개념
ONTAP Select

NetApp
January 31, 2025

This PDF was generated from https://docs.netapp.com/ko-kr/ontap-select-9111/concept_api_rest.html on January 31, 2025. Always check docs.netapp.com for the latest.

목차

개념 1
REST 웹 서비스 기반 1
배포 API에 액세스하는 방법 2
API 버전 관리 배포 2
기본 작동 특성 3
요청 및 응답 API 트랜잭션 4
작업 개체를 사용한 비동기 처리 7

개념

REST 웹 서비스 기반

REST(Representational State Transfer)는 분산된 웹 애플리케이션을 만드는 스타일입니다. 웹 서비스 API 설계에 적용할 경우 서버 기반 리소스를 노출하고 상태를 관리하기 위한 일련의 기술과 Best Practice를 수립합니다. 이 솔루션은 메인스트림 프로토콜과 표준을 사용하여 ONTAP Select 클러스터를 배포하고 관리하기 위한 유연한 기반을 제공합니다.

기존의 제약과 아키텍처를 동시에 사용합니다

휴식은 로이 필딩의 박사 학위를 받아 공식적으로 표현되었습니다 "학위 논문" 2000년 UC Irvine에서. 이 아키텍처는 일련의 제약 조건을 통해 아키텍처 스타일을 정의하며, 이러한 제약은 웹 기반 응용 프로그램과 기본 프로토콜을 모두 개선합니다. 이 제약 조건은 상태 비저장 통신 프로토콜을 사용하는 클라이언트/서버 아키텍처를 기반으로 RESTful 웹 서비스 애플리케이션을 설정합니다.

리소스 및 상태 표시

리소스는 웹 기반 시스템의 기본 구성 요소입니다. REST 웹 서비스 응용 프로그램을 만들 때 초기 설계 작업은 다음과 같습니다.

- 시스템 또는 서버 기반 리소스 식별
  모든 시스템은 리소스를 사용하고 유지합니다. 리소스는 파일, 비즈니스 트랜잭션, 프로세스 또는 관리 엔티티가 될 수 있습니다. REST 웹 서비스를 기반으로 애플리케이션을 설계하는 첫 번째 작업 중 하나는 리소스를 식별하는 것입니다.
- 자원 상태 및 연관된 상태 작업의 정의
  리소스는 항상 한정된 수의 상태 중 하나에 있습니다. 상태 변경에 영향을 주는 데 사용되는 상태 및 관련 작업을 명확하게 정의해야 합니다.

일반 CRUD(생성, 읽기, 업데이트 및 삭제) 모델에 따라 리소스 상태를 액세스하고 변경하기 위해 클라이언트와 서버 간에 메시지가 교환됩니다.

URI 끝점

모든 REST 리소스는 잘 정의된 주소 지정 체계를 사용하여 정의되고 사용 가능해야 합니다. 리소스가 있고 식별된 끝점은 URI(Uniform Resource Identifier)를 사용합니다. URI는 네트워크의 각 리소스에 대해 고유한 이름을 만들기 위한 일반 프레임워크를 제공합니다. URL(Uniform Resource Locator)은 리소스를 식별하고 액세스하기 위해 웹 서비스와 함께 사용되는 URI 유형입니다. 일반적으로 리소스는 파일 디렉터리와 비슷한 계층적 구조로 표시됩니다.

HTTP 메시지

HTTP(Hypertext Transfer Protocol)는 웹 서비스 클라이언트 및 서버가 리소스에 대한 요청 및 응답 메시지를 교환하기 위해 사용하는 프로토콜입니다. 웹 서비스 응용 프로그램을 디자인하는 과정에서 HTTP 동사(예: GET 및 POST)가 리소스와 해당 상태 관리 작업에 매핑됩니다.

HTTP는 상태 비저장 입니다. 따라서 하나의 트랜잭션에서 관련 요청 및 응답 집합을 연결하려면 요청/응답 데이터 플로우와 함께 전달된 HTTP 헤더에 추가 정보가 포함되어야 합니다.

