



개념

ONTAP Select

NetApp
January 29, 2026

목차

개념	1
ONTAP Select 클러스터 배포 및 관리를 위한 REST 웹 서비스 기반	1
건축과 고전적 제약	1
자원과 국가 대표	1
URI 엔드포인트	1
HTTP 메시지	1
JSON 포맷팅	2
ONTAP Select Deploy API에 액세스하는 방법	2
유ти리티 네이티브 사용자 인터페이스 배포	2
ONTAP Select Deploy 온라인 문서 페이지	2
맞춤형 프로그램	2
ONTAP Select Deploy API 기본 운영 특성	2
하이퍼바이저 호스트 대 ONTAP Select 노드	2
객체 식별자	3
요청 식별자	3
동기 및 비동기 호출	3
장기 실행 작업의 완료 확인	3
보안	3
ONTAP Select 대한 요청 및 응답 API 트랜잭션	4
API 요청을 제어하는 입력 변수	4
API 응답 해석	6
ONTAP Select 위한 Job 객체를 사용한 비동기 처리	7
Job 객체를 사용하여 설명된 비동기 요청	7
API 요청과 연관된 Job 객체를 쿼리합니다.	7
비동기 요청을 발행하기 위한 일반 절차	7

개념

ONTAP Select 클러스터 배포 및 관리를 위한 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. 주요 프로토콜과 표준을 사용하여 ONTAP Select 클러스터를 배포하고 관리하기 위한 유연한 기반을 제공합니다.

건축과 고전적 제약

REST는 Roy Fielding이 그의 박사 학위 논문에서 공식적으로 언급했습니다. "[논문](#)" 2000년에 UC 어바인에서. 이 논문은 일련의 제약 조건을 통해 아키텍처 스타일을 정의하며, 이러한 제약 조건들이 모여 웹 기반 애플리케이션과 그 기반 프로토콜을 개선합니다. 제약 조건은 상태 비저장 통신 프로토콜을 사용하는 클라이언트/서버 아키텍처 기반의 RESTful 웹 서비스 애플리케이션을 구축합니다.

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 관리합니다. 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 작업 정의 리소스는 항상 유한한 수의 상태 중 하나에 속합니다. 상태와 상태 변경에 영향을 미치는 관련 작업은 명확하게 정의되어야 합니다.

클라이언트와 서버 간에는 일반적인 CRUD(생성, 읽기, 업데이트, 삭제) 모델에 따라 리소스에 액세스하고 상태를 변경하기 위한 메시지가 교환됩니다.

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 동작(예: GET 및 POST)은 리소스 및 해당 상태 관리 작업에 매핑됩니다.

HTTP는 상태를 저장하지 않습니다. 따라서 관련된 요청과 응답 집합을 하나의 트랜잭션으로 연결하려면 요청/응답 데이터 흐름과 함께 전달되는 HTTP 헤더에 추가 정보를 포함해야 합니다.

JSON 포맷팅

클라이언트와 서버 간에 정보를 구조화하고 전송하는 방법은 여러 가지가 있지만, 가장 널리 사용되는 방식(그리고 Deploy REST API와 함께 사용되는 방식)은 JavaScript Object Notation(JSON)입니다. JSON은 간단한 데이터 구조를 일반 텍스트로 표현하는 업계 표준이며, 리소스를 설명하는 상태 정보를 전송하는 데 사용됩니다.

ONTAP Select Deploy API에 액세스하는 방법

REST 웹 서비스의 본질적인 유연성 덕분에 ONTAP Select Deploy API에 여러 가지 방법으로 액세스할 수 있습니다.



ONTAP Select Deploy에 포함된 REST API에는 버전 번호가 지정되어 있습니다. API 버전 번호는 Deploy 릴리스 번호와 무관합니다. ONTAP Select 9.17.1 Deploy 관리 유ти리티에는 REST API 버전 3이 포함되어 있습니다.

