Astra REST Implementierung

Astra Automation

NetApp
March 17, 2024

Inhalt

Astra REST Implementierung

Kerndesign

REST-Web-Services

Representational State Transfer (REST) ist ein Stil für die Erstellung von verteilten Web-Anwendungen. Bei der Anwendung auf das Design einer Web-Services-API, stellt sie eine Reihe von Mainstream-Technologien und Best Practices für die Offenlegung serverbasierter Ressourcen und die Verwaltung ihrer Status. REST bietet eine konsistente Grundlage für die Applikationsentwicklung, doch je nach den jeweiligen Designoptionen können die Details jeder API variieren. Vor dem Einsatz in einer Live-Implementierung sollten Sie sich der Merkmale der Astra Control REST API bewusst sein.

Ressourcen- und Zustandsdarstellung

Ressourcen sind die Grundkomponenten eines webbasierten Systems. Beim Erstellen einer ANWENDUNG FÜR REST-Webservices umfassen die frühen Designaufgaben Folgendes:

- Identifizierung von System- oder serverbasierten Ressourcen

  Jedes System nutzt und verwaltet Ressourcen. Eine Ressource kann eine Datei-, Geschäftstransaktion-, Prozess- oder Verwaltungseinheit sein. Eine der ersten Aufgaben bei der Entwicklung einer auf REST-Webservices basierenden Applikation ist die Identifizierung der Ressourcen.

- Definition von Ressourcenstatus und zugehörigen Statusoperationen

  Die Ressourcen befinden sich immer in einer endlichen Anzahl von Staaten. Die Zustände sowie die damit verbundenen Operationen, die zur Auswirkung der Statusänderungen verwendet werden, müssen klar definiert werden.

URI-Endpunkte

Jede REST-Ressource muss definiert und über ein gut definiertes Adressierungssystem verfügbar gemacht werden. Die Endpunkte, in denen die Ressourcen gefunden und identifiziert werden, verwenden einen einheitlichen Resource Identifier (URI). Der URI bietet ein allgemeines Framework zum Erstellen eines eindeutigen Namens für jede Ressource im Netzwerk. Der Uniform Resource Locator (URL) ist ein URI-Typ, der mit Webservices zur Identifizierung und zum Zugriff von Ressourcen verwendet wird. Ressourcen werden in der Regel in einer hierarchischen Struktur ausgesetzt, die einem Dateiverzeichnis ähnelt.

HTTP-Meldungen

Hypertext Transfer Protocol (HTTP) ist das Protokoll, das vom Webservice-Client und -Server zum Austausch von Anforderungs- und Antwortmeldungen zu den Ressourcen verwendet wird. Im Rahmen der Entwicklung einer Web-Services-Anwendung werden HTTP-Methoden den Ressourcen und entsprechenden Statusmanagement-Aktionen zugeordnet. HTTP ist statusfrei. Um im Rahmen einer Transaktion eine Reihe verwandter Anforderungen und Antworten zuzuordnen, müssen daher zusätzliche Informationen in die HTTP-Header enthalten sein, die mit den Anforderungs- und Antwortdatenströmen verwendet werden.

JSON-Formatierung

Während Informationen auf verschiedene Weise zwischen einem Web-Services-Client und Server strukturiert und übertragen werden können, ist die beliebteste Option JavaScript Object Notation (JSON). JSON ist ein Branchenstandard für die Darstellung einfacher Datenstrukturen im Klartext und wird zur Übertragung von Zustandsdaten zur Beschreibung der Ressourcen verwendet. Die Astra Control REST API verwendet JSON, um die Daten zu formatieren, die im Körper von jeder HTTP-Anfrage und Antwort.

Ressourcen und Sammlungen

DIE Rest-API von Astra Control bietet Zugriff auf Ressourceninstanzen und Ressourcensammlungen.

> Konzeptionell ist eine REST **Ressource** ähnlich wie ein **Objekt** wie mit den objektorientierten Programmiersprachen und -Systemen definiert. Manchmal werden diese Begriffe synonym verwendet. Aber im Allgemeinen wird „Ressource“ bevorzugt, wenn sie im Kontext der externen REST API verwendet wird, während „Objekt“ für die entsprechenden zustandsorientierte Instanz Daten verwendet wird, die auf dem Server gespeichert sind.

