



# **REST API를 사용하여 자동화**

## SnapCenter software

NetApp  
November 06, 2025

# 목차

REST API를 사용하여 자동화	1
REST API를 사용한 SnapCenter 자동화	1
SnapCenter REST API에 기본적으로 액세스하는 방법	1
REST 웹 서비스 기반	1
자원과 국가 대표	1
URI 엔드포인트	2
HTTP 메시지	2
JSON 포맷팅	2
기본 작동 특성	2
요청 및 응답 API 트랜잭션	2
CRUD 작업 지원	2
객체 식별자	3
객체 인스턴스 및 컬렉션	3
동기 및 비동기 작업	3
보안	3
API 요청을 제어하는 입력 변수	4
HTTP 메서드	4
요청 헤더	4
요청 본문	4
객체 필터링	5
특정 객체 필드 요청	5
출력 세트에서 객체 정렬	6
컬렉션에서 객체를 검색할 때의 페이지 매김	6
크기 속성	7
API 응답 해석	7
HTTP 상태 코드	7
응답 헤더	8
대응 기관	8
오류	9
SnapCenter 서버 및 플러그인에 지원되는 REST API	9
인증	9
도메인	9
일자리	10
설정	10
호스트	10
리소스	11
백업	12
클론	13
클론 분할	13

리소스 그룹	13
정책	14
스토리지	14
공유하다	14
플러그인	15
보고서	16
알림	16
알박	16
구성	16
인증서 설정	17
저장소	17
버전	17
Swagger API 웹 페이지를 사용하여 REST API에 액세스하는 방법	17
REST API 시작하기	18
안녕하세요 세상	18

# REST API를 사용하여 자동화

## REST API를 사용한 SnapCenter 자동화

REST API를 사용하여 다양한 SnapCenter 관리 작업을 수행할 수 있습니다. REST API는 Swagger 웹 페이지를 통해 공개됩니다. REST API 문서를 표시하고 수동으로 API 호출을 실행하려면

`\https://<SnapCenter_IP_address_or_name>:<SnapCenter_port>/swagger/_`에서 제공되는 Swagger 웹 페이지에 액세스할 수 있습니다.

REST API를 지원하는 플러그인은 다음과 같습니다.

- Microsoft SQL Server용 플러그인
- SAP HANA 데이터베이스용 플러그인
- Oracle 데이터베이스용 플러그인

SnapCenter Plug-in for VMware vSphere에 대한 정보는 다음을 참조하세요. "["SnapCenter Plug-in for VMware vSphere"](#)"

## SnapCenter REST API에 기본적으로 액세스하는 방법

REST 클라이언트를 지원하는 모든 프로그래밍 언어를 사용하여 SnapCenter REST API에 직접 액세스할 수 있습니다. 인기 있는 언어로는 Python, PowerShell, Java 등이 있습니다.

## REST 웹 서비스 기반

REST(Representational State Transfer)는 분산 웹 애플리케이션을 구축하는 스타일입니다. 웹 서비스 API 설계에 적용되면 서버 기반 리소스를 노출하고 상태를 관리하는 데 필요한 기술과 모범 사례를 확립합니다. SnapCenter 관리하기 위한 유연한 기반을 제공하기 위해 주류 프로토콜과 표준을 사용합니다.

### 자원과 국가 대표

리소스는 웹 기반 시스템의 기본 구성 요소입니다. REST 웹 서비스 애플리케이션을 구축할 때 초기 설계 작업은 다음과 같습니다.

#### 시스템 또는 서버 기반 리소스 식별

모든 시스템은 리소스를 사용하고 유지관리합니다. 리소스는 파일, 비즈니스 트랜잭션, 프로세스 또는 관리 개체일 수 있습니다. REST 웹 서비스 기반 애플리케이션을 설계할 때 가장 먼저 해야 할 작업 중 하나는 리소스를 식별하는 것입니다.

#### 리소스 상태 및 관련 상태 작업의 정의

자원은 항상 유한한 수의 상태 중 하나에 머물러 있습니다. 상태와 상태 변경에 영향을 미치는 관련 작업은 명확하게

정의되어야 합니다.

## URI 엔드포인트

모든 REST 리소스는 명확하게 정의된 주소 지정 체계를 사용하여 정의하고 제공해야 합니다. 리소스가 위치하고 식별되는 엔드포인트는 URI(Uniform Resource Identifier)를 사용합니다.

URI는 네트워크의 각 리소스에 고유한 이름을 생성하는 일반적인 프레임워크를 제공합니다. URL(Uniform Resource Locator)은 웹 서비스에서 리소스를 식별하고 액세스하는 데 사용되는 URI 유형입니다. 리소스는 일반적으로 파일 디렉터리와 유사한 계층 구조로 노출됩니다.

## HTTP 메시지

HTTP(Hypertext Transfer Protocol)는 웹 서비스 클라이언트와 서버가 리소스에 대한 요청 및 응답 메시지를 교환하는데 사용하는 프로토콜입니다.