유ти리티 네이티브 사용자 인터페이스 배포

API에 접근하는 주요 방법은 ONTAP Select Deploy 웹 사용자 인터페이스를 이용하는 것입니다. 브라우저가 API를 호출하고 사용자 인터페이스 디자인에 따라 데이터를 재구성합니다. Deploy 유ти리티 명령줄 인터페이스를 통해서도 API에 접근할 수 있습니다.

ONTAP Select Deploy 온라인 문서 페이지

ONTAP Select Deploy 온라인 설명서 페이지는 브라우저를 사용할 때 다른 접근 방식을 제공합니다. 개별 API 호출을 직접 실행하는 방법을 제공할 뿐만 아니라, 각 호출에 대한 입력 매개변수 및 기타 옵션을 포함한 API에 대한 자세한 설명도 제공합니다. API 호출은 여러 기능 영역 또는 범주로 구성됩니다.

맞춤형 프로그램

다양한 프로그래밍 언어와 도구를 사용하여 Deploy API에 액세스할 수 있습니다. Python, Java, cURL 등이 많이 사용됩니다. API를 사용하는 프로그램, 스크립트 또는 도구는 REST 웹 서비스 클라이언트 역할을 합니다. 프로그래밍 언어를 사용하면 API를 더 잘 이해하고 ONTAP Select 배포를 자동화할 수 있습니다.

ONTAP Select Deploy API 기본 운영 특성

REST는 공통된 기술과 모범 사례를 제공하지만, 각 API의 세부 사항은 설계 방식에 따라 달라질 수 있습니다. ONTAP Select Deploy API를 사용하기 전에 해당 API의 세부 사항과 운영 특성을 숙지해야 합니다.

하이퍼바이저 호스트 대 ONTAP Select 노드

_하이퍼바이저 호스트_는 ONTAP Select 가상 머신을 호스팅하는 핵심 하드웨어 플랫폼입니다. ONTAP Select 가상 머신이 하이퍼바이저 호스트에 배포되고 활성화되면 해당 가상 머신은 _ONTAP Select 노드_로 간주됩니다. Deploy REST API 버전 3에서는 호스트 객체와 노드 객체가 분리되어 있습니다. 이를 통해 하나 이상의 ONTAP Select 노드가 동일한 하이퍼바이저 호스트에서 실행될 수 있는 일대다 관계가 가능합니다.

객체 식별자

각 리소스 인스턴스 또는 객체는 생성될 때 고유 식별자가 할당됩니다. 이러한 식별자는 ONTAP Select Deploy의 특정 인스턴스 내에서 전역적으로 고유합니다. 새로운 객체 인스턴스를 생성하는 API 호출을 발행한 후 연관된 ID 값이 호출자에게 반환됩니다. `Location` HTTP 응답의 헤더. 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.

요청 식별자

모든 성공적인 API 요청에는 고유 식별자가 할당됩니다. 식별자는 다음에서 반환됩니다. `request-id` 연관된 HTTP 응답의 헤더. 요청 식별자를 사용하여 단일 특정 API 요청-응답 트랜잭션의 활동을 집합적으로 참조할 수 있습니다. 예를 들어, 요청 ID를 기반으로 트랜잭션의 모든 이벤트 메시지를 검색할 수 있습니다.

동기 및 비동기 호출

서버가 클라이언트로부터 받은 HTTP 요청을 처리하는 주요 방법은 두 가지가 있습니다.

- 동기식 서버는 요청을 즉시 수행하고 200, 201 또는 204 상태 코드로 응답합니다.
- 비동기: 서버가 요청을 수락하고 상태 코드 202로 응답합니다. 이는 서버가 클라이언트 요청을 수락하고 요청을 완료하기 위한 백그라운드 작업을 시작했음을 나타냅니다. 최종 성공 또는 실패 여부는 즉시 알 수 없으며 추가 API 호출을 통해 결정해야 합니다.

장기 실행 작업의 완료 확인

일반적으로 완료하는 데 시간이 오래 걸릴 수 있는 모든 작업은 서버에서 백그라운드 작업을 사용하여 비동기적으로 처리됩니다. Deploy REST API를 사용하면 모든 백그라운드 작업은 작업을 추적하고 현재 상태와 같은 정보를 제공하는 Job 객체에 의해 고정됩니다. 백그라운드 작업이 생성된 후 고유 식별자를 포함한 Job 객체가 HTTP 응답으로 반환됩니다.