Eigenschaften der Astra-Ressourcen

Die Astra Control REST API entspricht den Prinzipien des RESTful Designs. Jede Astra-Ressourceninstanz wird auf Basis eines klar definierten Ressourcentyps erstellt. Eine Reihe von Ressourceninstanzen desselben Typs wird als **Sammlung** bezeichnet. Die API-Aufrufe wirken sich auf einzelne Ressourcen oder Ressourcensammlungen aus.

Ressourcentypen

Die in der Astra Control REST API enthaltenen Ressourcentypen weisen folgende Merkmale auf:

- Jeder Ressourcentyp wird mit einem Schema definiert (in der Regel in JSON)
- Jedes Ressourcenschema enthält den Ressourcentyp und die -Version
- Ressourcentypen sind global eindeutig

Ressourceninstanzen

Die über die Astra Control REST-API verfügbaren Ressourceinstanzen weisen folgende Merkmale auf:

- Ressourceninstanzen werden auf Basis eines einzelnen Ressourcentyps erstellt
- Der Ressourcentyp wird mit dem Wert Medientyp angezeigt
- Instanzen bestehen aus statusorientierten Daten, die vom Astra-Service gewartet werden
- Auf jede Instanz kann über eine eindeutige und langlebige URL zugegriffen werden
- In Fällen, in denen eine Ressourceninstanz mehr als eine Darstellung haben kann, können verschiedene Medientypen verwendet werden, um die gewünschte Darstellung anzufordern

Ressourcensammlungen

Die Ressourcensammlungen, die über die ASTRA Control REST-API verfügbar sind, weisen folgende Merkmale auf:

- Der Satz von Ressourceninstanzen eines einzelnen Ressourcentyps wird als Sammlung bezeichnet
- Ressourcensammlungen haben eine einzigartige und langlebige URL

Instanz-IDs

Jeder Ressourceninstanz wird bei der Erstellung eine Kennung zugewiesen. Diese Kennung ist ein 128-Bit UUIDv4-Wert. Die zugewiesenen UUUIDv4-Werte sind global eindeutig und unveränderbar. Nachdem ein API-Aufruf ausgegeben wurde, der eine neue Instanz erstellt, wird eine URL mit der zugehörigen id an den Anrufer in A zurückgegeben `Location` Kopfzeile der HTTP-Antwort. Sie können die Kennung extrahieren und bei nachfolgenden Anrufen verwenden, wenn Sie sich auf die Ressourceninstanz beziehen.

> Die Ressourcen-ID ist der primäre Schlüssel, der für Sammlungen verwendet wird.

Gemeinsame Struktur für Astra-Ressourcen

Jede Astra Control-Ressource ist mit einer gemeinsamen Struktur definiert.

Einheitliche Daten

Jede Astra-Ressource enthält die in der folgenden Tabelle aufgeführten Schlüsselwerte.

Taste	Beschreibung
Typ	Ein global eindeutiger Ressourcentyp, der als **Ressourcentyp** bezeichnet wird.
Version	Eine Version-ID, die als **Resource-Version** bezeichnet wird.
id	Ein global eindeutiger Bezeichner, der als **Resource Identifier** bezeichnet wird.
Metadaten	Ein JSON-Objekt mit verschiedenen Informationen, einschließlich Benutzer- und Systemetiketten.

Metadatenobjekt

Das JSON-Metadatenobjekt, das in jeder Astra-Ressource enthalten ist, enthält die in der folgenden Tabelle aufgeführten Schlüsselwerte.

Taste	Beschreibung
Etiketten	JSON-Array mit Client-angegebenen Beschriftungen, die der Ressource zugeordnet sind.
CreationZeitstempel	JSON-Zeichenfolge mit einem Zeitstempel, der angibt, wann die Ressource erstellt wurde.
Änderungszeitstempel	JSON-Zeichenfolge mit einem ISO-8601-formatierten Zeitstempel, der angibt, wann die Ressource zuletzt geändert wurde.
Erstellt von	JSON-Zeichenfolge mit der UUUIDv4-Kennung der Benutzer-id, die die Ressource erstellt hat. Wenn die Ressource von einer internen Systemkomponente erstellt wurde und der Erstellungseinheit keine UUID zugeordnet ist, wird die **Null** UUID verwendet.