웹 서비스 애플리케이션을 설계하는 과정에서 HTTP 메서드는 리소스와 해당 상태 관리 작업에 매핑됩니다. HTTP는 상태를 저장하지 않습니다. 따라서 관련된 요청과 응답 세트를 하나의 트랜잭션의 일부로 연관시키려면 요청 및 응답 데이터 헤더와 함께 전달되는 HTTP 헤더에 추가 정보를 포함해야 합니다.

## JSON 포맷팅

웹 서비스 클라이언트와 서버 간에 정보를 구성하고 전송하는 방법은 여러 가지가 있지만, 가장 인기 있는 옵션은 JSON(JavaScript Object Notation)입니다.

JSON은 간단한 데이터 구조를 일반 텍스트로 표현하는 업계 표준이며, 리소스를 설명하는 상태 정보를 전송하는 데 사용됩니다. SnapCenter REST API는 JSON을 사용하여 각 HTTP 요청 및 응답 본문에 포함된 데이터를 형식화합니다.

## 기본 작동 특성

REST는 공통된 기술과 모범 사례를 제공하지만, 각 API의 세부 사항은 설계 방식에 따라 달라질 수 있습니다.

### 요청 및 응답 API 트랜잭션

모든 REST API 호출은 SnapCenter 서버 시스템에 대한 HTTP 요청으로 수행되며, 이를 통해 클라이언트에 대한 관련 응답이 생성됩니다. 이 요청과 응답 쌍은 API 트랜잭션으로 간주됩니다.

API를 사용하기 전에 요청을 제어하는 데 사용할 수 있는 입력 변수와 응답 출력의 내용을 잘 알고 있어야 합니다.

## CRUD 작업 지원

SnapCenter REST API를 통해 사용 가능한 각 리소스는 CRUD 모델을 기반으로 액세스됩니다.

- 만들다
- 읽다
- 업데이트

- 삭제

일부 리소스의 경우 작업의 하위 집합만 지원됩니다.

## 객체 식별자

각 리소스 인스턴스 또는 객체는 생성될 때 고유 식별자가 할당됩니다. 대부분의 경우 식별자는 128비트 UUID입니다. 이러한 식별자는 특정 SnapCenter 서버 내에서 전역적으로 고유합니다.

새로운 객체 인스턴스를 생성하는 API 호출을 발행한 후, 연관된 ID가 있는 URL이 HTTP 응답의 위치 헤더에 포함되어 호출자에게 반환됩니다. 식별자를 추출하여 리소스 인스턴스를 참조할 때 후속 호출에서 사용할 수 있습니다.



객체 식별자의 내용과 내부 구조는 언제든지 변경될 수 있습니다. 연관된 객체를 참조할 때 해당 API 호출에서 필요에 따라서만 식별자를 사용해야 합니다.

## 객체 인스턴스 및 컬렉션

리소스 경로와 HTTP 메서드에 따라 API 호출은 특정 개체 인스턴스나 개체 컬렉션에 적용될 수 있습니다.

### 동기 및 비동기 작업

SnapCenter 클라이언트로부터 수신한 HTTP 요청을 동기적 또는 비동기적으로 수행합니다.

#### 동기 처리

SnapCenter 요청을 즉시 수행하고 성공하면 HTTP 상태 코드 200 또는 201로 응답합니다.

GET 메서드를 사용하는 모든 요청은 항상 동기적으로 수행됩니다. 또한 POST를 사용하는 요청은 2초 이내에 완료될 것으로 예상되는 경우 동기적으로 실행되도록 설계되었습니다.

#### 비동기 처리

비동기 요청이 유효한 경우 SnapCenter 요청을 처리하기 위한 백그라운드 작업과 작업을 고정하기 위한 작업 객체를 생성합니다. HTTP 상태 코드 202는 작업 객체와 함께 호출자에게 반환됩니다. 성공 또는 실패를 확인하려면 작업 상태를 검색해야 합니다.

POST 및 DELETE 메서드를 사용하는 요청은 완료하는 데 2초 이상 걸릴 것으로 예상되는 경우 비동기적으로 실행되도록 설계되었습니다.

## 보안

REST API가 제공하는 보안은 주로 SnapCenter에서 제공하는 기존 보안 기능을 기반으로 합니다. API에서는 다음과 같은 보안을 사용합니다.

### 전송 계층 보안

SnapCenter 서버와 클라이언트 간 네트워크를 통해 전송되는 모든 트래픽은 일반적으로 SnapCenter 구성 설정에 따라 TLS를 사용하여 암호화됩니다.

## HTTP 인증

HTTP 수준에서는 API 트랜잭션에 기본 인증이 사용됩니다. 사용자 이름과 비밀번호를 base64 문자열로 담은 HTTP 헤더가 각 요청에 추가됩니다.

## API 요청을 제어하는 입력 변수

HTTP 요청에 설정된 매개변수와 변수를 통해 API 호출이 처리되는 방식을 제어할 수 있습니다.

