Automatisierung mit REST-APIs

SnapCenter Software 4.6

NetApp
January 18, 2024

This PDF was generated from https://docs.netapp.com/de-de/snapcenter-46/sc-automation/overview_rest_apis.html on January 18, 2024. Always check docs.netapp.com for the latest.

Inhalt

Automatisierung mit REST-APIs . . . 1
- Übersicht ÜBER REST-APIs . . . 1
- Wie kann man nativ auf die SnapCenter REST-API zugreifen . . . 1
- REST-Web-Services-Grundlage . . . 1
- Grundlegende betriebliche Eigenschaften . . . 2
- Eingabevariablen, die eine API-Anforderung steuern . . . 4
- Interpretation einer API-Antwort . . . 7
- Unterstützte REST-APIs . . . 10
- Zugriff auf REST-APIs über die Swagger API-Webseite . . . 17
- Legen Sie los mit DER REST API . . . 17

Automatisierung mit REST-APIs

Übersicht ÜBER REST-APIs

REST-APIs können zur Durchführung mehrerer SnapCenter-Managementvorgänge verwendet werden. REST-APIs sind über die Swagger Webseite zugänglich.

Sie können auf die Swagger-Webseite unter *https://<SnapCenter_IP_address_or_name>:<SnapCenter_port>/swagger/* zugreifen, um die REST-API-Dokumentation anzuzeigen, sowie einen API-Aufruf manuell ausstellen.

Folgende Plug-ins unterstützen REST-APIs:

- Plug-in für Microsoft SQL Server
- Plug-in für SAP HANA Database
- Benutzerdefinierte Plug-ins
- Plug-in für Oracle Database

Wie kann man nativ auf die SnapCenter REST-API zugreifen

Sie können über jede Programmiersprache, die einen REST-Client unterstützt, direkt auf die SnapCenter REST-API zugreifen. Beliebte Sprachen sind Python, PowerShell und Java.

REST-Web-Services-Grundlage

Representational State Transfer (REST) ist ein Stil für die Erstellung von verteilten Web-Anwendungen. Bei der Anwendung auf das Design einer Web-Services-API werden eine Reihe von Technologien und Best Practices erstellt, um serverbasierte Ressourcen freizulegen und deren Status zu verwalten. Die flexible Grundlage für das Management von SnapCenter bildet mit Mainstream-Protokollen und -Standards.

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, sollten 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 SnapCenter REST API verwendet JSON, um die Daten zu formatieren, die im Körper jeder HTTP-Anfrage und Antwort verwendet werden.

Grundlegende betriebliche Eigenschaften

IM RUHEZUSTAND werden einheitliche Technologien und Best Practices erstellt, jedoch können die Details jeder API je nach dem verfügbaren Design variieren.

API-Transaktion bei Anfrage und Reaktion

Jeder REST-API-Aufruf wird als HTTP-Anfrage an das SnapCenter-Serversystem durchgeführt, das eine entsprechende Antwort auf den Client generiert. Dieses Anforderungs- und Antwortpaar wird als API-Transaktion betrachtet.

Bevor Sie die API verwenden, sollten Sie mit den verfügbaren Eingabevariablen zur Steuerung einer Anfrage und dem Inhalt der Antwortausgabe vertraut sein.

Unterstützung von CRUD-Vorgängen

Auf alle über das SnapCenter REST API verfügbaren Ressourcen kann basierend auf dem CRUD-Modell zugegriffen werden:

- Erstellen

- Lesen
- Aktualisierung
- Löschen

Für einige der Ressourcen wird nur ein Teil der Vorgänge unterstützt.

Objektkennungen

Jeder Ressourceninstanz oder jedem Objekt wird eine eindeutige Kennung zugewiesen, wenn sie erstellt wird. In den meisten Fällen ist die Kennung eine 128-Bit-UUID. Diese Kennungen sind global eindeutig in einem bestimmten SnapCenter-Server.

Nachdem ein API-Aufruf ausgegeben wurde, der eine neue Objektinstanz erstellt, wird eine URL mit der zugehörigen ID an den Anrufer in der Kopfzeile der HTTP-Antwort zurückgegeben. Sie können die Kennung extrahieren und bei nachfolgenden Anrufen verwenden, wenn Sie sich auf die Ressourceninstanz beziehen.