JSON 형식

여러 가지 방법으로 클라이언트와 서버 간에 정보를 구성 및 전송할 수 있지만 가장 널리 사용되는 옵션(및 배포 REST API와 함께 사용되는 옵션)은 JSON(JavaScript Object Notation)입니다. JSON은 단순 데이터 구조를 일반 텍스트로 나타내는 업계 표준이며 리소스를 설명하는 상태 정보를 전송하는 데 사용됩니다.

배포 API에 액세스하는 방법

REST 웹 서비스의 고유한 유연성 때문에 ONTAP Select Deploy API는 여러 가지 방법으로 액세스할 수 있습니다.

유틸리티 네이티브 사용자 인터페이스를 구축합니다

API에 액세스하는 주요 방법은 ONTAP Select 배포 웹 사용자 인터페이스를 사용하는 것입니다. 브라우저는 API를 호출하고 사용자 인터페이스 설계에 따라 데이터를 다시 포맷합니다. 또한 deploy utility 명령줄 인터페이스를 통해 API에 액세스할 수 있습니다.

ONTAP Select 온라인 설명서 페이지 배포

ONTAP Select 배포 온라인 설명서 페이지에서는 브라우저를 사용할 때 다른 액세스 지점을 제공합니다. 개별 API 호출을 직접 실행하는 방법을 제공할 뿐만 아니라 각 호출에 대한 입력 매개 변수와 기타 옵션을 포함하여 API에 대한 자세한 설명도 페이지에 포함되어 있습니다. API 호출은 여러 기능 영역 또는 범주로 구성됩니다.

맞춤형 프로그램

여러 가지 프로그래밍 언어 및 도구를 사용하여 배포 API에 액세스할 수 있습니다. Python, Java 및 Curl을 선택할 수 있습니다. API를 사용하는 프로그램, 스크립트 또는 도구는 REST 웹 서비스 클라이언트로 작동합니다. 프로그래밍 언어를 사용하면 API를 더 잘 이해할 수 있으며 ONTAP Select 배포를 자동화할 수 있습니다.

API 버전 관리 배포

ONTAP Select 배포에 포함된 REST API에는 버전 번호가 할당됩니다. API 버전 번호는 배포 릴리스 번호와 무관합니다. 배포 릴리스에 포함된 API 버전과 이 버전이 API 사용에 어떤 영향을 줄 수 있는지 알고 있어야 합니다.

배포 관리 유틸리티의 현재 릴리즈에는 REST API 버전 3이 포함되어 있습니다. 배포 유틸리티의 이전 릴리스에는 다음과 같은 API 버전이 포함되어 있습니다.

2.8 이상 버전 배포

ONTAP Select Deploy 2.8 및 이후의 모든 릴리즈에는 REST API 버전 3이 포함됩니다.

2.7.2 이전 버전 배포

ONTAP Select 배포 2.7.2 및 이전 모든 릴리즈에는 REST API 버전 2가 포함되어 있습니다.

REST API 버전 2와 3은 호환되지 않습니다. API 버전 2가 포함된 이전 릴리즈에서 2.8 이상을 배포하도록 업그레이드하는 경우 명령줄 인터페이스를 사용하여 API에 직접 액세스하는 기존 코드와 스크립트를 모두 업데이트해야 합니다.

기본 작동 특성

REST는 일반적인 기술과 모범 사례를 설정하지만 각 API의 세부 사항은 설계 선택에 따라 달라질 수 있습니다. API를 사용하기 전에 ONTAP Select 배포 API의 세부 정보와 운영 특성을 알고 있어야 합니다.