Ressourcenstatus

Ausgewählte Ressourcen A `state` Wert, der zur Orchestrierung von Lifecycle-Übergängen und zur Steuerung des Zugriffs eingesetzt wird.

HTTP-Details

Die Astra Control REST-API verwendet HTTP und zugehörige Parameter, um auf die Ressourceinstanzen und -Sammlungen zu reagieren. Einzelheiten zur HTTP-Implementierung finden Sie unten.

API-Transaktionen und das CRUD-Modell

Die Astra Control REST API implementiert ein transaktionsorientiertes Modell mit klar definierten Abläufen und Zustandsübergängen.

API-Transaktion bei Anfrage und Reaktion

Jeder REST-API-Aufruf erfolgt als HTTP-Anfrage an den Astra-Service. Jede Anforderung generiert eine entsprechende Antwort zurück an den Client. Dieses Request-Response-Paar kann als API-Transaktion betrachtet werden.

Unterstützung für CRUD-Betriebsmodell

Auf Grundlage des **CRUD**-Modells kann auf alle über die Astra Control REST API verfügbaren Ressourcen und Sammlungen zugegriffen werden. Es gibt vier Vorgänge, von denen jede einer einzigen HTTP-Methode zugeordnet wird. Dazu gehören:

- Erstellen
- Lesen
- Aktualisierung
- Löschen

Bei einigen der Astra-Ressourcen wird nur ein Teil dieser Vorgänge unterstützt. Sie sollten die überprüfen "Online-API-Referenz" Weitere Informationen zu einem bestimmten API-Aufruf.

HTTP-Methoden

Die von der API unterstützten HTTP-Methoden oder Verben werden in der folgenden Tabelle dargestellt.

Methode	CRUD	Beschreibung
GET	Lesen	Ruft Objekteigenschaften für eine Ressourceninstanz oder -Sammlung ab. Dies wird als **list**-Operation bei Verwendung mit einer Sammlung betrachtet.
POST	Erstellen	Erstellt eine neue Ressourceninstanz basierend auf den Eingabeparametern. Die langfristige URL wird in A zurückgegeben `Location` Kopfzeile der Antwort.
PUT	Aktualisierung	Aktualisiert eine gesamte Ressourceninstanz mit dem mitgelieferten JSON Request Body. Wichtige Werte, die nicht vom Benutzer änderbar sind, bleiben erhalten.
Löschen	Löschen	Löscht eine vorhandene Ressourceninstanz.

Header für Anfragen und Antworten

Die folgende Tabelle fasst die HTTP-Header zusammen, die mit der Astra Control REST API verwendet werden.

Siehe "RFC 7232" Und "RFC 7233" Finden Sie weitere Informationen.

Kopfzeile	Typ	Nutzungshinweise
Akzeptieren	Anfrage	Wenn der Wert „/“ ist oder nicht angegeben wird, `application/json` Wird in der Kopfzeile der Inhaltstyp-Antwort zurückgegeben. Wenn der Wert auf den Astra Resource Media Type gesetzt ist, wird derselbe Medientyp in der Kopfzeile des Inhaltstyps zurückgegeben.
Autorisierung	Anfrage	Träger-Token mit dem API-Schlüssel für den Benutzer.
Inhaltstyp	Antwort	Wird basierend auf dem zurückgegeben `Accept` Kopfzeile der Anfrage.
Etag	Antwort	Im Lieferumfang eines erfolgreichen RFC 7232-Standards enthalten. Der Wert ist eine hexadezimale Darstellung des MD5-Werts für die gesamte JSON-Ressource.
If-Match	Anfrage	Ein Precondition Request Header, wie in Abschnitt 3.1 RFC 7232 beschrieben und unterstützt **PUT** Anforderungen.
Wenn-Geändert-Seit	Anfrage	Ein Precondition Request Header, wie in Abschnitt 3.4 RFC 7232 beschrieben und unterstützt **PUT** Anforderungen.
Wenn-Unmodified-Since	Anfrage	Ein Precondition Request Header, wie in Abschnitt 3.4 RFC 7232 beschrieben und unterstützt **PUT** Anforderungen.
Standort	Antwort	Enthält die vollständige URL der neu erstellten Ressource.