Der Inhalt und die interne Struktur der Objektkennungen können jederzeit geändert werden. Wenn Sie auf die zugeordneten Objekte verweisen, sollten Sie die Kennungen für die entsprechenden API-Aufrufe nur nach Bedarf verwenden.

Objektinstanzen und -Sammlungen

Je nach Ressourcenpfad und HTTP-Methode kann ein API-Aufruf auf eine bestimmte Objektinstanz oder eine Sammlung von Objekten angewendet werden.

Synchroner und asynchroner Betrieb

SnapCenter führt eine HTTP-Anforderung durch, die von einem Client entweder synchron oder asynchron empfangen wird.

Synchrone Verarbeitung

SnapCenter führt die Anfrage sofort aus und antwortet mit einem HTTP-Statuscode von 200 oder 201, wenn er erfolgreich ist.

Jede Anfrage, die die Methode GET verwendet, wird immer synchron ausgeführt. Zusätzlich sind Anfragen, die POST verwenden, so ausgelegt, dass sie synchron ausgeführt werden können, wenn sie in weniger als zwei Sekunden abgeschlossen sein sollen.

Asynchrone Verarbeitung

Wenn eine asynchrone Anforderung gültig ist, erstellt SnapCenter eine Hintergrundaufgabe zur Verarbeitung der Anforderung und ein Jobobjekt zum Anker der Aufgabe. Der HTTP-Statuscode 202 wird zusammen mit dem Jobobjekt an den Anrufer zurückgegeben. Sie sollten den Status des Jobs abrufen, um den Erfolg oder den Fehler zu ermitteln.

Anfragen, die DIE POST- und LÖSCHMETHODEN verwenden, werden asynchron ausgeführt, wenn dies voraussichtlich mehr als zwei Sekunden dauert.

Sicherheit

Die Sicherheit der REST-API basiert in erster Linie auf den vorhandenen Sicherheitsfunktionen von

SnapCenter. Die folgende Sicherheit wird von der API verwendet:

Sicherheit In Transportschicht

Der gesamte über das Netzwerk zwischen dem SnapCenter-Server und dem Client gesendete Datenverkehr wird basierend auf den SnapCenter-Konfigurationseinstellungen in der Regel mit TLS verschlüsselt.

HTTP-Authentifizierung

Auf HTTP-Ebene wird die grundlegende Authentifizierung für die API-Transaktionen verwendet. Jeder Anforderung wird ein HTTP-Header mit dem Benutzernamen und Passwort in einem Base64-String hinzugefügt.

Eingabevariablen, die eine API-Anforderung steuern

Sie können steuern, wie ein API-Aufruf über Parameter und Variablen verarbeitet wird, die in der HTTP-Anforderung festgelegt sind.

HTTP-Methoden

Die von der SnapCenter REST API unterstützte HTTP-Methoden sind in der folgenden Tabelle aufgeführt.

Nicht alle HTTP-Methoden sind an jedem REST-Endpunkt verfügbar.

HTTP-Methode	Beschreibung
GET	Ruft Objekteigenschaften auf einer Ressourceninstanz oder -Sammlung ab.
POST	Erstellt eine neue Ressourceninstanz basierend auf der angegebenen Eingabe.
Löschen	Löscht eine vorhandene Ressourceninstanz.
PUT	Ändert eine vorhandene Ressourceninstanz.

Anfragekopfzeilen

Sie sollten mehrere Header in die HTTP-Anfrage aufnehmen.

Inhaltstyp

Wenn der Anforderungsinstanz JSON enthält, sollte dieser Header auf *Application/json* gesetzt werden.

Akzeptieren

Dieser Header sollte auf *Application/json* gesetzt werden.

Autorisierung

Die grundlegende Authentifizierung sollte mit dem Benutzernamen und dem Passwort als base64-Zeichenfolge codiert werden.

Text anfordern

Der Inhalt der Anfraentext variiert je nach Anruf. Der HTTP-Request-Text besteht aus einem der folgenden Elemente:

- JSON-Objekt mit Eingabevariablen
- Leer

Objekte filtern

Wenn Sie einen API-Aufruf ausgeben, der GET verwendet, können Sie die zurückgegebenen Objekte anhand eines beliebigen Attributs einschränken oder filtern. Sie können beispielsweise einen genauen Wert angeben, der übereinstimmt:

```
<field>=<query value>
```

Neben einer genauen Übereinstimmung stehen auch andere Operatoren zur Verfügung, um einen Satz von Objekten über einen Wertebereich zurückzugeben. Die SnapCenter REST API unterstützt die in der nachfolgenden Tabelle aufgeführten Filteroperatoren.

