flusso di lavoro:

- Viene utilizzata un'istantanea dell'applicazione
- L'operazione di cloni viene eseguita all'interno dello stesso cluster

> Per clonare un'applicazione in un cluster diverso, è necessario aggiornare `clusterId` Nell'input JSON appropriato per il proprio ambiente.

1. Selezionare l'applicazione da clonare

Eseguire il flusso di lavoro "Elencare le applicazioni" e selezionare l'applicazione che si desidera clonare. Per la chiamata DI PAUSA utilizzata per clonare l'applicazione sono necessari diversi valori delle risorse.

2. Selezionare l'istantanea da utilizzare

Eseguire il flusso di lavoro "Elencare le istantanee" e selezionare lo snapshot da utilizzare.

3. Clonare l'applicazione

Eseguire la seguente chiamata API REST.

Metodo HTTP	Percorso
POST	/account/{account_id}/k8s/v2/apps

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
JSON	Corpo	Sì	Fornisce i parametri per l'applicazione clonata. Vedere l'esempio riportato di seguito.

Esempio di input JSON

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "name": "mysql-clone2",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "mysql",
  "snapshotID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"
}
```

Esempio di curl: Clonare un'applicazione da uno snapshot

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Content-Type: application/astra-app+json' --header '*/*' --header
'Authorization: Bearer <API_TOKEN>' --data @JSONinput
```

Clonare un'applicazione da un backup

È possibile creare una nuova applicazione clonandola da un backup.

Prima di iniziare

Tenere presente quanto segue a proposito di questo flusso di lavoro:

- Viene utilizzato un backup dell'applicazione
- L'operazione di cloni viene eseguita all'interno dello stesso cluster

> Per clonare un'applicazione in un cluster diverso, è necessario aggiornare `clusterId` Nell'input JSON appropriato per il proprio ambiente.

1. Selezionare l'applicazione da clonare

Eseguire il flusso di lavoro "Elencare le applicazioni" e selezionare l'applicazione che si desidera clonare. Per la chiamata DI PAUSA utilizzata per clonare l'applicazione sono necessari diversi valori delle risorse.

2. Selezionare il backup da utilizzare

Eseguire il flusso di lavoro "Elencare i backup" e selezionare il backup che si desidera utilizzare.

3. Clonare l'applicazione

Eseguire la seguente chiamata API REST.

Metodo HTTP	Percorso
POST	/account/{account_id}/k8s/v2/qpps

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
JSON	Corpo	Sì	Fornisce i parametri per l'applicazione clonata. Vedere l'esempio riportato di seguito.

Esempio di input JSON

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "name": "mysql-clone3",
  "clusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "sourceClusterID": "30880586-d579-4d27-930f-a9633e59173b",
  "namespace": "mysql",
  "backupID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"
}
```

Esempio di curl: Clonare un'applicazione da un backup

```
curl --location -i --request POST
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps' --header
'Content-Type: application/astra-app+json' --header '*/*' --header
'Authorization: Bearer <API_TOKEN>' --data @JSONinput
```

Ripristinare un'applicazione da un backup

È possibile ripristinare un'applicazione creando una nuova applicazione da un backup.

1. Selezionare l'applicazione da ripristinare

Eseguire il flusso di lavoro "Elencare le applicazioni" e selezionare l'applicazione che si desidera clonare. Per la chiamata DI PAUSA utilizzata per ripristinare l'applicazione sono necessari diversi valori di risorse.

2. Selezionare il backup da utilizzare

Eseguire il flusso di lavoro "Elencare i backup" e selezionare il backup che si desidera utilizzare.

3. Ripristinare l'applicazione

Eseguire la seguente chiamata API REST. È necessario fornire l'ID per un backup (come mostrato di seguito) o uno snapshot.

Metodo HTTP	Percorso
IN PRIMO PIANO	/account/{account_id}/k8s/v2/apps/{app_id}

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
JSON	Corpo	Sì	Fornisce i parametri per l'applicazione clonata. Vedere l'esempio riportato di seguito.

Esempio di input JSON

```
{
  "type": "application/astra-app",
  "version": "2.0",
  "backupID": "e24515bd-a28e-4b28-b832-f3c74dbf32fb"
}
```

Esempio di curl: Ripristinare un'applicazione in uso da un backup