Abfrageparameter

Die folgenden Abfrageparameter stehen zur Verwendung mit Ressourcensammlungen zur Verfügung. Siehe "Arbeit mit Sammlungen" Finden Sie weitere Informationen.

Abfrageparameter	Beschreibung
Einschließlich	Enthält die Felder, die beim Lesen einer Sammlung zurückgegeben werden sollen.
Filtern	Gibt die Felder an, die für die Rückgabe einer Ressource beim Lesen einer Sammlung übereinstimmen müssen.
Orderby	Bestimmt die Reihenfolge der beim Lesen einer Sammlung zurückgegebenen Ressourcen.
Grenze	Begrenzt die maximale Anzahl an Ressourcen, die beim Lesen einer Sammlung zurückgegeben werden.
überspringen	Legt fest, wie viele Ressourcen beim Lesen einer Sammlung weitergehen und überspringen sollen.
Zählen	Gibt an, ob die Gesamtzahl der Ressourcen im Metadatenobjekt zurückgegeben werden soll.

HTTP-Statuscodes

Im Folgenden werden die HTTP-Statuscodes beschrieben, die von der REST-API von Astra Control verwendet werden.

Die Astra Control REST API nutzt auch den **Problemdetails für HTTP APIs** Standard. Siehe "Diagnose und Support" Finden Sie weitere Informationen.

Codieren	Bedeutung	Beschreibung
200	OK	Zeigt Erfolg für Anrufe an, die keine neue Ressourceninstanz erstellen.
201	Erstellt	Ein Objekt wurde erfolgreich erstellt, und die Kopfzeile für die Standortantwort enthält die eindeutige Kennung für das Objekt.
204	Kein Inhalt	Die Anfrage war erfolgreich, obwohl kein Inhalt zurückgegeben wurde.
400	Schlechte Anfrage	Die Eingabe der Anfrage ist nicht erkannt oder nicht angemessen.
401	Nicht Autorisiert	Der Benutzer ist nicht autorisiert und muss authentieren.
403	Verboten	Der Zugriff wird aufgrund eines Autorisierungsfehlers verweigert.
404	Nicht gefunden	Die Ressource, auf die in diesem Antrag verwiesen wird, ist nicht vorhanden.
409	Konflikt	Der Versuch, ein Objekt zu erstellen, ist fehlgeschlagen, weil das Objekt bereits vorhanden ist.
500	Interner Fehler	Ein allgemeiner interner Fehler ist auf dem Server aufgetreten.
503	Service nicht verfügbar	Der Dienst ist aus irgendeinem Grund nicht bereit, die Anfrage zu bearbeiten.

URL-Format

Die allgemeine Struktur der URL, die für den Zugriff auf eine Ressourceninstanz oder -Sammlung über DIE REST-API verwendet wird, besteht aus mehreren Werten. Diese Struktur spiegelt das zugrunde liegende Objektmodell und das Systemdesign wider.

Konto als Root

Die Wurzel des Ressourcenpfads zu jedem REST-Endpunkt ist das Astra-Konto. Daher beginnen alle Pfade in der URL mit `/account/{account_id}` Wo `account_id` Ist der eindeutige UUUIDv4-Wert für das Konto. Interne Struktur Dies ist ein Design, in dem der gesamte Ressourcenzugriff auf einem bestimmten Konto basiert.

Kategorie der Endpoint-Ressourcen

Die Astra-Ressourcenendpunkte lassen sich in drei verschiedene Kategorien einteilen:

- Kern (`/core`)
- Gemanagte Applikation (`/k8s`)
- Topologie (`/topology`)

Siehe "Ressourcen" Finden Sie weitere Informationen.

Kategorienversion

Jede der drei Ressourcenkategorien verfügt über eine globale Version, die die Version der Ressourcen steuert, auf die zugegriffen wird. Nach Konventionen und Definition zu einer neuen Hauptversion einer Ressourcenkategorie wechseln (z. B. von) `/v1` Bis `/v2`) Wird Bruchänderungen in der API einführen.

Ressourceinstanz oder -Sammlung

Eine Kombination von Ressourcentypen und Identifikatoren kann im Pfad verwendet werden, basierend darauf, ob auf eine Ressourceninstanz oder -Sammlung zugegriffen wird.