Operator	Beschreibung
=	Gleich
<	Kleiner als
>	Größer als
⇐	Kleiner oder gleich
>=	Größer oder gleich
AKTUALISIERUNG	Oder
!	Nicht gleich
*	Gierige Wildcard

Sie können auch eine Sammlung von Objekten zurückgeben, basierend darauf, ob ein bestimmtes Feld gesetzt wird oder nicht, indem Sie das Schlüsselwort **Null** oder dessen Negation **!null** als Teil der Abfrage verwenden.

Nicht festgelegte Felder werden in der Regel von übereinstimmenden Abfragen ausgeschlossen.

Es werden bestimmte Objektfelder angefordert

Standardmäßig gibt die Ausgabe eines API-Aufrufs mithilfe VON GET nur die Attribute zurück, die das Objekt oder die Objekte eindeutig identifizieren. Dieser minimale Feldsatz dient als Schlüssel für jedes Objekt und variiert je nach Objekttyp. Sie können zusätzliche Objekteigenschaften mithilfe der auswählen `fields` Abfrageparameter auf folgende Weise:

Allgemeine oder Standardfelder

Geben Sie **Fields=*** an, um die am häufigsten verwendeten Objektfelder abzurufen. Diese Felder werden normalerweise im lokalen Serverspeicher verwaltet oder erfordern nur wenig Verarbeitung für den Zugriff. Dies

sind die gleichen Eigenschaften, die für ein Objekt zurückgegeben werden, nachdem GET mit einem URL-Pfadschlüssel (UUID) verwendet wurde.

Alle Felder

Geben Sie **fields=**** an, um alle Objektfelder abzurufen, einschließlich derer, die für den Zugriff auf zusätzliche Serververarbeitung erforderlich sind.

Benutzerdefinierte Feldauswahl

Geben Sie mit **fields=<field_Name>** das genaue Feld ein. Wenn Sie mehrere Felder anfordern, müssen die Werte durch Kommas ohne Leerzeichen getrennt werden.

> Als Best Practice sollten Sie immer die gewünschten Felder identifizieren. Sie sollten nur die gemeinsamen Felder oder alle Felder abrufen, wenn Sie dies benötigen. Welche Felder sind als „Common“ klassifiziert und mit *fields=** zurückgegeben werden, wird durch NetApp aufgrund der internen Performance-Analyse bestimmt. Die Klassifizierung eines Felds kann sich in zukünftigen Releases ändern.

Sortieren von Objekten im Ausgabungsset

Die Datensätze in einer Ressourcensammlung werden in der vom Objekt definierten Standardreihenfolge zurückgegeben. Sie können die Bestellung über ändern `order_by` Abfrage-Parameter mit Feldname und Sortierrichtung wie folgt:

```
order_by=<field name> asc|desc
```

Sie können beispielsweise das Typfeld in absteigender Reihenfolge, gefolgt von id in aufsteigender Reihenfolge sortieren:

```
order_by=type desc, id asc
```

- Wenn Sie ein Sortierfeld angeben, aber keine Richtung angeben, werden die Werte in aufsteigender Reihenfolge sortiert.
- Wenn Sie mehrere Parameter eingeben, müssen Sie die Felder mit einem Komma trennen.

Paginierung beim Abrufen von Objekten in einer Sammlung

Wenn ein API-Aufruf über GET auf eine Sammlung von Objekten desselben Typs zugreifen soll, versucht SnapCenter, auf der Grundlage von zwei Einschränkungen so viele Objekte wie möglich zurückzugeben. Mit zusätzlichen Abfrageparametern auf der Anforderung können Sie jede dieser Einschränkungen steuern. Die erste Bedingung, die für eine bestimmte GET-Anforderung erreicht wurde, beendet die Anforderung und begrenzt damit die Anzahl der zurückgegebenen Datensätze.

> Wenn eine Anfrage endet, bevor sie alle Objekte anführt, enthält die Antwort den Link, der zum Abrufen des nächsten Stapels von Datensätzen benötigt wird.