```
curl --location -i --request PUT
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/k8s/v2/apps/<APP_ID>'
--header 'Content-Type: application/astra-app+json' --header '*/*'
--header 'ForceUpdate: true' --header 'Authorization: Bearer <API_TOKEN>'
--data @JSONinput
```

Spazi dei nomi

Elencare gli spazi dei nomi

È possibile elencare gli spazi dei nomi disponibili.

1. Elencare gli spazi dei nomi

Eseguire la seguente chiamata API REST per elencare gli spazi dei nomi.

Metodo HTTP	Percorso
OTTIENI	/account/{account_id}/topology/v1/namespaces

Esempio di curl: Restituisce tutti i dati per tutti gli spazi dei nomi

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

Esempio di curl: Restituisce nome, stato e ID cluster per tutti gli spazi dei nomi

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/topology/v1/namespaces?incl
ude=name,namespaceState,clusterID' --header 'Accept: */*' --header
'Authorization: Bearer <API_TOKEN>'
```

Esempio di output JSON


```
{
    "items": [
        [
            "default",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "kube-node-lease",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "kube-public",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "kube-system",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "mysql",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "mysql-clone1",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "netapp-acc-operator",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
```

```
        [
            "openshift",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ],
        [
            "trident",
            "discovered",
            "922f924a-a476-4a79-97f6-472571698154"
        ]
    ],
    "metadata": {}
}
```

Supporto

Elencare le notifiche

Puoi elencare le notifiche per un account Astra specifico. Questa operazione potrebbe essere eseguita durante il monitoraggio dell'attività del sistema o il debug di un problema.

1. Elencare le notifiche

Eseguire la seguente chiamata API REST.

Metodo HTTP	Percorso
OTTIENI	/account/{account_id}/core/v1/notifications

Parametri di input aggiuntivi

Oltre ai parametri comuni a tutte le chiamate API REST, negli esempi di curl vengono utilizzati anche i seguenti parametri.

Parametro	Tipo	Obbligatorio	Descrizione
filtro	Query	No	Se si desidera, filtrare le notifiche che si desidera restituire nella risposta.
includi	Query	No	Se si desidera, selezionare i valori che si desidera restituire nella risposta.

Esempio di curl: Restituisce tutte le notifiche

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

Esempio di curl: Restituisce la descrizione delle notifiche con severità di avviso

```
curl --location -i --request GET
'https://astra.netapp.io/accounts/<ACCOUNT_ID>/core/v1/notifications?filte
r=severity%20eq%20'warning'&include=description' --header 'Accept: */*'
--header 'Authorization: Bearer <API_TOKEN>'
```

Esempio di output JSON

```
{
  "items": [
    [
      "Trident on cluster david-ie-00 has failed or timed out;
installation of the Trident operator failed or is not yet complete;
operator failed to reach an installed state within 300.00 seconds;
container trident-operator not found in operator deployment"
    ],
    [
      "Trident on cluster david-ie-00 has failed or timed out;
installation of the Trident operator failed or is not yet complete;
operator failed to reach an installed state within 300.00 seconds;
container trident-operator not found in operator deployment"
    ]
  ],
  "metadata": {}
}
```

Eliminare un’applicazione non riuscita

Potrebbe non essere possibile rimuovere un’applicazione gestita in caso di backup o snapshot in stato di errore. In questo caso, puoi rimuovere manualmente l’applicazione utilizzando il flusso di lavoro descritto di seguito.

1. Selezionare l’applicazione da eliminare

Eseguire il flusso di lavoro "Elencare le applicazioni" e selezionare l’applicazione che si desidera rimuovere.

2. Elencare i backup esistenti per l’applicazione

Eseguire il flusso di lavoro "Elencare i backup".

3. Eliminare tutti i backup

Eliminare tutti i backup delle applicazioni eseguendo il workflow "Eliminare un backup" per ogni backup nell’elenco.

4. Elencare le snapshot esistenti per l’applicazione

Eseguire il flusso di lavoro "Elencare le istantanee".

5. Eliminare tutte le istantanee

Eseguire il flusso di lavoro "Eliminare uno snapshot" da ogni snapshot nell’elenco.

6. Rimuovere l’applicazione

Eseguire il flusso di lavoro "Annullare la gestione di un’applicazione" per rimuovere l’applicazione.

Utilizzo di Python

SDK NetApp Astra Control Python

NetApp Astra Control Python SDK è un pacchetto open source che puoi utilizzare per automatizzare un'implementazione di Astra Control. Il pacchetto è anche una risorsa preziosa per imparare a conoscere l'API REST di Astra Control, magari come parte della creazione della tua piattaforma di automazione.

> Per semplicità, NetApp Astra Control Python SDK verrà indicato come **SDK** nella parte restante di questa pagina.