하이퍼바이저 호스트와 **ONTAP Select** 노드 비교

h하이퍼바이저 host_는 ONTAP Select 가상 머신을 호스팅하는 핵심 하드웨어 플랫폼입니다. ONTAP Select 가상 머신을 하이퍼바이저 호스트에 구축하고 활성화하면 가상 머신이 _ONTAP Select node_로 간주됩니다. Deploy REST API의 버전 3에서는 호스트 및 노드 객체가 구분되고 구분됩니다. 따라서 일대다 관계가 허용되며, 이 경우 하나 이상의 ONTAP Select 노드가 동일한 하이퍼바이저 호스트에서 실행될 수 있습니다.

오브젝트 식별자

각 리소스 인스턴스 또는 개체는 만들 때 고유한 식별자가 할당됩니다. 이러한 식별자는 ONTAP Select 배포의 특정 인스턴스 내에서 전역적으로 고유합니다. 새 개체 인스턴스를 만드는 API 호출을 실행한 후 연결된 ID 값이 의 호출자에게 반환됩니다 `location` HTTP 응답의 헤더입니다. 식별자를 추출하여 리소스 인스턴스를 참조할 때 후속 호출에 사용할 수 있습니다.

> 개체 식별자의 내용 및 내부 구조는 언제든지 변경할 수 있습니다. 관련 객체를 참조할 때는 필요한 경우 해당 API 호출에서만 식별자를 사용해야 합니다.

요청 식별자

성공한 모든 API 요청에는 고유 식별자가 할당됩니다. 식별자는 에 반환됩니다 `request-id` 연결된 HTTP 응답의 헤더입니다. 요청 식별자를 사용하여 단일 특정 API 요청 응답 트랜잭션의 활동을 종합적으로 참조할 수 있습니다. 예를 들어 요청 ID를 기반으로 트랜잭션에 대한 모든 이벤트 메시지를 검색할 수 있습니다

동기 및 비동기 호출

서버에서 클라이언트로부터 받은 HTTP 요청을 수행하는 방법에는 두 가지가 있습니다.

- 동기식이다
  서버는 요청을 즉시 수행하고 상태 코드 200, 201 또는 204로 응답합니다.
- 비동기식
  서버는 요청을 수락하고 상태 코드 202로 응답합니다. 서버가 클라이언트 요청을 수락하고 요청을 완료하기 위해 백그라운드 작업을 시작했음을 나타냅니다. 최종 성공 또는 실패는 즉시 사용할 수 없으며 추가 API 호출을 통해 결정되어야 합니다.

장기 실행 작업의 완료를 확인합니다

일반적으로 완료하는 데 시간이 오래 걸릴 수 있는 작업은 을 사용하여 비동기적으로 처리됩니다
서버의 백그라운드 작업입니다. REST API 배포 를 사용하면 모든 백그라운드 작업이 에 의해 고정됩니다
작업을 추적하고 현재 상태와 같은 정보를 제공하는 작업 개체입니다. 작업 객체,
고유 식별자를 포함하여 백그라운드 작업이 생성된 후 HTTP 응답에 반환됩니다.

Job 객체를 직접 쿼리하여 연결된 API 호출의 성공 또는 실패를 확인할 수 있습니다.
자세한 내용은 작업 객체를 사용한 _ 비동기 처리를 참조하십시오.

Job 객체를 사용하는 것 외에도 의 성공 또는 실패를 확인할 수 있는 다른 방법이 있습니다
다음을 포함한 요청:

- 이벤트 메시지
  원래 응답과 함께 반환된 요청 ID를 사용하여 특정 API 호출과 관련된 모든 이벤트 메시지를 검색할 수 있습니다. 이벤트 메시지는 일반적으로 성공 또는 실패를 나타내는 것으로, 오류 조건을 디버깅할 때 유용할 수 있습니다.
- 리소스 상태 또는 상태입니다
  일부 리소스는 요청의 성공 또는 실패를 간접적으로 결정하기 위해 쿼리할 수 있는 상태 또는 상태 값을 유지 관리합니다.