Die Anzahl der Objekte wird begrenzt

Standardmäßig gibt SnapCenter maximal 10,000 Objekte für EINE GET-Anforderung aus. Sie können diese Grenze mit dem Abfrageparameter *max_Records* ändern. Beispiel:

```
max_records=20
```

Die Anzahl der tatsächlich zurückgegebenen Objekte kann aufgrund der entsprechenden Zeitbeschränkung sowie der Gesamtanzahl der Objekte im System kleiner sein als die maximale Wirkung.

Begrenzung der Zeit, die zum Abrufen der Objekte verwendet wird

Standardmäßig gibt SnapCenter so viele Objekte wie möglich innerhalb der für die GET-Anforderung zulässigen Zeit zurück. Die Standard-Zeitüberschreitung beträgt 15 Sekunden. Sie können diese Grenze mit dem Abfrageparameter *return_timeout* ändern. Beispiel:

```
return_timeout=5
```

Die Anzahl der tatsächlich zurückgegebenen Objekte kann aufgrund der damit verbundenen Beschränkung auf die Anzahl der Objekte sowie die Gesamtanzahl der Objekte im System kleiner sein als die maximal zulässige Anzahl.

Verengung des Ergebnisset

Bei Bedarf können Sie diese beiden Parameter mit zusätzlichen Abfrageparametern kombinieren, um den Ergebnissatz einzugrenzen. Im Folgenden werden z. B. bis zu 10 EMS-Ereignisse zurückgegeben, die nach der angegebenen Zeit generiert wurden:

```
time⇒ 2018-04-04T15:41:29.140265Z&max_records=10
```

Sie können mehrere Anfragen zur Seite durch die Objekte ausgeben. Jeder nachfolgende API-Aufruf sollte einen neuen Zeitwert verwenden, der auf dem letzten Ereignis des letzten Ergebnisset basiert.

Größeneigenschaften

Die bei einigen API-Aufrufen verwendeten Eingabewerte sowie bestimmte Abfrageparameter sind numerisch. Anstatt eine ganze Zahl in Byte bereitzustellen, können Sie optional ein Suffix wie in der folgenden Tabelle aufgeführt verwenden.

Suffix	Beschreibung
KB	KB-Kilobyte (1024 Byte) oder Kibibyte
MB	MB Megabyte (KB x 1024 Byte) oder Mebibyte
GB	GB Gigabyte (MB x 1024 Byte) oder Gibibyte
TB	TB Terabyte (GB x 1024 byes) oder Tebibyte
PB	PB (TB x 1024 byes) oder Pebibyte

Interpretation einer API-Antwort

Jede API-Anfrage generiert eine Antwort an den Client. Sie sollten die Antwort überprüfen, um festzustellen, ob sie erfolgreich war, und weitere Daten nach Bedarf abrufen.

HTTP-Statuscode

Im Folgenden werden die von der SnapCenter REST API verwendeten HTTP-Statuscodes beschrieben.

Codieren	Beschreibung
200	OK zeigt Erfolg für Anrufe an, die kein neues Objekt erstellen.
201	Ein Objekt wurde erfolgreich erstellt. Der Positionskopf in der Antwort enthält die eindeutige Kennung für das Objekt.
202	Angenommen Ein Hintergrundjob wurde gestartet, um die Anfrage auszuführen, wurde aber noch nicht abgeschlossen.
400	Ungültige Anfrage die Eingabe der Anforderung wurde nicht erkannt oder ist nicht angemessen.
401	Nicht autorisierte Benutzerauthentifizierung fehlgeschlagen.
403	Aufgrund eines Autorisierungsfehlers (RBAC) wird der Zugriff verweigert.
404	Die Ressource, auf die in der Anfrage verwiesen wird, wurde nicht gefunden.
405	Methode nicht zulässig die HTTP-Methode in der Anfrage wird für die Ressource nicht unterstützt.
409	Konflikt ein Versuch, ein Objekt zu erstellen, ist fehlgeschlagen, weil zuerst ein anderes Objekt erstellt werden muss oder das angeforderte Objekt bereits vorhanden ist.
500	Interner Fehler Beim Server ist ein allgemeiner interner Fehler aufgetreten.

Antwortkopfzeilen

In der vom SnapCenter erzeugten HTTP-Antwort sind mehrere Header enthalten.