Job 객체를 직접 쿼리하여 관련 API 호출의 성공 또는 실패 여부를 확인할 수 있습니다. 자세한 내용은 `_Job` 객체를 사용한 비동기 처리_를 참조하세요.

Job 객체를 사용하는 것 외에도 요청의 성공 또는 실패를 판별하는 데 사용할 수 있는 다른 방법이 있습니다.

- 이벤트 메시지: 원래 응답과 함께 반환된 요청 ID를 사용하여 특정 API 호출과 관련된 모든 이벤트 메시지를 검색할 수 있습니다. 이벤트 메시지에는 일반적으로 성공 또는 실패 여부가 표시되며, 오류 조건을 디버깅할 때도 유용할 수 있습니다.
- 리소스 상태 또는 상태 여러 리소스는 요청의 성공 또는 실패를 간접적으로 확인하기 위해 쿼리할 수 있는 상태 또는 상태 값을 유지합니다.

보안

Deploy API는 다음과 같은 보안 기술을 사용합니다.

- 전송 계층 보안: 배포 서버와 클라이언트 간에 네트워크를 통해 전송되는 모든 트래픽은 TLS를 통해 암호화됩니다. 암호화되지 않은 채널을 통한 HTTP 프로토콜 사용은 지원되지 않습니다. TLS 버전 1.2가 지원됩니다.

- HTTP 인증: 기본 인증은 모든 API 트랜잭션에 사용됩니다. 사용자 이름과 비밀번호를 base64 문자열로 포함하는 HTTP 헤더가 모든 요청에 추가됩니다.

ONTAP Select 대한 요청 및 응답 API 트랜잭션

모든 Deploy API 호출은 Deploy 가상 머신에 대한 HTTP 요청으로 수행되며, 이 요청은 클라이언트에 연관된 응답을 생성합니다. 이 요청/응답 쌍은 API 트랜잭션으로 간주됩니다. Deploy API를 사용하기 전에 요청을 제어하는 데 사용할 수 있는 입력 변수와 응답 출력 내용을 숙지해야 합니다.

API 요청을 제어하는 입력 변수

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

요청 헤더

HTTP 요청에는 다음을 포함한 여러 헤더를 포함해야 합니다.

- content-type 요청 본문에 JSON이 포함되어 있는 경우 이 헤더를 application/json으로 설정해야 합니다.
- 응답 본문에 JSON이 포함되는 경우 이 헤더를 application/json으로 설정해야 합니다.
- 인증 기본 인증은 base64 문자열로 인코딩된 사용자 이름과 비밀번호로 설정해야 합니다.

요청 본문

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

- 입력 변수(예: 새 클러스터의 이름)가 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 값 범위에 대한 객체 집합을 반환하는 다른 연산자도 있습니다. ONTAP Select 아래에 표시된 필터링 연산자를 지원합니다.

연산자	설명
=	동일하다
<	미만
>	보다 크다
≤	이하
≥	이상 또는 같음

연산자	설명
	또는
!	같지 않다
*	탐욕스러운 와일드카드

null 키워드나 부정(!null)을 쿼리의 일부로 사용하면 특정 필드가 설정되었는지 여부에 따라 객체 집합을 반환할 수도 있습니다.

객체 필드 선택

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

- 저렴한 필드 지정 `fields=*` 로컬 서버 메모리에 유지 관리되거나 액세스하는 데 거의 처리가 필요하지 않은 개체 필드를 검색합니다.
- 비싼 필드 지정 `fields=**` 추가적인 서버 처리가 필요한 항목을 포함하여 모든 개체 필드를 검색합니다.
- 사용자 정의 필드 선택 사용 `fields=FIELDNAME` 원하는 정확한 필드를 지정하세요. 여러 필드를 요청할 경우, 공백 없이 쉼표로 값을 구분해야 합니다.