보안

배포 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. base64 문자열에 사용자 이름과 암호를 포함하는 HTTP 헤더가 모든 요청에 추가됩니다.

요청 및 응답 API 트랜잭션

모든 배포 API 호출은 클라이언트에 대한 관련 응답을 생성하는 배포 가상 시스템에 대한 HTTP 요청으로 수행됩니다. 이 요청/응답 쌍은 API 트랜잭션으로 간주됩니다. Deploy API를 사용하기 전에 요청을 제어하는 데 사용할 수 있는 입력 변수와 응답 출력의 내용을 숙지해야 합니다.

API 요청을 제어하는 입력 변수입니다

HTTP 요청에 설정된 매개 변수를 통해 API 호출 처리 방법을 제어할 수 있습니다.

요청 헤더

HTTP 요청에 다음과 같은 여러 헤더를 포함해야 합니다.

- 컨텐츠 유형
  요청 본문에 JSON이 포함된 경우 이 헤더는 application/json으로 설정해야 합니다.
- 수락
  응답 본문에 JSON이 포함될 경우 이 헤더는 application/json으로 설정해야 합니다.
- 권한 부여
  기본 인증은 base64 문자열로 인코딩된 사용자 이름과 암호로 설정해야 합니다.

요청 본문

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

- 입력 변수가 있는 JSON 개체(예: 새 클러스터의 이름)
- 비어 있습니다

개체를 필터링합니다

Get을 사용하는 API 호출을 실행할 때 모든 특성에 따라 반환된 객체를 제한하거나 필터링할 수 있습니다. 예를 들어, 다음과 같이 정확하게 일치하는 값을 지정할 수 있습니다.

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

정확한 일치 항목 외에 값 범위에 있는 개체 집합을 반환할 수 있는 다른 연산자도 있습니다. ONTAP Select는 아래 표시된 필터링 연산자를 지원합니다.

운영자	설명
=	같음
를 누릅니다	보다 작음
를 누릅니다	보다 큼
lt;=.(&L	보다 작거나 같음
GT;=.(&T	보다 크거나 같음
	또는
!	같지 않음
*	greedy 와일드카드

또한 쿼리의 일부로 Null 키워드 또는 해당 부정(!null)을 사용하여 특정 필드가 설정되었는지 여부를 기준으로 개체 집합을 반환할 수도 있습니다.

개체 필드 선택

기본적으로 Get 을 사용하여 API 호출을 실행하면 개체나 개체를 고유하게 식별하는 특성만 반환됩니다. 이 최소 필드 집합은 각 개체의 키 역할을 하며 개체 유형에 따라 달라집니다. 다음과 같은 방법으로 필드 쿼리 매개 변수를 사용하여 추가 개체 속성을 선택할 수 있습니다.

- 저렴한 필드
  를 지정합니다 `fields=*` 로컬 서버 메모리에 유지되거나 액세스하는 데 약간의 처리가 필요한 개체 필드를 검색합니다.
- 비싼 필드
  를 지정합니다 `fields=**` 액세스를 위해 추가 서버 처리가 필요한 개체를 포함하여 모든 개체 필드를 검색합니다.
- 사용자 정의 필드 선택
  사용 `fields=FIELDNAME` 를 눌러 원하는 정확한 필드를 지정합니다. 여러 필드를 요청할 때는 공백 없이 쉼표를 사용하여 값을 구분해야 합니다.

가장 좋은 방법은 항상 원하는 특정 필드를 식별하는 것입니다. 필요한 경우 저렴한 필드나 값비싼 필드 집합만 검색해야 합니다. 저렴하고 값비싼 분류는 내부 성능 분석을 기반으로 NetApp에서 결정합니다. 지정된 필드의 분류는 언제든지 변경할 수 있습니다.