Beispiel

- Ressourcenpfad

  Basierend auf der oben dargestellten Struktur ist ein typischer Pfad zu einem Endpunkt: `/accounts/{account_id}/core/v1/users`.

- Vollständige URL

  Die vollständige URL für den entsprechenden Endpunkt lautet: `https://astra.netapp.io/accounts/{account_id}/core/v1/users`.

Ressourcen und Endpunkte

Zur Automatisierung einer Astra Implementierung können Sie auf die Ressourcen zugreifen, die über die ASTRA Control REST-API bereitgestellt werden. Jede Ressource ist über einen oder mehrere Endpunkte verfügbar. Nachfolgend finden Sie eine Einführung zu DEN REST-Ressourcen, die Sie im Rahmen einer Automatisierungsimplementierung nutzen können.

> Das Format des Pfads und der vollständigen URL für den Zugriff auf die Astra Control-Ressourcen basiert auf mehreren Werten. Siehe "URL-Format" Finden Sie weitere Informationen. Siehe auch "Online-API-Referenz" Weitere Informationen zur Verwendung der Astra-Ressourcen und -Endpunkte.

Zusammenfassung der Astra Control REST-Ressourcen

Die primären Ressourcenendpunkte in der Astra Control REST API sind in drei Kategorien unterteilt. Auf jede Ressource kann mit allen CRUD-Vorgängen (Erstellen, Lesen, Aktualisieren, Löschen) zugegriffen werden, sofern nicht anders angegeben.

Die Spalte **Release** zeigt den Astra-Release an, als die Ressource zum ersten Mal eingeführt wurde. Dieses Feld ist für die Ressourcen erweitert, die zuletzt der REST-API hinzugefügt wurden.

Kernressourcen

Die Kernressourcenendpunkte bieten die grundlegenden Services, die zum Aufbau und zur Wartung der Astra-Laufzeitumgebung erforderlich sind.

Ressource	Freigabe	Beschreibung
Konto	21.12	Mithilfe der Account-Ressourcen können Sie die isolierten Mandanten innerhalb der mandantenfähigen Astra Control Implementierungsumgebung managen.
ASUP	21.08	Die ASUP Ressourcen stellen die AutoSupport Bundles dar, die an den NetApp Support weitergeleitet werden.

Ressource	Freigabe	Beschreibung
Zertifikat	22.08	Die Zertifikatressourcen stellen die installierten Zertifikate dar, die für eine starke Authentifizierung für ausgehende Verbindungen verwendet werden.
Anmeldedaten	21.04	Die Ressourcen für Zugangsdaten enthalten sicherheitsbezogene Informationen, die mit Astra-Benutzern, Clustern, Buckets und Storage-Back-Ends verwendet werden können.
Berechtigung	21.08	Die Berechtigungsressourcen stellen die Funktionen und Kapazitäten dar, die für ein Konto auf Basis der aktiven Lizenzen und Abonnements verfügbar sind.
Ereignis	21.04	Die Event-Ressourcen repräsentieren alle Ereignisse, die im System auftreten, einschließlich der Untergruppe, die als Benachrichtigungen klassifiziert ist.
Execution Hook	21.12	Die Hook-Ressourcen für die Ausführung stellen benutzerdefinierte Skripts dar, die Sie entweder vor oder nach einem Snapshot einer verwalteten App ausführen können.
Merkmal	21.08	Die Funktionsressourcen stellen ausgewählte Astra-Funktionen dar, die Sie abfragen können, um festzustellen, ob diese im System aktiviert oder deaktiviert sind. Der Zugriff ist auf schreibgeschützt beschränkt.
Gruppieren	22.08	Die Gruppenressourcen sind die Astra-Gruppen und die damit verbundenen Ressourcen. In der aktuellen Version werden nur LDAP-Gruppen unterstützt.
Hook-Quelle	21.12	Die Hakenquellenressourcen stellen den aktuellen Quellcode dar, der mit einem Testsuite verwendet wird. Die Trennung des Quellcodes von der Ausführungskontrolle hat mehrere Vorteile, wie z. B. die Freigabe der Skripte.
LDAP-Gruppe	22.1	Sie können die Gruppen im konfigurierten LDAP-Server auflisten. Der Zugriff auf die LDAP-Gruppen ist schreibgeschützt.
LDAP-Benutzer	22.11	Sie können die Benutzer im konfigurierten LDAP-Server auflisten. Der Zugriff auf LDAP-Benutzer ist schreibgeschützt.
Lizenz	21.08	Die Lizenzressourcen stellen die für ein Astra-Konto verfügbaren Lizenzen dar.
Benachrichtigung	21.04	Die Benachrichtigungsressourcen sind Astra-Ereignisse mit einem Benachrichtigungsziel. Der Zugriff erfolgt auf Benutzerbasis.
Paket	22.04	Die Paketressourcen ermöglichen die Registrierung und den Zugriff auf Paketdefinitionen. Softwarepakete bestehen aus verschiedenen Komponenten, einschließlich Dateien, Bildern und anderen Artefakten.
Berechtigung	23.06	Die Berechtigungsressourcen stellen Berechtigungen für Vorgänge im System dar. Die API bietet schreibgeschützten Zugriff auf die Berechtigungen.
Rolle	23.06	Die Rollenressourcen stellen Rollen dar, die im System verfügbar sind. Die API bietet schreibgeschützten Zugriff auf die Rollen.