### HTTP 메서드

SnapCenter REST API에서 지원하는 HTTP 메서드는 다음 표에 나와 있습니다.



각 REST 엔드포인트에서 모든 HTTP 메서드를 사용할 수 있는 것은 아닙니다.

HTTP 메서드	설명
얻다	리소스 인스턴스 또는 컬렉션에서 개체 속성을 검색합니다.
우편	제공된 입력을 기반으로 새로운 리소스 인스턴스를 생성합니다.
삭제	기존 리소스 인스턴스를 삭제합니다.
놓다	기존 리소스 인스턴스를 수정합니다.

### 요청 헤더

HTTP 요청에는 여러 개의 헤더를 포함해야 합니다.

#### 콘텐츠 유형

요청 본문에 JSON이 포함되어 있는 경우 이 헤더를 `_application/json_`으로 설정해야 합니다.

#### 수용하다

이 헤더는 `_application/json_`으로 설정되어야 합니다.

#### 권한 부여

기본 인증은 사용자 이름과 비밀번호를 base64 문자열로 인코딩하여 설정해야 합니다.

### 요청 본문

요청 본문의 내용은 특정 호출에 따라 다릅니다. HTTP 요청 본문은 다음 중 하나로 구성됩니다.

- 입력 변수가 있는 JSON 객체
- 비어 있는

## 객체 필터링

GET을 사용하는 API 호출을 실행할 때 반환되는 객체를 속성에 따라 제한하거나 필터링할 수 있습니다. 예를 들어, 일치시킬 정확한 값을 지정할 수 있습니다.

<field>=<query value>

정확한 일치 외에도 다양한 연산자를 사용하여 값 범위에 대한 객체 집합을 반환할 수 있습니다. SnapCenter REST API는 아래 표에 표시된 필터링 연산자를 지원합니다.

연산자	설명
=	동일하다
<	미만
>	보다 크다
≤	이하
≥	이상 또는 같음
업데이트	또는
!	같지 않다
*	탐욕스러운 와일드카드

**null** 키워드나 부정 \*!null\*을 쿼리의 일부로 사용하면 특정 필드가 설정되었는지 여부에 따라 개체 컬렉션을 반환할 수도 있습니다.



설정되지 않은 필드는 일반적으로 일치하는 쿼리에서 제외됩니다.

## 특정 객체 필드 요청

기본적으로 GET을 사용하여 API 호출을 실행하면 객체를 고유하게 식별하는 속성만 반환됩니다. 이 최소 필드 집합은 각 객체의 키 역할을 하며 객체 유형에 따라 달라집니다. 다음을 사용하여 추가 개체 속성을 선택할 수 있습니다. **fields** 다음과 같은 방법으로 쿼리 매개변수를 사용합니다.

### 공통 또는 표준 필드

가장 일반적으로 사용되는 객체 필드를 검색하려면 **fields=\***를 지정합니다. 이러한 필드는 일반적으로 로컬 서버 메모리에 유지되거나 액세스하는 데 거의 처리가 필요하지 않습니다. 이는 URL 경로 키(UUID)와 함께 GET을 사용한 후 객체에 대해 반환되는 속성과 동일합니다.

### 모든 분야

**fields=\*\***를 지정하면 추가 서버 처리가 필요한 필드를 포함하여 모든 개체 필드를 검색합니다.

### 사용자 정의 필드 선택

원하는 정확한 필드를 지정하려면 **\*fields=<field\_name>\***을 사용하세요. 여러 필드를 요청할 경우, 공백 없이 쉼표로 값을 구분해야 합니다.



의 필요할 때만 공통 필드 세트나 모든 필드를 검색해야 합니다. 어떤 필드가 공통으로 분류되고 `_fields=*`을 사용하여 반환되는지는 NetApp에서 내부 성능 분석을 기반으로 결정합니다. 필드 분류는 향후 릴리스에서 변경될 수 있습니다.

## 출력 세트에서 객체 정렬

리소스 컬렉션의 레코드는 객체에 정의된 기본 순서대로 반환됩니다. 다음을 사용하여 순서를 변경할 수 있습니다. `order_by` 필드 이름과 정렬 방향을 포함하는 쿼리 매개변수는 다음과 같습니다.

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

예를 들어, 유형 필드를 내림차순으로 정렬한 다음 ID를 오름차순으로 정렬할 수 있습니다.

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

- 정렬 필드를 지정했지만 방향을 제공하지 않으면 값이 오름차순으로 정렬됩니다.
- 여러 개의 매개변수를 포함하는 경우 필드를 쉼표로 구분해야 합니다.

## 컬렉션에서 객체를 검색할 때의 페이지 매김

GET을 사용하여 동일한 유형의 객체 컬렉션에 액세스하기 위한 API 호출을 발행할 때 SnapCenter 두 가지 제약 조건에 따라 가능한 한 많은 객체를 반환하려고 시도합니다. 요청에 대한 추가 쿼리 매개변수를 사용하여 이러한 각 제약 조건을 제어할 수 있습니다. 특정 GET 요청에 대해 도달한 첫 번째 제약 조건은 요청을 종료하고 따라서 반환되는 레코드 수를 제한합니다.