Due tool software correlati

L'SDK include due tool diversi, sebbene correlati, che operano a diversi livelli di astrazione quando si accede all'API REST di Astra Control.

SDK Astra

Astra SDK offre le funzionalità principali della piattaforma. Include un insieme di classi Python che astraggono le chiamate API REST sottostanti. Le classi supportano azioni amministrative su varie risorse di Astra Control, tra cui app, backup, snapshot e cluster.

Astra SDK è una parte del pacchetto e viene fornito nel singolo `astraSDK.py` file. È possibile importare questo file nel proprio ambiente e utilizzare direttamente le classi.

> L'SDK * NetApp Astra Control Python (o solo SDK) è il nome dell'intero pacchetto. L'SDK * Astra si riferisce alle classi Python principali nel singolo file `astraSDK.py`.

Script del toolkit

Oltre al file Astra SDK, il `toolkit.py` è disponibile anche uno script. Questo script opera a un livello di astrazione superiore fornendo l'accesso a azioni amministrative discrete definite internamente come funzioni Python. Lo script importa l'SDK Astra ed effettua chiamate alle classi in base alle necessità.

Come accedere

È possibile accedere all'SDK nei seguenti modi.

Pacchetto Python

L'SDK è disponibile all'indirizzo "Python Package Index" sotto il nome **actoolkit**. Al pacchetto viene assegnato un numero di versione e continuerà ad essere aggiornato in base alle necessità. Per installare il pacchetto nel proprio ambiente, è necessario utilizzare l'utility di gestione dei pacchetti **PIP**.

Una volta installate, le astraSDK.py classi possono essere utilizzate collocando `import astraSDK` negli script. Inoltre, `actoolkit` può essere richiamato direttamente dal prompt dei comandi ed è equivalente a. `toolkit.py` (`actoolkit list clusters` è uguale a. `./toolkit.py list clusters`).

Vedere "PyPI: SDK NetApp Astra Control Python" per ulteriori informazioni.

Codice sorgente di GitHub

Il codice sorgente dell'SDK è disponibile anche su GitHub. Il repository include quanto segue:

- `astraSDK.py` (SDK Astra con classi Python)
- `toolkit.py` (script basato sulle funzioni di livello superiore)
- Istruzioni e requisiti di installazione dettagliati
- Script di installazione
- Documentazione aggiuntiva

È possibile clonare "GitHub: NetApp/netapp-astra-toolkit" repository nel tuo ambiente locale.

Installazione e requisiti di base

Esistono diverse opzioni e requisiti da prendere in considerazione durante l'installazione del pacchetto e la preparazione per l'utilizzo.

Riepilogo delle opzioni di installazione

È possibile installare l'SDK in uno dei seguenti modi:

- Utilizzare il preparato "Docker: NetApp/astra-toolkit" immagine, che ha tutte le dipendenze necessarie installate, tra cui `actoolkit`
- Utilizzare PIP per installare `actoolkit` Pacchetto da PyPI nel tuo ambiente Python
- Clonare il repository di GitHub e copiare/modificare i due file Python principali in modo che siano accessibili al codice client Python

Per ulteriori informazioni, fare riferimento alle pagine PyPI e GitHub.

Requisiti per l'ambiente Astra Control

Sia che si utilizzi direttamente le classi Python nell'SDK Astra o le funzioni in `toolkit.py` Script, in ultima analisi, potrai accedere all'API REST in un'implementazione di Astra Control. Per questo motivo, avrai bisogno di un account Astra con un token API. Vedere "Prima di iniziare" E le altre pagine della sezione **Get Started** di questa documentazione per ulteriori informazioni.

Requisiti per NetApp Astra Control Python SDK

L'SDK ha diversi prerequisiti relativi all'ambiente Python locale. Ad esempio, è necessario utilizzare Python 3.8 o versione successiva. Inoltre, sono necessari diversi pacchetti Python. Per ulteriori informazioni, consulta la pagina del repository GitHub o la pagina del pacchetto PyPI.

Riepilogo delle risorse utili

Ecco alcune delle risorse necessarie per iniziare.

- "PyPI: SDK NetApp Astra Control Python"
- "GitHub: NetApp/netapp-astra-toolkit"
- "Docker: NetApp/astra-toolkit"

Python nativo

Prima di iniziare

Python è un popolare linguaggio di sviluppo per l'automazione dei data center. Prima di utilizzare le funzionalità native di Python insieme a diversi pacchetti comuni, è necessario preparare l'ambiente e i file di input richiesti.