Ressource	Freigabe	Beschreibung
Rollenbindung	21.04	Die Role Binding Ressourcen stellen die Beziehungen zwischen bestimmten Paaren von Benutzern und Konten dar. Zusätzlich zur Verknüpfung zwischen den beiden wird für jede über eine bestimmte Rolle ein Satz von Berechtigungen festgelegt.
Einstellung	21.08	Die Einstellungsressourcen stellen eine Sammlung von Schlüsselwert-Paaren dar, die ein Feature für ein bestimmtes Astra-Konto beschreiben.
Abonnement	21.08	Die Abonnementressourcen stellen die aktiven Abonnements für ein Astra-Konto dar.
Aufgabe	22.11	Die Task-Ressourcen bieten schreibgeschützten Zugriff auf verwaltete Aufgabe und können verwendet werden, um den Status der internen Langlaufaufgaben anzuzeigen.
Token	21.04	Die Token-Ressourcen stellen die Token dar, die für den programmatischen Zugriff auf die Astra Control REST API verfügbar sind.
Ungelesene Benachrichtigung	21.04	Die nicht gelesenen Benachrichtigungsressourcen stellen Benachrichtigungen dar, die einem bestimmten Benutzer zugewiesen, aber noch nicht gelesen wurden.
Upgrade	22.04	Die Upgrade-Ressourcen bieten Zugriff auf Softwarekomponenten und können Upgrades initiieren.
Benutzer	21.04	Die Benutzerressourcen sind Astra-Benutzer, die auf das System basierend auf ihrer definierten Rolle zugreifen können.

Gemanagte Applikationsressourcen

Die Endpunkte der gemanagten Applikationsressourcen bieten Zugriff auf die gemanagten Kubernetes-Applikationen.

Ressource	Freigabe	Beschreibung
Anwendungsressource	21.04	Die Anwendungsressourcen stellen interne Sammlungen von staatlichen Informationen dar, die für das Management der Astra-Anwendungen erforderlich sind.
Applikations-Backup	21.04	Die Backup-Ressourcen der Applikation stellen Backups der gemanagten Applikationen dar.
Anwendungs-Snapshot	21.04	Die Snapshot-Ressourcen der Anwendung stellen Snapshots der verwalteten Anwendungen dar.
Überschreiben des Testablaufanhängehaken	21.12	Über die Ressourcen zum Überschreiben der Execution Hooks können Sie die vorab geladenen NetApp Standard-Testausführungshaken für bestimmte Applikationen nach Bedarf deaktivieren.
Zeitplan	21.04	Die Zeitplanressourcen sind Datensicherungsvorgänge, die im Rahmen einer Datenschutzrichtlinie für die gemanagten Applikationen geplant sind.

Topologieressourcen

Die Endpunkte der Topologieressourcen bieten Zugriff auf nicht verwaltete Applikationen und Storage-Ressourcen.