모든 객체를 반복하기 전에 요청이 종료되면 응답에는 다음 레코드 배치를 검색하는 데 필요한 링크가 포함됩니다.

### 객체 수 제한

기본적으로 SnapCenter GET 요청에 대해 최대 10,000개의 객체를 반환합니다. `max_records` 쿼리 매개변수를 사용하여 이 제한을 변경할 수 있습니다. 예를 들어:

```
max_records=20
```

실제로 반환되는 객체의 수는 관련 시간 제약과 시스템의 총 객체 수에 따라 적용되는 최대값보다 적을 수 있습니다.

### 객체를 검색하는 데 사용되는 시간 제한

기본적으로 SnapCenter GET 요청에 허용된 시간 내에 가능한 한 많은 객체를 반환합니다. 기본 제한 시간은 15초입니다. `return_timeout` 쿼리 매개변수를 사용하여 이 제한을 변경할 수 있습니다. 예를 들어:

```
return_timeout=5
```

실제로 반환되는 객체의 수는 시스템의 객체 수와 관련된 제약 조건 및 객체 총 수에 따라 적용되는 최대값보다 적을 수 있습니다.

### 결과 집합 줍히기

필요한 경우 이 두 매개변수를 추가 쿼리 매개변수와 결합하여 결과 집합을 줍힐 수 있습니다. 예를 들어, 다음은 지정된

시간 이후에 생성된 최대 10개의 EMS 이벤트를 반환합니다.

time⇒ 2018-04-04T15:41:29.140265Z&max\_records=10

여러 개의 요청을 발행하여 객체를 탐색할 수 있습니다. 이후의 각 API 호출은 마지막 결과 집합의 최신 이벤트를 기반으로 새로운 시간 값을 사용해야 합니다.

## 크기 속성

일부 API 호출과 특정 쿼리 매개변수에 사용되는 입력 값은 숫자형입니다. 바이트 단위의 정수를 제공하는 대신 다음 표에 표시된 것처럼 접미사를 사용할 수도 있습니다.

접미사	설명
KB	KB 킬로바이트(1024바이트) 또는 키비바이트
엠비	MB 메가바이트(KB x 1024바이트) 또는 메비바이트
영국	GB 기가바이트(MB x 1024바이트) 또는 기비바이트
결핵	TB 테라바이트(GB x 1024바이트) 또는 테비바이트
피비	PB 페타바이트(TB x 1024바이트) 또는 페비바이트

## API 응답 해석

각 API 요청은 클라이언트에게 응답을 생성합니다. 응답을 검토하여 성공했는지 확인하고 필요에 따라 추가 데이터를 검색해야 합니다.

## HTTP 상태 코드

SnapCenter REST API에서 사용하는 HTTP 상태 코드는 아래와 같습니다.

암호	설명
200	OK 새로운 객체를 생성하지 않는 호출의 경우 성공을 나타냅니다.
201	생성됨 객체가 성공적으로 생성되었습니다. 응답의 위치 헤더에는 객체의 고유 식별자가 포함됩니다.
202	수락됨 요청을 수행하기 위한 백그라운드 작업이 시작되었지만 아직 완료되지 않았습니다.
400	잘못된 요청 요청 입력이 인식되지 않거나 부적절합니다.
401	권한이 없는 사용자 인증에 실패했습니다.
403	권한 부여(RBAC) 오류로 인해 금지된 액세스가 거부되었습니다.
404	찾을 수 없음 요청에서 언급한 리소스가 존재하지 않습니다.
405	허용되지 않는 메서드 요청의 HTTP 메서드는 해당 리소스에 대해 지원되지 않습니다.

암호	설명
409	충돌 객체를 생성하려는 시도는 다른 객체를 먼저 생성해야하거나 요청된 객체가 이미 존재하기 때문에 실패했습니다.
500	내부 오류 서버에서 일반적인 내부 오류가 발생했습니다.

## 응답 헤더

SnapCenter에서 생성된 HTTP 응답에는 여러 개의 헤더가 포함됩니다.

### 위치

객체가 생성되면 위치 헤더에는 객체에 할당된 고유 식별자를 포함하여 새 객체의 전체 URL이 포함됩니다.

### 콘텐츠 유형

이것은 일반적으로 다음과 같습니다. `application/json`.

## 대응 기관

API 요청으로 인해 생성되는 응답 본문의 내용은 객체, 처리 유형, 요청의 성공 또는 실패 여부에 따라 달라집니다. 응답은 항상 JSON으로 렌더링됩니다.

### 단일 객체

요청에 따라 필드 집합이 포함된 단일 객체가 반환될 수 있습니다. 예를 들어, GET을 사용하면 고유 식별자를 사용하여 클러스터의 선택된 속성을 검색할 수 있습니다.