> Oltre ad accedere direttamente all'API REST di Astra Control utilizzando Python, NetApp fornisce anche un pacchetto di toolkit che astratta l'API e rimuove alcune delle complessità. Vedere "SDK NetApp Astra Control Python" per ulteriori informazioni.

Preparare l'ambiente

I requisiti di configurazione di base per eseguire gli script Python sono descritti di seguito.

Python 3

Devi avere l'ultima versione di Python 3 installata.

Librerie aggiuntive

Le librerie **requests** e **urllib3** devono essere installate. È possibile utilizzare pip o un altro tool di gestione Python appropriato per il proprio ambiente.

Accesso alla rete

La workstation in cui vengono eseguiti gli script deve disporre dell'accesso di rete e poter raggiungere Astra Control. Quando si utilizza Astra Control Service, è necessario essere connessi a Internet ed essere in grado di connettersi al servizio all'indirizzo `https://astra.netapp.io`.

Informazioni sull'identità

È necessario un account Astra valido con l'identificativo dell'account e il token API. Vedere "Ottieni un token API" per ulteriori informazioni.

Creare i file di input JSON

Gli script Python si basano sulle informazioni di configurazione contenute nei file di input JSON. I file di esempio sono forniti di seguito.

> È necessario aggiornare gli esempi in base all'ambiente in uso.

Informazioni sull'identità

Il seguente file contiene il token API e l'account Astra. È necessario passare questo file agli script Python utilizzando `-i` (o. `--identity`) Parametro CLI.

```
{
  "api_token": "kH4CA_uVIa8q9UuPzhJaAHaGlaR7-no901DkkrVjIXk=",
  "account_id": "5131dfdf-03a4-5218-ad4b-fe84442b9786"
}
```

Elencare le applicazioni

Puoi utilizzare il seguente script per elencare le applicazioni per il tuo account Astra.

> Vedere "Prima di iniziare" Per un esempio del file di input JSON richiesto.

```
#!/usr/bin/env python3
##-----------------------------------------------------------------------
------
#
# Usage: python3 list_man_apps.py -i identity_file.json
#
# (C) Copyright 2022 NetApp, Inc.
#
# This sample code is provided AS IS, with no support or warranties of
# any kind, including but not limited for warranties of merchantability
# or fitness of any kind, expressed or implied. Permission to use,
# reproduce, modify and create derivatives of the sample code is granted
# solely for the purpose of researching, designing, developing and
# testing a software application product for use with NetApp products,
# provided that the above copyright notice appears in all copies and
# that the software application product is distributed pursuant to terms
# no less restrictive than those set forth herein.
#
##-----------------------------------------------------------------------
------

import argparse
import json
import requests
import urllib3
import sys

# Global variables
api_token = ""
account_id = ""

def get_managed_apps():
    ''' Get and print the list of apps '''

    # Global variables
    global api_token
    global account_id

    # Create an HTTP session
    sess1 = requests.Session()
```

```
    # Suppress SSL unsigned certificate warning
    urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

    # Create URL
    url1 = "https://astra.netapp.io/accounts/" + account_id +
"/k8s/v2/apps"

    # Headers and response output
    req_headers  = {}
    resp_headers = {}
    resp_data    = {}

    # Prepare the request headers
    req_headers.clear
    req_headers['Authorization'] = "Bearer " + api_token
    req_headers['Content-Type'] = "application/astra-app+json"
    req_headers['Accept'] = "application/astra-app+json"

    # Make the REST call
    try:
        resp1 = sess1.request('get', url1, headers=req_headers,
allow_redirects=True, verify=False)

    except requests.exceptions.ConnectionError:
        print("Connection failed")
        sys.exit(1)

    # Retrieve the output
    http_code = resp1.status_code
    resp_headers = resp1.headers

    # Print the list of apps
    if resp1.ok:
        resp_data = json.loads(resp1.text)
        items = resp_data['items']
        for i in items:
            print(" ")
            print("Name: " + i['name'])
            print("ID: " + i['id'])
            print("State: " + i['state'])
    else:
        print("Failed with HTTP status code: " + str(http_code))

    print(" ")
```

```
    # Close the session
    sess1.close()

    return

def read_id_file(idf):
    ''' Read the identity file and save values '''

    # Global variables
    global api_token
    global account_id

    with open(idf) as f:
        data = json.load(f)

    api_token = data['api_token']
    account_id = data['account_id']

    return

def main(args):
    ''' Main top level function '''

    # Global variables
    global api_token
    global account_id

    # Retrieve name of JSON input file
    identity_file = args.id_file

    # Get token and account
    read_id_file(identity_file)

    # Issue REST call
    get_managed_apps()

    return

def parseArgs():
        ''' Parse the CLI input parameters '''

        parser = argparse.ArgumentParser(description='Astra REST API -
List the apps',
                                add_help = True)
        parser.add_argument("-i", "--identity", action="store", dest
="id_file", default=None,
                                help='(Req) Name of the identity input file',
```

```
required=True)

        return parser.parse_args()

if __name__ == '__main__':
    ''' Begin here '''

    # Parse input parameters
    args = parseArgs()

    # Call main function
    main(args)
```