출력 집합에서 개체를 정렬합니다

리소스 컬렉션의 레코드는 개체에서 정의한 기본 순서로 반환됩니다. 다음과 같이 필드 이름 및 정렬 방향이 있는 ORDER_BY 쿼리 매개 변수를 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 매개 변수를 포함할 때는 필드를 쉼표로 구분해야 합니다.

페이지 매김

Get 을 사용하여 API 호출을 실행하면 동일한 형식의 개체 컬렉션에 액세스할 때 일치하는 모든 개체가 기본적으로 반환됩니다. 필요한 경우 요청과 함께 max_records 쿼리 매개 변수를 사용하여 반환되는 레코드 수를 제한할 수 있습니다. 예를 들면 다음과 같습니다.
```
max_records=20
```

필요한 경우 이 매개 변수를 다른 쿼리 매개 변수와 결합하여 결과 집합의 범위를 좁힐 수 있습니다. 예를 들어, 다음과 같이 지정된 시간 이후에 생성된 최대 10개의 시스템 이벤트를 반환합니다.
```
time⇒ 2019-04-04T15:41:29.140265Z&max_records=10
```

이벤트(또는 모든 개체 유형)를 통해 페이지를 넘도록 여러 요청을 실행할 수 있습니다. 이후의 각 API 호출은 마지막 결과 집합의 최신 이벤트를 기반으로 새 시간 값을 사용해야 합니다.

API 응답을 해석합니다

각 API 요청은 클라이언트에 대한 응답을 다시 생성합니다. 응답을 검토하여 확인할 수 있습니다
성공 여부 및 필요에 따라 추가 데이터 검색

HTTP 상태 코드입니다

Deploy REST API에 사용되는 HTTP 상태 코드는 아래와 같다.

코드	의미	설명
200	좋습니다	새 개체를 만들지 않는 호출에 대한 성공 여부를 나타냅니다.
201	작성됨	객체가 성공적으로 생성되었습니다. 위치 응답 헤더에는 객체의 고유 식별자가 포함됩니다.
202	수락됨	요청을 수행하기 위해 오래 실행되는 백그라운드 작업이 시작되었지만 작업이 아직 완료되지 않았습니다.
400	잘못된 요청입니다	요청 입력이 인식되지 않거나 부적절합니다.
403	금지됨	인증 오류로 인해 액세스가 거부되었습니다.
404	찾을 수 없습니다	요청에서 참조되는 리소스가 없습니다.
405	메서드가 허용되지 않습니다	요청의 HTTP 동사는 리소스에 대해 지원되지 않습니다.
409	충돌	개체가 이미 있으므로 개체를 만들지 못했습니다.
500입니다	내부 오류입니다	서버에서 일반적인 내부 오류가 발생했습니다.
501	구현되지 않았습니다	URI를 알고 있지만 요청을 수행할 수 없습니다.

응답 헤더

배포 서버에서 생성된 HTTP 응답에는 다음과 같은 여러 헤더가 포함됩니다.

- 요청 ID입니다
  성공한 모든 API 요청에는 고유한 요청 식별자가 할당됩니다.
- 위치
  개체가 만들어지면 위치 헤더에 고유한 개체 식별자를 포함하여 새 개체에 대한 전체 URL이 포함됩니다.

응답 바디

API 요청과 관련된 응답 내용은 객체, 처리 유형 및 요청의 성공 또는 실패에 따라 달라집니다. 응답 본문은 JSON으로 렌더링됩니다.