### 여러 개체

리소스 컬렉션에서 여러 개의 개체가 반환될 수 있습니다. 모든 경우에 일관된 형식이 사용됩니다. `num_records` 레코드 수와 객체 인스턴스 배열을 포함하는 레코드 수를 나타냅니다. 예를 들어, 특정 클러스터에 정의된 노드를 검색할 수 있습니다.

### Job 객체

API 호출이 비동기적으로 처리되는 경우 백그라운드 작업을 고정하는 Job 객체가 반환됩니다. 예를 들어, 클러스터 구성을 업데이트하는 데 사용되는 PATCH 요청은 비동기적으로 처리되고 Job 객체를 반환합니다.

### 오류 객체

오류가 발생하면 항상 Error 객체가 반환됩니다. 예를 들어, 클러스터에 대해 정의되지 않은 필드를 변경하려고 하면 오류가 발생합니다.

### 비어 있는

어떤 경우에는 데이터가 반환되지 않고 응답 본문에 빈 JSON 개체가 포함됩니다.

## 오류

오류가 발생하면 응답 본문에 오류 객체가 반환됩니다.

### 체재

오류 객체의 형식은 다음과 같습니다.

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

코드 값을 사용하여 일반적인 오류 유형이나 범주를 파악하고, 메시지를 사용하여 특정 오류를 파악할 수 있습니다. 가능한 경우 대상 필드에는 오류와 관련된 특정 사용자 입력이 포함됩니다.

### 일반적인 오류 코드

일반적인 오류 코드는 다음 표에 설명되어 있습니다. 특정 API 호출에는 추가 오류 코드가 포함될 수 있습니다.

암호	설명
409	동일한 식별자를 가진 객체가 이미 존재합니다.
400	필드 값에 잘못된 값이 있거나 값이 누락되었거나 추가 필드가 제공되었습니다.
400	해당 작업은 지원되지 않습니다.
405	지정된 식별자를 가진 객체를 찾을 수 없습니다.
403	요청을 수행할 수 있는 권한이 거부되었습니다.
409	리소스가 사용 중입니다.

## SnapCenter 서버 및 플러그인에 지원되는 REST API

SnapCenter REST API를 통해 사용할 수 있는 리소스는 SnapCenter API 문서 페이지에 표시된 대로 범주별로 정리되어 있습니다. 아래에는 기본 리소스 경로를 포함한 각 리소스에 대한 간략한 설명이 나와 있으며, 필요한 경우 추가 사용 고려 사항도 나와 있습니다.

### 인증

이 API를 사용하여 SnapCenter 서버에 로그인할 수 있습니다. 이 API는 후속 요청을 인증하는 데 사용되는 사용자 인증 토큰을 반환합니다.

### 도메인

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- SnapCenter 의 모든 도메인을 검색합니다
- 특정 도메인의 세부 정보 검색
- 도메인 등록 또는 등록 취소
- 도메인 수정

## 일자리

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- SnapCenter 에서 모든 작업 검색
- 작업 상태 검색
- 작업을 취소하거나 중지하다

## 설정

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 자격 증명을 등록, 수정 또는 제거합니다.
- SnapCenter 서버에 등록된 자격 증명 정보를 표시합니다.
- 알림 설정 구성
- 현재 이메일 알림을 보내도록 구성된 SMTP 서버에 대한 정보를 검색하고 SMTP 서버 이름, 수신자 이름, 발신자 이름을 표시합니다.
- SnapCenter 서버 로그인의 다중 인증 요소(MFA) 구성을 표시합니다.
- SnapCenter 서버 로그인에 대한 MFA를 활성화하거나 비활성화하고 구성합니다.
- MFA 설정에 필요한 구성 파일을 만듭니다.

## 호스트

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 모든 SnapCenter 호스트 쿼리
- SnapCenter 에서 하나 이상의 호스트 제거
- 이름으로 호스트 검색
- 호스트의 모든 리소스 검색
- 리소스 ID를 사용하여 리소스를 검색합니다.
- 플러그인 구성 세부 정보 검색
- 플러그인 호스트 구성
- Microsoft SQL Server 호스트용 플러그인의 모든 리소스를 검색합니다.
- Oracle 데이터베이스 호스트용 플러그인의 모든 리소스를 검색합니다.
- 사용자 정의 애플리케이션 호스트에 대한 플러그인의 모든 리소스를 검색합니다.

- SAP HANA 호스트용 플러그인의 모든 리소스를 검색합니다.
- 설치된 플러그인을 검색합니다
- 기존 호스트에 플러그인 설치
- 호스트 패키지 업그레이드
- 기존 호스트에서 플러그인 제거
- 호스트에 플러그인 추가
- 호스트 추가 또는 수정
- Linux 호스트의 서명을 가져옵니다
- Linux 호스트의 서명을 등록합니다
- 호스트를 유지 관리 또는 프로덕션 모드로 전환
- 호스트에서 플러그인 서비스를 시작하거나 다시 시작합니다.
- 호스트 이름 바꾸기