Ressource	Freigabe	Beschreibung
API-Ressource	22.11	Die API-Ressourcenendpunkte bieten einen schreibgeschützten Zugriff auf die Kubernetes-Ressourcen in einem bestimmten gemanagten Cluster.
App.	21.04	Die App-Ressourcen umfassen alle Kubernetes-Applikationen, auch die, die nicht von Astra gemanagt werden.
AppMirror	22.08	Die AppMirror-Ressourcen stellen die AppMirror-Ressourcen dar, die für das Management von Applikationsspiegelungsbeziehungen bereitgestellt werden.
Eimer	21.08	Die Bucket-Ressourcen sind die S3-Cloud-Buckets, die für die Speicherung von Backups der vom Astra gemanagten Applikationen verwendet werden.
Cloud	21.08	Die Cloud-Ressourcen stellen Clouds dar, mit denen Astra-Clients verbunden werden können, um Cluster und Applikationen zu managen.
Cluster	21.08	Die Cluster-Ressourcen stellen die Kubernetes-Cluster dar, die nicht von Kubernetes gemanagt werden.
Cluster-Node	21.12	Die Cluster-Node-Ressourcen bieten eine zusätzliche Auflösung, durch die Sie auf die einzelnen Nodes innerhalb eines Kubernetes-Clusters zugreifen können.
Verwalteter Cluster	21.08	Die gemanagten Cluster-Ressourcen stellen die Kubernetes-Cluster dar, die derzeit von Kubernetes gemanagt werden.
Namespace	21.12	Die Namespace-Ressourcen bieten Zugriff auf die innerhalb eines Kubernetes-Clusters verwendeten Namespaces.
Storage-Back-End	21.08	Die Storage-Back-End-Ressourcen stellen Anbieter von Storage-Services dar, die von den von Astra gemanagten Clustern und Applikationen verwendet werden können.
Storage-Klasse	21.08	Ressourcen der Storage-Klasse stellen unterschiedliche Storage-Klassen oder -Typen dar, die für ein bestimmtes gemanagtes Cluster erkannt und verfügbar sind.
Datenmenge	21.04	Die Volume-Ressourcen stellen die Kubernetes Storage Volumes dar, die mit den gemanagten Applikationen verknüpft sind.

Zusätzliche Ressourcen und Endpunkte

Zur Unterstützung einer Astra-Implementierung stehen mehrere zusätzliche Ressourcen und Endpunkte zur Verfügung.

> Diese Ressourcen und Endpunkte sind derzeit nicht in der Astra Control REST API-Referenzdokumentation enthalten.

OpenAPI

Die OpenAPI-Endpunkte bieten Zugriff auf das aktuelle OpenAPI JSON-Dokument und andere zugehörige Ressourcen.

OpenMetrics

Die OpenMetrics-Endpunkte bieten über die OpenMetrics-Ressource Zugriff auf die Kontokennzahlen. Support ist mit dem Astra Control Center Implementierungsmodell verfügbar.

Weitere Überlegungen

RBAC-Sicherheit

Die Astra REST API unterstützt die rollenbasierte Zugriffssteuerung (RBAC), um den Zugriff auf Systemfunktionen zu gewähren und einzuschränken.

Astra Rollen

Jeder Astra-Benutzer wird einer einzigen Rolle zugewiesen, die die Aktionen bestimmt, die durchgeführt werden können. Die Rollen sind in einer Hierarchie angeordnet, wie in der folgenden Tabelle beschrieben.

Rolle	Beschreibung
Eigentümer	Hat alle Berechtigungen der Admin-Rolle und kann auch Astra-Konten löschen.
Admin	Verfügt über alle Berechtigungen der Mitgliedsrolle und kann Benutzer auch dazu einladen, einem Konto beizutreten.
Mitglied	Kunden können ihre Astra-Applikations- und Computing-Ressourcen vollständig managen.
Prüfer	Beschränkt auf die Anzeige von Ressourcen.

Erweiterte RBAC mit Namespace-Granularität

Diese Funktion wurde mit Version 22.04 des Astra REST API eingeführt.

Wenn eine Rollenbindung für einen bestimmten Benutzer festgelegt wird, kann eine Einschränkung angewendet werden, um die Namespaces zu begrenzen, auf die der Benutzer Zugriff hat. Diese Bedingung kann auf verschiedene Weise definiert werden, wie in der nachstehenden Tabelle beschrieben. Siehe Parameter `roleContraints` In der Role Binding API für weitere Informationen.