Standort

Wenn ein Objekt erstellt wird, enthält die Standortkopfzeile die komplette URL zum neuen Objekt einschließlich der eindeutigen Kennung, die dem Objekt zugewiesen ist.

Inhaltstyp

Dies ist normalerweise der Fall `application/json`.

Antwortkörper

Der Inhalt des Antwortkörpers, der sich aus einer API-Anfrage ergibt, unterscheidet sich je nach Objekt, Verarbeitungstyp und Erfolg oder Misserfolg der Anforderung. Die Antwort wird immer in JSON gerendert.

Einzelnes Objekt

Je nach Anforderung kann ein einzelnes Objekt mit einer Reihe von Feldern zurückgegeben werden. Beispielsweise können Sie GET verwenden, um ausgewählte Eigenschaften eines Clusters mit der eindeutigen Kennung abzurufen.

Mehrere Objekte

Es können mehrere Objekte aus einer Ressourcensammlung zurückgegeben werden. In allen Fällen wird ein konsistentes Format verwendet, mit `num_records` Angabe der Anzahl der Datensätze und Datensätze, die ein Array der Objektinstanzen enthalten. Beispielsweise können Sie die in einem bestimmten Cluster definierten Nodes abrufen.

Jobobjekt

Wenn ein API-Aufruf asynchron verarbeitet wird, wird ein Job-Objekt zurückgegeben, das den Hintergrund-Task ankers. Beispielsweise wird die PATCH-Anfrage, die zum Aktualisieren der Cluster-Konfiguration verwendet wird, asynchron verarbeitet und ein Job-Objekt zurückgegeben.

Fehlerobjekt

Wenn ein Fehler auftritt, wird immer ein Fehlerobjekt zurückgegeben. Beispielsweise erhalten Sie einen Fehler beim Versuch, ein Feld zu ändern, das nicht für ein Cluster definiert ist.

Leer

In bestimmten Fällen werden keine Daten zurückgegeben und der Antwortkörper enthält ein leeres JSON-Objekt.

Fehler

Wenn ein Fehler auftritt, wird ein Fehlerobjekt im Antwortkörper zurückgegeben.

Formatieren

Ein Fehlerobjekt hat das folgende Format:

```
"error": {
"message": "<string>",
"code": <integer>[,
"target": "<string>"]
}
```

Sie können den Codewert verwenden, um den allgemeinen Fehlertyp oder die allgemeine Fehlerkategorie zu bestimmen, und die Meldung, um den spezifischen Fehler zu ermitteln. Wenn verfügbar, enthält das Zielfeld die spezifische Benutzereingabe, die mit dem Fehler verknüpft ist.

Allgemeine Fehlercodes

Die gängigen Fehlercodes werden in der folgenden Tabelle beschrieben. Spezifische API-Aufrufe können zusätzliche Fehlercodes enthalten.

Codieren	Beschreibung
409	Ein Objekt mit derselben Kennung ist bereits vorhanden.
400	Der Wert für ein Feld hat einen ungültigen Wert oder fehlt oder es wurde ein zusätzliches Feld angegeben.
400	Der Vorgang wird nicht unterstützt.
405	Ein Objekt mit der angegebenen Kennung wurde nicht gefunden.
403	Die Berechtigung zur Durchführung der Anforderung wird verweigert.
409	Die Ressource wird verwendet.

Unterstützte REST-APIs

REST-APIs werden für andere Plug-ins unterstützt

Die über die SnapCenter REST API verfügbaren Ressourcen sind nach Kategorien sortiert, die auf der Dokumentationsseite der SnapCenter-API angezeigt werden. Nachfolgend finden Sie eine kurze Beschreibung der einzelnen Ressourcen mit den grundlegenden Ressourcenpfaden sowie weitere Nutzungsüberlegungen.

Auth

Sie können diesen API-Aufruf verwenden, um sich beim SnapCenter-Server anzumelden. Diese API gibt ein Benutzerautorisierungs-Token zurück, das zur Authentifizierung weiterer Anforderungen verwendet wird.