## 리소스

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 모든 리소스 검색
- 리소스 ID를 사용하여 리소스를 검색합니다.
- Microsoft SQL Server 호스트용 플러그인의 모든 리소스를 검색합니다.
- Oracle 데이터베이스 호스트용 플러그인의 모든 리소스를 검색합니다.
- 사용자 정의 애플리케이션 호스트에 대한 플러그인의 모든 리소스를 검색합니다.
- SAP HANA 호스트용 플러그인의 모든 리소스를 검색합니다.
- 키를 사용하여 Microsoft SQL Server 리소스 검색
- 키를 사용하여 사용자 정의 리소스 검색
- 사용자 정의 애플리케이션 호스트에 대한 플러그인 리소스 수정
- 키를 사용하여 사용자 정의 애플리케이션 호스트의 플러그인 리소스를 제거합니다.
- 키를 사용하여 SAP HANA 리소스 검색
- SAP HANA 호스트용 플러그인 리소스 수정
- 키를 사용하여 SAP HANA 호스트용 플러그인 리소스 제거
- 키를 사용하여 Oracle 리소스 검색
- Oracle 애플리케이션 볼륨 리소스 생성
- Oracle 애플리케이션 볼륨 리소스 수정
- 키를 사용하여 Oracle 애플리케이션 볼륨 리소스 제거
- Oracle 리소스의 보조 세부 정보를 검색합니다.
- Microsoft SQL Server용 플러그인을 사용하여 Microsoft SQL Server 리소스 백업

- Oracle 데이터베이스용 플러그인을 사용하여 Oracle 리소스 백업
- 사용자 정의 애플리케이션용 플러그인을 사용하여 사용자 정의 리소스 백업
- SAP HANA 데이터베이스 구성
- Oracle 데이터베이스 구성
- SQL 데이터베이스 백업 복원
- Oracle 데이터베이스 백업 복원
- 사용자 정의 애플리케이션 백업 복원
- SAP HANA 리소스 생성
- 사용자 정의 애플리케이션용 플러그인을 사용하여 사용자 정의 리소스 보호
- Microsoft SQL Server용 플러그인을 사용하여 Microsoft SQL Server 리소스 보호
- 보호된 Microsoft SQL Server 리소스 수정
- Microsoft SQL Server 리소스에 대한 보호 제거
- Oracle 데이터베이스용 플러그인을 사용하여 Oracle 리소스 보호
- 보호된 Oracle 리소스 수정
- Oracle 리소스에서 보호 제거
- 사용자 정의 애플리케이션용 플러그인을 사용하여 백업에서 리소스 복제
- Oracle 데이터베이스용 플러그인을 사용하여 백업에서 Oracle 애플리케이션 볼륨 복제
- Microsoft SQL Server용 플러그인을 사용하여 백업에서 Microsoft SQL Server 리소스를 복제합니다.
- Microsoft SQL Server 리소스의 복제 수명 주기를 만듭니다.
- Microsoft SQL Server 리소스의 복제 수명 주기 수정
- Microsoft SQL Server 리소스의 복제 수명 주기 삭제
- 기존 Microsoft SQL Server 데이터베이스를 로컬 디스크에서 NetApp LUN으로 이동합니다.
- Oracle 데이터베이스에 대한 복제 사양 파일 생성
- Oracle 리소스의 주문형 복제 새로 고침 작업 시작
- 복제 사양 파일을 사용하여 백업에서 Oracle 리소스를 만듭니다.
- 데이터베이스를 보조 복제본으로 복원하고 데이터베이스를 가용성 그룹에 다시 조인합니다.
- Oracle 애플리케이션 볼륨 리소스 생성

## 백업

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 백업 이름, 유형, 플러그인, 리소스 또는 날짜별로 백업 세부 정보 검색
- 모든 백업 검색
- 백업 세부 정보 검색
- 백업 이름 바꾸기 또는 삭제

- Oracle 백업 마운트
- Oracle 백업 마운트 해제
- Oracle 백업 카탈로그화
- Oracle 백업 카탈로그 해제
- 특정 시점 복구를 수행하기 위해 마운트해야 하는 모든 백업을 가져옵니다.

## 클론

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- Oracle 데이터베이스 복제 사양 파일을 생성, 표시, 수정 및 삭제합니다.
- Oracle 데이터베이스 복제 계층 표시
- 클론 세부 정보 검색
- 모든 클론을 검색합니다
- 클론 삭제
- ID로 클론 세부 정보 검색
- Oracle 리소스의 주문형 복제 새로 고침 작업 시작
- 복제 사양 파일을 사용하여 백업에서 Oracle 리소스를 복제합니다.