Namespaces	Beschreibung
Alle	Der Benutzer kann über den Platzhalterparameter „*“ auf alle Namespaces zugreifen. Dies ist der Standardwert, um die Abwärtskompatibilität beizubehalten.
Keine	Die Bedingungsliste wird angegeben, obwohl sie leer ist. Dies bedeutet, dass der Benutzer keinen Zugriff auf einen Namespace hat.
Namespace-Liste	Die UUID eines Namespace enthält, die den Benutzer auf den Single Namespace beschränkt. Eine kommagetrennte Liste kann auch verwendet werden, um den Zugriff auf mehrere Namespaces zu ermöglichen.
Etikett	Ein Etikett wird angegeben und der Zugriff ist allen übereinstimmenden Namespaces erlaubt.

Arbeit mit Sammlungen

Die Astra Control REST API bietet verschiedene Möglichkeiten, über die definierten Abfrageparameter auf Ressourcensammlungen zuzugreifen.

Wählen Sie Werte aus

Sie können angeben, welche Schlüsselwertpaare für jede Ressourceninstanz mit dem zurückgegeben werden

sollen `include` Parameter. Alle Fälle werden im Antwortkörper zurückgegeben.

Filtern

Mithilfe der Filterung von Sammlungsressourcen kann ein API-Benutzer Bedingungen festlegen, die bestimmen, ob eine Ressource im Antwortkörper zurückgegeben wird. Der `filter` Parameter wird verwendet, um die Filterbedingung anzuzeigen.

Sortieren

Die Sammelressource-Sortierung ermöglicht einem API-Benutzer, die Reihenfolge anzugeben, in der Ressourcen im Antwortkörper zurückgegeben werden. Der `orderBy` Parameter wird verwendet, um die Filterbedingung anzuzeigen.

Paginierung

Sie können Paginierung erzwingen, indem Sie die Anzahl der Ressourceninstanzen beschränken, die für eine Anforderung über die zurückgegeben werden `limit` Parameter.

Zählen

Wenn Sie den Booleschen Parameter angeben `count` Auf einstellen `true`, Die Anzahl der Ressourcen im zurückgegebenen Array für eine bestimmte Antwort ist im Abschnitt Metadaten angegeben.

Diagnose und Support

Mit der Astra Control REST API stehen verschiedene Supportfunktionen zur Verfügung, die für Diagnose und Debugging genutzt werden können.

API-Ressourcen

Verschiedene Astra-Funktionen sind über API-Ressourcen zugänglich und bieten diagnostische Informationen und Support.

Typ	Beschreibung
Ereignis	Systemaktivitäten, die im Rahmen der Astra-Verarbeitung erfasst werden.
Benachrichtigung	Eine Untergruppe der Ereignisse, die als wichtig genug betrachtet werden, um dem Benutzer präsentiert zu werden.
Ungelesene Benachrichtigung	Die Benachrichtigungen, die noch vom Benutzer gelesen oder abgerufen werden müssen.

Ein API-Token widerrufen

Sie können ein API-Token an der Astra-Webschnittstelle widerrufen, wenn es nicht mehr benötigt wird.

Bevor Sie beginnen

Sie benötigen die Zugangsdaten, um sich für Ihre Implementierung in der Astra Web-Benutzeroberfläche anzumelden. Sie sollten auch die Token identifizieren, die Sie widerrufen möchten.

Über diese Aufgabe

Nachdem ein Token entzogen wurde, ist es sofort und dauerhaft unbrauchbar.

Schritte

1. Melden Sie sich mit Ihren Account-Anmeldedaten für Astra wie folgt an:
    a. Astra Control Service: "https://astra.netapp.io"
    b. Astra Control Center: Verwenden Sie die bei der Installation festgelegte URL für Ihre lokale Umgebung
2. Klicken Sie auf das Figurensymbol oben rechts auf der Seite und wählen Sie **API Access**.
3. Wählen Sie das Token oder die Token aus, die Sie widerrufen möchten.
4. Klicken Sie im Dropdown-Feld **Aktionen** auf **Token aufheben**.