Domänen

Sie können die folgenden API-Aufrufe verwenden:

- Rufen Sie alle Domänen ab
- Abrufen von Details einer bestimmten Domäne
- Registrieren oder Aufheben der Registrierung einer Domain
- Ändern einer Domäne

Jobs

Sie können die folgenden API-Aufrufe verwenden:

- Rufen Sie alle Jobs ab
- Abrufen des Status eines Jobs
- Einen Job abbrechen oder beenden

Einstellungen

Sie können die folgenden API-Aufrufe verwenden:

- Registrieren, Anzeigen, Ändern oder Entfernen von Anmeldeinformationen
- Benachrichtigungseinstellungen konfigurieren

Hosts

Sie können die folgenden API-Aufrufe verwenden:

- Abrufen der Host-Details
- Abrufen der Details des installierten Plug-ins und der zugehörigen Ressourcen
- Hinzufügen, Entfernen oder Ändern eines Plug-in-Hosts
- Plug-ins installieren oder aktualisieren

Ressourcen

Sie können die folgenden API-Aufrufe verwenden:

- Abrufen von Ressourcen
- Ressourcen erstellen, ändern oder entfernen
- Schützen Sie eine Ressource
- Sichern, Wiederherstellen oder Klonen von Ressourcen

Backups

Sie können die folgenden API-Aufrufe verwenden:

- Rufen Sie Backup-Details ab
- Backups umbenennen oder löschen

Klone

Sie können die folgenden API-Aufrufe verwenden:

- Abrufen von Klondetails
- Klone löschen

Klonesplit

Sie können die folgenden API-Aufrufe verwenden:

- Abrufen des Status eines geteilten Klonvorgangs
- Starten oder stoppen Sie einen Klon-Split-Vorgang

Ressourcengruppen

Sie können die folgenden API-Aufrufe verwenden:

- Abrufen von Details einer Ressourcengruppe
- Erstellen, Ändern oder Löschen einer Ressourcengruppe
- Sichern einer Ressourcengruppe

Richtlinien

Sie können die folgenden API-Aufrufe verwenden:

- Abrufen von Richtliniendetails
- Erstellen, Ändern oder Löschen von Richtlinien

Storage

Sie können die folgenden API-Aufrufe verwenden:

- Abrufen von Storage-Details
- Erstellen, Ändern oder Löschen von Speicher
- Entdecken Sie Ressourcen auf einem Storage
- Erstellen oder löschen Sie eine Freigabe auf dem Speicher

Share

Sie können die folgenden API-Aufrufe verwenden:

- Rufen Sie die Details einer Freigabe ab
- Erstellen oder löschen Sie eine Freigabe auf dem Speicher

Plug-Ins

Mit diesen API-Aufrufen können Sie alle Plug-ins auf einem Host abrufen und verschiedene Vorgänge durchführen.

Berichte An

Sie können die folgenden API-Aufrufe verwenden:

- Erstellung von Berichten für Backup, Restore, Klonen und Plug-in
- Hinzufügen, Ausführen, Löschen oder Ändern von Zeitplänen

Meldungen

Sie können die folgenden API-Aufrufe verwenden:

- Rufen Sie alle Meldungen ab
- Meldungen löschen

Rbac

Sie können die folgenden API-Aufrufe verwenden:

- Abrufen von Details zu Benutzern, Gruppen und Rollen
- Benutzer hinzufügen
- Erstellen, Ändern oder Löschen von Rollen
- Rollen und Gruppen zuweisen oder deren Zuordnung aufheben

Konfiguration

Sie können die folgenden API-Aufrufe verwenden:

- Zeigen Sie die Konfigurationseinstellungen an
- Ändern Sie die Konfigurationseinstellungen

Zertifikateinstellungen

Sie können die folgenden API-Aufrufe verwenden:

- Zeigen Sie den Zertifikatsstatus an
- Ändern Sie die Zertifikatseinstellungen

Repository

Sie können die folgenden API-Aufrufe verwenden:

- Sicherung und Wiederherstellung des NSM-Repositorys
- Schutz und Schutz des NSM-Repositorys
- Failover
- Stellen Sie das NSM-Repository wieder her

REST-APIs werden für Oracle Database unterstützt

Die Oracle Datenbank verwendet REST-APIs, um die folgenden Operationen auszuführen:

Anmeldedaten

- Registrieren Sie die Anmeldedaten für den SnapCenter-Server
- Abrufen der mit SnapCenter-Server registrierten Anmeldedaten
- Anmeldedaten nach Namen abrufen
- Anmeldedaten ändern
- Anmeldedaten löschen