- 단일 개체
  요청에 따라 필드 집합과 함께 단일 개체를 반환할 수 있습니다. 예를 들어, 가져오기를 사용하여 고유 식별자를 사용하여 클러스터의 선택된 속성을 검색할 수 있습니다.
- 여러 개의 개체
  리소스 컬렉션의 여러 개체를 반환할 수 있습니다. 어떤 경우든 에는 일관된 형식이 사용됩니다 `num_records` 개체 인스턴스의 배열을 포함하는 레코드 및 레코드 수를 나타냅니다. 예를 들어, 특정 클러스터에 정의된 모든 노드를 검색할 수 있습니다.
- 작업 객체
  API 호출이 비동기적으로 처리되는 경우 백그라운드 작업을 고정하는 Job 개체가 반환됩니다. 예를 들어, 클러스터를 배포하는 데 사용되는 POST 요청은 비동기적으로 처리되고 작업 개체를 반환합니다.
- 오류 개체
  오류가 발생하면 항상 Error 개체가 반환됩니다. 예를 들어, 이름이 이미 존재하는 클러스터를 생성하려고 하면 오류가 발생합니다.
- 비어 있습니다
  경우에 따라 데이터가 반환되지 않고 응답 본문이 비어 있습니다. 예를 들어, 삭제 기능을 사용하여 기존 호스트를 삭제한 후 응답 본문이 비어 있습니다.

작업 개체를 사용한 비동기 처리

일부 배포 API 호출, 특히 리소스를 생성하거나 수정하는 호출은 다른 호출보다 완료하는 데 시간이 오래 걸릴 수 있습니다. ONTAP Select 배포는 오래 실행되는 이러한 요청을 비동기적으로 처리합니다.

작업 개체를 사용하여 설명된 비동기 요청

비동기적으로 실행되는 API 호출을 수행한 후 HTTP 응답 코드 202는 요청이 성공적으로 유효성 확인 및 승인되었지만 아직 완료되지 않았음을 나타냅니다. 요청은 클라이언트에 대한 초기 HTTP 응답 후 계속 실행되는 백그라운드 작업으로 처리됩니다. 응답에는 고유한 식별자를 포함하여 요청을 고정하는 작업 객체가 포함됩니다.

ONTAP Select 배포 온라인 설명서 페이지에서 비동기적으로 작동하는 API 호출을 확인해야 합니다.

API 요청과 연결된 작업 객체를 쿼리합니다

HTTP 응답에서 반환된 작업 개체에는 여러 속성이 포함되어 있습니다. 상태 속성을 쿼리하여 요청이 성공적으로 완료되었는지 확인할 수 있습니다. 작업 오브젝트는 다음 상태 중 하나일 수 있습니다.

- 대기열에 있습니다
- 실행 중입니다
- 성공
- 실패

작업 개체를 폴링하여 작업의 터미널 상태를 감지할 때 성공 또는 실패 등 두 가지 방법을 사용할 수 있습니다.

- 표준 폴링 요청입니다
  현재 작업 상태가 즉시 반환됩니다
- 긴 폴링 요청
  작업 상태는 다음 중 하나가 발생할 경우에만 반환됩니다.
  - 상태가 폴링 요청에 제공된 날짜-시간 값보다 최근 변경되었습니다
  - 시간 초과 값이 만료되었습니다(1-120초).

표준 폴링과 긴 폴링은 동일한 API 호출을 사용하여 작업 객체를 쿼리합니다. 그러나 긴 폴링 요청에는 두 가지 쿼리 매개 변수가 포함됩니다. `poll_timeout` 및 `last_modified`.

항상 긴 폴링을 사용하여 배포 가상 머신의 워크로드를 줄여야 합니다.

비동기 요청을 실행하는 일반 절차입니다

다음 고급 절차를 사용하여 비동기 API 호출을 완료할 수 있습니다.

1. 비동기 API 호출을 실행합니다.
2. 요청을 성공적으로 수락했음을 나타내는 HTTP 응답 202 을 수신합니다.
3. 응답 본문에서 작업 객체의 식별자를 추출합니다.
4. 루프 내에서 각 사이클에서 다음을 수행합니다.
   a. 장기 폴링 요청으로 작업의 현재 상태를 가져옵니다
   b. 작업이 비터미널 상태(대기, 실행 중)인 경우 루프를 다시 수행합니다.
5. 작업이 터미널 상태(성공, 실패)에 도달하면 중지합니다.

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 그래픽, 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 없으며 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/TM에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.