의 필요할 때만 비용이 많이 들거나 저렴한 필드 집합을 검색해야 합니다. 비용이 많이 들거나 저렴한 필드 분류는 NetApp에서 내부 성능 분석을 기반으로 결정합니다. 분류는 언제든지 변경될 수 있습니다.

출력 세트의 객체 정렬

리소스 컬렉션의 레코드는 객체에 정의된 기본 순서대로 반환됩니다. 다음과 같이 필드 이름과 정렬 방향을 사용하여 `order_by` 쿼리 매개변수를 사용하여 순서를 변경할 수 있습니다.

`order_by=<field name> asc|desc`

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

`order_by=type desc, id asc`

여러 개의 매개변수를 포함하는 경우 필드를 쉼표로 구분해야 합니다.

쪽수 매기기

동일한 유형의 객체 컬렉션에 액세스하기 위해 GET 방식으로 API 호출을 실행하면 기본적으로 일치하는 모든 객체가 반환됩니다. 필요한 경우, 요청과 함께 `max_records` 쿼리 매개변수를 사용하여 반환되는 레코드 수를 제한할 수 있습니다. 예를 들어:

`max_records=20`

필요한 경우 이 매개변수를 다른 쿼리 매개변수와 결합하여 결과 집합을 좁힐 수 있습니다. 예를 들어, 다음은 지정된 시간 이후에 생성된 최대 10개의 시스템 이벤트를 반환합니다.

`time⇒ 2019-04-04T15:41:29.140265Z&max_records=10`

이벤트(또는 모든 객체 유형)를 페이지별로 탐색하기 위해 여러 요청을 보낼 수 있습니다. 이후의 각 API 호출은 마지막 결과 집합의 최신 이벤트를 기반으로 새로운 시간 값을 사용해야 합니다.

API 응답 해석

각 API 요청은 클라이언트에게 응답을 생성합니다. 응답을 검토하여 성공 여부를 확인하고 필요에 따라 추가 데이터를 검색할 수 있습니다.

HTTP 상태 코드

Deploy REST API에서 사용하는 HTTP 상태 코드는 아래와 같습니다.

암호	의미	설명
200	OK	새로운 객체를 생성하지 않는 호출의 경우 성공을 나타냅니다.
201	생성됨	객체가 성공적으로 생성되었습니다. 위치 응답 헤더에 객체의 고유 식별자가 포함되어 있습니다.
202	수락됨	요청을 수행하기 위해 장기 실행 백그라운드 작업이 시작되었지만 작업이 아직 완료되지 않았습니다.
400	잘못된 요청	요청 입력이 인식되지 않거나 부적절합니다.
403	금지됨	인증 오류로 인해 접근이 거부되었습니다.
404	찾을 수 없음	요청에 언급된 리소스가 존재하지 않습니다.
405	허용되지 않는 메서드입니다	요청의 HTTP 동사는 해당 리소스에서 지원되지 않습니다.
409	갈등	개체가 이미 존재하기 때문에 개체를 생성하려는 시도가 실패했습니다.
500	내부 오류	서버에서 일반적인 내부 오류가 발생했습니다.
501	구현되지 않음	URI는 알려져 있지만 요청을 수행할 수 없습니다.

응답 헤더

Deploy 서버에서 생성된 HTTP 응답에는 다음을 포함한 여러 헤더가 포함됩니다.

- 요청 ID 모든 성공적인 API 요청에는 고유한 요청 식별자가 지정됩니다.
- 위치 객체가 생성되면 위치 헤더에는 고유한 객체 식별자를 포함한 새 객체의 전체 URL이 포함됩니다.

대응 기관

API 요청과 관련된 응답 내용은 객체, 처리 유형, 요청의 성공 또는 실패 여부에 따라 달라집니다. 응답 본문은 JSON 형식으로 렌더링됩니다.