## 클론 분할

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 복제된 리소스의 복제 분할 작업을 추정합니다.
- 복제 분할 작업의 상태를 검색합니다.
- 클론 분할 작업 시작 또는 중지

## 리소스 그룹

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 모든 리소스 그룹의 세부 정보 검색
- 이름으로 리소스 그룹을 검색합니다.
- 사용자 정의 애플리케이션용 플러그인에 대한 리소스 그룹을 생성합니다.
- Microsoft SQL Server용 플러그인에 대한 리소스 그룹을 만듭니다.
- Oracle 데이터베이스 플러그인에 대한 리소스 그룹 생성
- 사용자 정의 애플리케이션용 플러그인에 대한 리소스 그룹 수정
- Microsoft SQL Server용 플러그인에 대한 리소스 그룹 수정
- Oracle 데이터베이스 플러그인에 대한 리소스 그룹 수정
- Microsoft SQL Server용 플러그인 리소스 그룹의 복제 수명 주기를 생성, 수정 또는 삭제합니다.

- 리소스 그룹 백업
- 리소스 그룹을 유지 관리 또는 프로덕션 모드로 전환
- 리소스 그룹 제거

## 정책

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 정책 세부 정보 검색
- 이름으로 정책 세부 정보 검색
- 정책 삭제
- 기존 정책의 사본을 만듭니다
- 사용자 정의 애플리케이션용 플러그인에 대한 정책을 생성하거나 수정합니다.
- Microsoft SQL Server용 플러그인에 대한 정책을 만들거나 수정합니다.
- Oracle 데이터베이스용 플러그인에 대한 정책을 생성하거나 수정합니다.
- SAP HANA 데이터베이스용 플러그인에 대한 정책을 생성하거나 수정합니다.

## 스토리지

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 모든 주식을 검색합니다
- 이름으로 공유 검색
- 공유 만들기 또는 삭제
- 저장 세부 정보 검색
- 이름으로 저장소 세부 정보 검색
- 저장소 생성, 수정 또는 삭제
- 스토리지 클러스터에서 리소스 검색
- 스토리지 클러스터에서 리소스 검색

## 공유하다

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 공유 세부 정보를 검색합니다
- 모든 주식의 세부 정보를 검색합니다
- 저장소에서 공유를 생성하거나 삭제합니다.
- 이름으로 공유 검색

## 플러그인

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 호스트의 모든 플러그인을 나열합니다
- 키를 사용하여 Microsoft SQL Server 리소스 검색
- 키를 사용하여 사용자 정의 리소스 수정
- 키를 사용하여 사용자 정의 리소스 제거
- 키를 사용하여 SAP HANA 리소스 검색
- 키를 사용하여 SAP HANA 리소스 수정
- 키를 사용하여 SAP HANA 리소스 제거
- 키를 사용하여 Oracle 리소스 검색
- 키를 사용하여 Oracle 애플리케이션 볼륨 리소스 수정
- 키를 사용하여 Oracle 애플리케이션 볼륨 리소스 제거
- Microsoft SQL Server용 플러그인과 키를 사용하여 Microsoft SQL Server 리소스를 백업합니다.
- Oracle 데이터베이스용 플러그인과 키를 사용하여 Oracle 리소스를 백업합니다.
- 사용자 정의 애플리케이션용 플러그인과 키를 사용하여 사용자 정의 애플리케이션 리소스를 백업합니다.
- 키를 사용하여 SAP HANA 데이터베이스 구성
- 키를 사용하여 Oracle 데이터베이스 구성
- 키를 사용하여 사용자 정의 애플리케이션 백업 복원
- SAP HANA 리소스 생성
- Oracle 애플리케이션 볼륨 리소스 생성
- 사용자 정의 애플리케이션용 플러그인을 사용하여 사용자 정의 리소스 보호
- Microsoft SQL Server용 플러그인을 사용하여 Microsoft SQL Server 리소스 보호
- 보호된 Microsoft SQL Server 리소스 수정
- Microsoft SQL Server 리소스에 대한 보호 제거
- Oracle 데이터베이스용 플러그인을 사용하여 Oracle 리소스 보호
- 보호된 Oracle 리소스 수정
- Oracle 리소스에서 보호 제거
- 사용자 정의 애플리케이션용 플러그인을 사용하여 백업에서 리소스 복제
- Oracle 데이터베이스용 플러그인을 사용하여 백업에서 Oracle 애플리케이션 볼륨 복제
- Microsoft SQL Server용 플러그인을 사용하여 백업에서 Microsoft SQL Server 리소스를 복제합니다.
- Microsoft SQL Server 리소스의 복제 수명 주기를 만듭니다.
- Microsoft SQL Server 리소스의 복제 수명 주기 수정
- Microsoft SQL Server 리소스의 복제 수명 주기 삭제
- Oracle 데이터베이스에 대한 복제 사양 파일 생성

- Oracle 리소스의 주문형 복제 수명 주기를 시작합니다.
- 복제 사양 파일을 사용하여 백업에서 Oracle 리소스를 복제합니다.

## 보고서

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 각 플러그인에 대한 백업, 복원 및 복제 작업 보고서를 검색합니다.
- 일정을 추가, 실행, 삭제 또는 수정합니다.
- 예약된 보고서에 대한 데이터 검색

## 알림

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 모든 알림을 검색합니다
- ID로 알림 검색
- 여러 개의 알림을 삭제하거나 ID로 알림을 삭제합니다.