Host

- Fügen Sie Host hinzu
- Entfernen Sie einen oder mehrere Hosts aus SnapCenter
- Rufen Sie einen Host nach Namen ab

Plug-in

- Installation von Plug-ins auf vorhandenem Host
- Upgrade Plug-in

Richtlinie

- Erstellen einer neuen Richtlinie
- Richtlinie nach Namen abrufen
- Richtlinie ändern
- Löschen der Richtlinie

Ressourcengruppen

- Ressourcengruppe schützen
- Schutz ändern
- Schutz entfernen
- Neue Ressourcengruppe erstellen
- Ressourcengruppe nach Namen abrufen
- Ressourcengruppe ändern
- Ressourcengruppe löschen

Ressourcen

- Erkennen von Datenbanken
- Konfigurieren Sie die Datenbank

Backups

- Starten Sie eine Ressourcensicherung
- Starten Sie einen Sicherungsvorgang
- Abrufen von Backups nach Namen
- Abrufen von Backups
- Löschen Sie ein Backup mit Namen

Wiederherstellen

- Stellen Sie ein Backup wieder her

Klonaktualisierung

- Erstellen Sie eine Oracle Database Clone Specification aus einem angegebenen Backup
- Anzeigen, Ändern und Löschen von klonspezifischen Dateien
- Klonaktualisierung

RBAC

- Weisen Sie einem Benutzer oder einer Gruppe Ressourcen zu
- Heben Sie die Zuweisung von Ressourcen zu einem Benutzer oder einer Gruppe auf

Konfigurationseinstellungen

- Abrufen und Ändern von Konfigurationseinstellungen

REST API wird für Disaster Recovery von SnapCenter Server unterstützt

Die SnapCenter Disaster Recovery-Funktion (DR) verwendet REST-APIs für das Backup des SnapCenter Servers. Mithilfe VON REST-APIs können Sie die folgenden Operationen auf der SEITE REST APIs Swagger ausführen. Informationen zum Zugriff auf die Seite Swagger finden Sie unter "Zugriff auf REST-APIs über die Swagger-API-Webseite".

Was Sie brauchen

- Sie sollten sich als SnapCenter Admin-Benutzer einloggen.
- Der SnapCenter-Server sollte betriebsbereit sein, um die DR-Wiederherstellungs-API auszuführen.

Über diese Aufgabe

SnapCenter Server DR unterstützt alle Plug-ins.

Schritt	Beschreibung	REST API	HTTP-Methode
1	Holen Sie vorhandene DR-Backups des SnapCenter Servers Sie müssen den Backup-Namen und den Zielpfad angeben, in dem die DR-Backups gespeichert werden müssen.	`/4.5/disasterrecovery/server/backup?targetpath={path}`	GET
2	Erstellen Sie ein neues Server-DR-Backup. Stellt einen SnapCenter-Server aus einem festgelegten Server-DR-Backup wieder her.	`/4.5/disasterrecovery/server/backup`	POST

Schritt	Beschreibung	REST API	HTTP-Methode
3	Stellt einen SnapCenter-Server aus einem festgelegten Server-DR-Backup wieder her. **Voraussetzungen** • Der Name des alternativen Serverhosts sollte mit dem primären Server identisch sein, die IP-Adresse kann jedoch unterschiedlich sein. • Die Serverversion sollte die gleiche sein wie der primäre Server. • Der Host-Name sollte mit dem primären Server identisch sein. • Stellen Sie sicher, dass die DR-Backup-Dateien mit dem folgenden Befehl auf den neuen SnapCenter-Server kopiert werden: `xcopy <Ssource_Path> \\<Destination_Server_IP> \<Folder_Path> /O /X /E /H /K {ex : xcopy C:\DRBackup \\10.225.81.114\c$\DRBackup /O /X /E /H /K}` Wenn das Plug-in den Hostnamen des Servers nicht lösen kann, melden Sie sich bei jedem Plug-in-Host an und fügen Sie den etc/Host-Eintrag für die neue IP im Format hinzu: `<New IP> SC_Server_Name` Beispiel: 10.225.81.35 SCServer1 Server etc/Host-Einträge werden nicht wiederhergestellt. Sie können sie manuell aus dem DR-Backup-Ordner wiederherstellen.	`/4.5/disasterrecovery/server/restore`	POST
4	Löschen Sie das Server-DR-Backup auf Basis des Backup-Namens.	`/4.5/disasterrecovery/server/backup`	Löschen