Riferimento API

È possibile accedere ai dettagli delle chiamate REST API di Astra Control, inclusi i metodi HTTP, i parametri di input e le risposte. Questo riferimento completo è utile quando si sviluppano applicazioni di automazione utilizzando l'API REST.

> La documentazione di riferimento API REST è attualmente fornita con Astra Control ed è disponibile online.

Prima di iniziare

Hai bisogno di un account per Astra Control Center o Astra Control Service.

Fasi

1. Accedi ad Astra utilizzando le credenziali del tuo account.

   Accedere al seguente sito per Astra Control Service: "https://astra.netapp.io"

2. Fare clic sull'icona a forma di figura nella parte superiore destra della pagina e selezionare **API access**.
3. Nella parte superiore della pagina, fare clic sull'URL visualizzato sotto **API Documentation** (documentazione API).
4. Se richiesto, fornire nuovamente le credenziali dell'account.

Risorse aggiuntive

Sono disponibili ulteriori risorse a cui è possibile accedere per ottenere assistenza e ottenere ulteriori informazioni sui servizi cloud e sul supporto NetApp, nonché sui concetti generali DI REST e cloud.

Astra

- "Documentazione di Astra Control Center 22.08"

  Documentazione per l'attuale release del software Astra Control Center implementato presso la sede del cliente.

- "Documentazione del servizio Astra Control"

  Documentazione per la release corrente del software Astra Control Service disponibile nel cloud pubblico.

- "Documentazione di Astra Trident"

  Documentazione per l'attuale release del software Astra Trident, un orchestrator di storage open source gestito da NetApp.

- "Documentazione della famiglia Astra"

  Posizione centrale per l'accesso a tutta la documentazione Astra per implementazioni di cloud pubblico e on-premise.

Risorse cloud di NetApp

- "NetApp BlueXP"

  Sito centrale per le soluzioni cloud di NetApp.

- "Console NetApp Cloud Central"

  Console di servizio NetApp Cloud Central con accesso.

- "Supporto NetApp"

  Accesso a strumenti per la risoluzione dei problemi, documentazione e assistenza tecnica.

Concetti DI REST e cloud

- Dottorato "dissertazione" Di Roy Fielding

  Questa pubblicazione ha introdotto e definito il modello di sviluppo dell'applicazione REST.

- "Auth0"

  Si tratta del servizio della piattaforma di autenticazione e autorizzazione utilizzato dal servizio Astra per l'accesso web.

- "Editor RFC"

  Fonte autorevole per gli standard web e Internet mantenuta come una raccolta di documenti RFC numerati in modo univoco.

Versioni precedenti della documentazione di Astra Control Automation

È possibile accedere alla documentazione di automazione per le precedenti release di Astra Control ai collegamenti riportati di seguito.

- "Documentazione di Astra Control Automation 22.04"
- "Documentazione di Astra Control Automation 21.12"
- "Documentazione di Astra Control Automation 21.08"

Note legali

Le note legali forniscono l’accesso a dichiarazioni di copyright, marchi, brevetti e altro ancora.

Copyright

"https://www.netapp.com/company/legal/copyright/"

Marchi

NETAPP, il logo NETAPP e i marchi elencati nella pagina dei marchi NetApp sono marchi di NetApp, Inc. Altri nomi di società e prodotti potrebbero essere marchi dei rispettivi proprietari.

"https://www.netapp.com/company/legal/trademarks/"

Brevetti

Un elenco aggiornato dei brevetti di proprietà di NetApp è disponibile all’indirizzo:

https://www.netapp.com/pdf.html?item=/media/11887-patentspage.pdf

Direttiva sulla privacy

"https://www.netapp.com/company/legal/privacy-policy/"

Licenza API Astra Control

https://docs.netapp.com/us-en/astra-automation/media/astra-api-license.pdf