## 알박

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 사용자, 그룹 및 역할의 세부 정보를 검색합니다.
- 사용자 추가 또는 삭제
- 사용자에게 역할 할당
- 역할에서 사용자 할당 해제
- 역할 생성, 수정 또는 삭제
- 그룹에 역할 할당
- 역할에서 그룹 할당 해제
- 그룹 추가 또는 삭제
- 기존 역할의 사본을 만듭니다
- 사용자 또는 그룹에 리소스 할당 또는 할당 해제

## 구성

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 구성 설정 보기
- 구성 설정을 수정하다

## 인증서 설정

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- SnapCenter 서버 또는 플러그인 호스트의 인증서 상태를 확인합니다.
- SnapCenter 서버 또는 플러그인 호스트에 대한 인증서 설정을 수정합니다.

## 저장소

API를 사용하여 다양한 작업을 수행할 수 있습니다.

- 저장소 백업을 검색합니다
- 저장소에 대한 구성 정보를 봅니다.
- SnapCenter 저장소를 보호하고 복원합니다.
- SnapCenter 저장소 보호 해제
- 저장소를 다시 빌드하고 장애 조치합니다.

## 버전

이 API를 사용하면 SnapCenter 버전을 볼 수 있습니다.

## Swagger API 웹 페이지를 사용하여 REST API에 액세스하는 방법

REST API는 Swagger 웹 페이지를 통해 공개됩니다. Swagger 웹 페이지에 접속하면 SnapCenter Server REST API를 표시하고 수동으로 API 호출을 실행할 수 있습니다. REST API를 사용하면 SnapCenter 서버를 관리하거나 데이터 보호 작업을 수행하는 데 도움이 됩니다.

REST API를 실행하려는 SnapCenter 서버의 관리 IP 주소 또는 도메인 이름을 알아야 합니다.

REST API 클라이언트를 실행하는 데 특별한 권한이 필요하지 않습니다. 모든 사용자가 Swagger 웹 페이지에 접속할 수 있습니다. REST API를 통해 액세스하는 개체에 대한 각각의 권한은 REST API에 로그인하기 위해 토큰을 생성하는 사용자를 기반으로 합니다.

### 단계

1. 브라우저에서 Swagger 웹 페이지에 접속하기 위한 URL을 [https://<SnapCenter\\_IP\\_address\\_or\\_name>:<SnapCenter\\_port>/swagger/](https://<SnapCenter_IP_address_or_name>:<SnapCenter_port>/swagger/) 형식으로 입력합니다.  
 REST API URL에 +, ., %, & 등의 문자가 포함되지 않도록 하세요.
2. **Swagger Explore** 필드에서 Swagger API 설명서가 자동으로 표시되지 않으면 다음을 입력합니다.  
[https://<SnapCenter\\_IP\\_address\\_or\\_name>:<SnapCenter\\_port>/Content/swagger/\\_SnapCenter.yaml](https://<SnapCenter_IP_address_or_name>:<SnapCenter_port>/Content/swagger/_SnapCenter.yaml)
3. \*탐색\*을 클릭하세요.

API 리소스 유형 또는 범주 목록이 표시됩니다.

4. API 리소스 유형을 클릭하면 해당 리소스 유형의 API가 표시됩니다.

SnapCenter REST API를 실행할 때 예기치 않은 동작이 발생하는 경우 로그 파일을 사용하여 원인을 파악하고 문제를 해결할 수 있습니다. SnapCenter 사용자 인터페이스에서 모니터 > 로그 > \*다운로드\*를 클릭하여 로그 파일을 다운로드할 수 있습니다.

## REST API 시작하기

SnapCenter REST API를 사용하면 빠르게 시작할 수 있습니다. API에 액세스하면 라이브 설정에서 보다 복잡한 워크플로 프로세스를 사용하기 전에 어느 정도 관점을 얻을 수 있습니다.

### 안녕하세요 세상

시스템에서 간단한 명령을 실행하여 SnapCenter REST API를 사용하고 가용성을 확인할 수 있습니다.

#### 시작하기 전에

- Curl 유틸리티가 시스템에서 사용 가능한지 확인하세요.
- SnapCenter 서버의 IP 주소 또는 호스트 이름
- SnapCenter REST API에 액세스할 권한이 있는 계정의 사용자 이름과 비밀번호입니다.



자격 증명에 특수 문자가 포함되어 있는 경우, 사용하는 셀을 기준으로 Curl에서 허용하는 방식으로 형식을 지정해야 합니다. 예를 들어, 각 특수 문자 앞에 백슬래시를 삽입하거나 전체를 래핑할 수 있습니다. `username:password` 작은따옴표로 묶은 문자열.

#### 단계

명령줄 인터페이스에서 다음을 실행하여 플러그인 정보를 검색합니다.

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

예:

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

## 저작권 정보

Copyright © 2025 NetApp, Inc. All Rights Reserved. 미국에서 인쇄됨 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그레픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이센스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이센스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 있으며 재사용이 불가능하며 취소 불가능한 라이센스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

## 상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.