Schritt	Beschreibung	REST API	HTTP-Methode
5	Aktiviert oder deaktiviert die Speicher-DR	`/4.5/disasterrecovery/storage`	POST

Weitere Informationen finden Sie im "Disaster Recovery-APIs" Video:

Zugriff auf REST-APIs über die Swagger API-Webseite

REST-APIs sind über die Swagger Webseite zugänglich. Sie können auf die Swagger-Webseite zugreifen, um die REST-APIs des SnapCenter-Servers anzuzeigen und einen API-Aufruf manuell auszuführen. MIT REST-APIs können Sie Ihren SnapCenter Server managen oder Datensicherungsvorgänge ausführen.

Sie sollten die Management-IP-Adresse oder den Domain-Namen des SnapCenter Servers kennen, auf dem Sie die REST-APIs ausführen möchten.

Für die Ausführung des REST-API-Clients sind keine speziellen Berechtigungen erforderlich. Jeder Benutzer kann auf die Swagger Webseite zugreifen. Die entsprechenden Berechtigungen für die Objekte, auf die über DIE REST-API zugegriffen wird, basieren auf dem Benutzer, der das Token generiert, um sich bei DER REST-API anzumelden.

Schritte

1. Geben Sie in einem Browser die URL ein, um die Swagger Webseite im Format *https://<SnapCenter_IP_address_or_name>:<SnapCenter_port>/swagger/* aufzurufen.

   Stellen Sie sicher, dass die REST-API-URL nicht die folgenden Zeichen hat: +, ., % und &.

2. Wenn die Dokumentation der Swagger API nicht automatisch angezeigt wird, geben Sie im Feld **Swagger Explore** *https://<SnapCenter_IP_address_or_name>:<SnapCenter_port>/Content/swagger/SnapCenter.yaml* ein
3. Klicken Sie Auf **Entdecken**.

   Es wird eine Liste der API-Ressourcentypen oder -Kategorien angezeigt.

4. Klicken Sie auf einen API-Ressourcentyp, um die APIs in diesem Ressourcentyp anzuzeigen.

Wenn bei der Ausführung von SnapCenter REST-APIs unerwartetes Verhalten auftritt, können Sie mithilfe der Protokolldateien die Ursache identifizieren und das Problem beheben. Sie können die Protokolldateien von der SnapCenter Benutzeroberfläche herunterladen, indem Sie auf **Monitor** > **Protokolle** > **Download** klicken.

Legen Sie los mit DER REST API

Die SnapCenter REST API ist ein schneller Einstieg. Der Zugriff auf die API bietet eine gewisse Perspektive, bevor Sie mit den komplexeren Workflow-Prozessen bei Live-Einrichtung beginnen.

Hallo Welt

Sie können einen einfachen Befehl auf Ihrem System ausführen, um die SnapCenter REST API zu verwenden und deren Verfügbarkeit zu bestätigen.

Was Sie brauchen

- Stellen Sie sicher, dass das Curl-Dienstprogramm auf Ihrem System verfügbar ist.
- IP-Adresse oder Hostname des SnapCenter-Servers
- Benutzername und Passwort für ein Konto mit Berechtigung für den Zugriff auf die SnapCenter REST API.

> Wenn Ihre Anmeldeinformationen Sonderzeichen enthalten, müssen Sie diese auf der Grundlage der verwendeten Shell so formatieren, dass sie für Curl akzeptabel sind. Sie können z. B. vor jedem Sonderzeichen einen umgekehrten Schrägstrich einfügen oder den gesamten Buchstaben umwickeln `username:password` Zeichenfolge in einzelnen Anführungszeichen.

Schritt

Führen Sie bei der Befehlszeilenschnittstelle Folgendes aus, um die Plug-in-Informationen abzurufen:

```
curl -X GET -u username:password -k
"https://<ip_address>/api/hosts?fields=IncludePluginInfo"
```

Beispiel:

```
curl -X GET -u admin:password -k
"'https://10.225.87.97/api/hosts?fields=IncludePluginInfo"
```

Copyright-Informationen

Copyright © 2024 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFTE SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGENDEINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter http://www.netapp.com/TM aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.