- 단일 객체: 요청에 따라 필드 집합과 함께 단일 객체를 반환할 수 있습니다. 예를 들어, GET을 사용하면 고유 식별자를 사용하여 클러스터의 선택된 속성을 검색할 수 있습니다.
- 여러 객체 리소스 컬렉션에서 여러 객체를 반환할 수 있습니다. 모든 경우에 일관된 형식이 사용됩니다. num_records 레코드의 개수와 객체 인스턴스의 배열을 포함하는 레코드를 나타냅니다. 예를 들어, 특정 클러스터에 정의된 모든 노드를 검색할 수 있습니다.
- Job 객체 API 호출이 비동기적으로 처리되면 백그라운드 작업을 고정하는 Job 객체가 반환됩니다. 예를 들어, 클러스터 배포에 사용되는 POST 요청은 비동기적으로 처리되어 Job 객체를 반환합니다.
- 오류 객체: 오류가 발생하면 항상 Error 객체가 반환됩니다. 예를 들어, 이미 존재하는 이름으로 클러스터를 생성하려고 하면 오류가 발생합니다.

- 비어 있음 특정 경우에는 데이터가 반환되지 않고 응답 본문이 비어 있습니다. 예를 들어, DELETE 키를 사용하여 기존 호스트를 삭제한 후 응답 본문이 비어 있는 경우가 있습니다.

ONTAP Select 위한 Job 객체를 사용한 비동기 처리

일부 Deploy API 호출, 특히 리소스를 생성하거나 수정하는 호출은 다른 호출보다 완료하는 데 시간이 더 오래 걸릴 수 있습니다. ONTAP Select Deploy는 이러한 장기 실행 요청을 비동기적으로 처리합니다.

Job 객체를 사용하여 설명된 비동기 요청

비동기적으로 실행되는 API 호출 후 HTTP 응답 코드 202는 요청이 성공적으로 검증되고 수락되었지만 아직 완료되지 않았음을 나타냅니다. 요청은 클라이언트에 대한 초기 HTTP 응답 이후에도 계속 실행되는 백그라운드 작업으로 처리됩니다. 응답에는 요청을 앵커링하는 Job 객체와 고유 식별자가 포함됩니다.



어떤 API 호출이 비동기적으로 작동하는지 확인하려면 ONTAP Select Deploy 온라인 설명서 페이지를 참조하세요.

API 요청과 연관된 Job 객체를 쿼리합니다.

HTTP 응답으로 반환되는 Job 객체에는 여러 속성이 포함되어 있습니다. state 속성을 쿼리하여 요청이 성공적으로 완료되었는지 확인할 수 있습니다. Job 객체는 다음 상태 중 하나일 수 있습니다.

- 대기 중
- 달리기
- 성공
- 실패

작업의 최종 상태(성공 또는 실패)를 감지하기 위해 Job 객체를 폴링할 때 사용할 수 있는 두 가지 기술이 있습니다.

- 표준 폴링 요청 현재 작업 상태가 즉시 반환됩니다.
- 롱 폴링 요청 작업 상태는 다음 중 하나가 발생할 때만 반환됩니다.
 - 여론조사 요청에 제공된 날짜-시간 값보다 최근에 상태가 변경되었습니다.
 - 시간 초과 값이 만료되었습니다(1~120초)

표준 폴링과 롱 폴링은 동일한 API 호출을 사용하여 Job 객체를 쿼리합니다. 그러나 긴 폴링 요청에는 두 개의 쿼리 매개변수가 포함됩니다. `poll_timeout` 그리고 `last_modified`.



배포 가상 머신의 작업 부하를 줄이려면 항상 롱 폴링을 사용해야 합니다.

비동기 요청을 발행하기 위한 일반 절차

다음의 고급 절차를 사용하여 비동기 API 호출을 완료할 수 있습니다.

1. 비동기 API 호출을 실행합니다.

2. 요청이 성공적으로 수락되었음을 나타내는 HTTP 응답 202를 받습니다.
3. 응답 본문에서 Job 객체의 식별자를 추출합니다.
4. 루프 내에서 각 사이클마다 다음을 수행합니다.
 - a. 롱폴 요청으로 Job의 현재 상태를 가져옵니다.
 - b. 작업이 종료 상태가 아닌 상태(대기 중, 실행 중)인 경우 루프를 다시 수행합니다.
5. 작업이 최종 상태(성공, 실패)에 도달하면 중지됩니다.

저작권 정보

Copyright © 2026 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.