



# **Concetti**

## ONTAP Select

NetApp  
January 29, 2026

# Sommario

Concetti . . . . .	1
Fondamenti di servizi Web REST per la distribuzione e la gestione di cluster ONTAP Select . . . . .	1
Architettura e vincoli classici . . . . .	1
Risorse e rappresentanza statale . . . . .	1
Endpoint URI . . . . .	1
Messaggi HTTP . . . . .	1
Formattazione JSON . . . . .	2
Come accedere all'API ONTAP Select Deploy . . . . .	2
Distribuisci l'interfaccia utente nativa dell'utilità . . . . .	2
Pagina della documentazione online ONTAP Select Deploy . . . . .	2
Programma personalizzato . . . . .	2
Controllo delle versioni dell'API ONTAP Select Deploy . . . . .	2
Caratteristiche operative di base dell'API ONTAP Select Deploy . . . . .	3
Host dell'hypervisor rispetto al nodo ONTAP Select . . . . .	3
Identificatori di oggetti . . . . .	3
Identificatori di richiesta . . . . .	3
Chiamate sincrone e asincrone . . . . .	3
Confermare il completamento di un lavoro di lunga durata . . . . .	4
Sicurezza . . . . .	4
Transazione API di richiesta e risposta per ONTAP Select . . . . .	4
Variabili di input che controllano una richiesta API . . . . .	4
Interpretare una risposta API . . . . .	6
Elaborazione asincrona tramite l'oggetto Job per ONTAP Select . . . . .	7
Richieste asincrone descritte utilizzando l'oggetto Job . . . . .	8
Interroga l'oggetto Job associato a una richiesta API . . . . .	8
Procedura generale per l'emissione di una richiesta asincrona . . . . .	8

# Concetti

## Fondamenti di servizi Web REST per la distribuzione e la gestione di cluster ONTAP Select

Il Representational State Transfer (REST) è uno stile per la creazione di applicazioni web distribuite. Applicato alla progettazione di un'API di servizi web, stabilisce un insieme di tecnologie e best practice per l'esposizione delle risorse basate su server e la gestione dei loro stati. Utilizza protocolli e standard tradizionali per fornire una base flessibile per l'implementazione e la gestione di cluster ONTAP Select .

### Architettura e vincoli classici

REST è stato formalmente articolato da Roy Fielding nel suo dottorato di ricerca "dissertazione" presso l'Università della California, Irvine, nel 2000. Definisce uno stile architettonico attraverso una serie di vincoli, che nel loro insieme migliorano le applicazioni web e i protocolli sottostanti. I vincoli stabiliscono un'applicazione di servizi web RESTful basata su un'architettura client/server che utilizza un protocollo di comunicazione stateless.

### Risorse e rappresentanza statale

Le risorse sono i componenti di base di un sistema basato sul web. Quando si crea un'applicazione di servizi web REST, le prime attività di progettazione includono:

- Identificazione delle risorse di sistema o basate su server Ogni sistema utilizza e gestisce risorse. Una risorsa può essere un file, una transazione aziendale, un processo o un'entità amministrativa. Uno dei primi compiti nella progettazione di un'applicazione basata su servizi web REST è l'identificazione delle risorse.
- Definizione degli stati delle risorse e delle operazioni di stato associate Le risorse si trovano sempre in uno di un numero finito di stati. Gli stati, così come le operazioni associate utilizzate per influenzare i cambiamenti di stato, devono essere chiaramente definiti.

I messaggi vengono scambiati tra il client e il server per accedere e modificare lo stato delle risorse secondo il modello CRUD (Crea, Leggi, Aggiorna ed Elimina) generico.

### Endpoint URI

Ogni risorsa REST deve essere definita e resa disponibile utilizzando uno schema di indirizzamento ben definito. Gli endpoint in cui le risorse sono localizzate e identificate utilizzano un Uniform Resource Identifier (URI). L'URI fornisce un framework generale per la creazione di un nome univoco per ciascuna risorsa nella rete. L'Uniform Resource Locator (URL) è un tipo di URI utilizzato con i servizi web per identificare e accedere alle risorse. Le risorse sono in genere esposte in una struttura gerarchica simile a una directory di file.

### Messaggi HTTP

Il protocollo HTTP (Hypertext Transfer Protocol) è il protocollo utilizzato dal client e dal server dei servizi web per scambiare messaggi di richiesta e risposta relativi alle risorse. Durante la progettazione di un'applicazione di servizi web, i verbi HTTP (come GET e POST) vengono mappati alle risorse e alle corrispondenti azioni di gestione dello stato.

HTTP è un protocollo stateless. Pertanto, per associare un insieme di richieste e risposte correlate in un'unica transazione, è necessario includere informazioni aggiuntive nelle intestazioni HTTP trasportate con i flussi di dati di richiesta/risposta.

## Formattazione JSON

Sebbene le informazioni possano essere strutturate e trasferite tra un client e un server in diversi modi, l'opzione più diffusa (e quella utilizzata con la Deploy REST API) è JavaScript Object Notation (JSON). JSON è uno standard industriale per la rappresentazione di strutture dati semplici in testo normale e viene utilizzato per trasferire informazioni sullo stato che descrivono le risorse.

## Come accedere all'API ONTAP Select Deploy

Grazie alla flessibilità intrinseca dei servizi Web REST, è possibile accedere all'API ONTAP Select Deploy in diversi modi.

### Distribuisci l'interfaccia utente nativa dell'utilità

Il modo principale per accedere all'API è tramite l'interfaccia utente web ONTAP Select Deploy. Il browser effettua chiamate all'API e riformatta i dati in base al design dell'interfaccia utente. È possibile accedere all'API anche tramite l'interfaccia a riga di comando dell'utilità Deploy.

### Pagina della documentazione online ONTAP Select Deploy

La pagina della documentazione online ONTAP Select Deploy fornisce un punto di accesso alternativo quando si utilizza un browser. Oltre a fornire un modo per eseguire direttamente singole chiamate API, la pagina include anche una descrizione dettagliata dell'API, inclusi i parametri di input e altre opzioni per ciascuna chiamata. Le chiamate API sono organizzate in diverse aree funzionali o categorie.

### Programma personalizzato

È possibile accedere all'API Deploy utilizzando diversi linguaggi di programmazione e strumenti. Tra i più diffusi ci sono Python, Java e cURL. Un programma, uno script o uno strumento che utilizza l'API funge da client di servizi web REST. L'utilizzo di un linguaggio di programmazione consente di comprendere meglio l'API e offre l'opportunità di automatizzare le distribuzioni ONTAP Select .

## Controllo delle versioni dell'API ONTAP Select Deploy

All'API REST inclusa in ONTAP Select Deploy viene assegnato un numero di versione. Il numero di versione dell'API è indipendente dal numero di release di Deploy. È necessario conoscere la versione dell'API inclusa nella release di Deploy e come questa potrebbe influire sull'utilizzo dell'API.

La versione corrente dell'utilità di amministrazione Deploy include la versione 3 dell'API REST. Le versioni precedenti dell'utilità Deploy includono le seguenti versioni dell'API:

### Distribuisci 2.8 e versioni successive

ONTAP Select Deploy 2.8 e tutte le versioni successive includono la versione 3 dell'API REST.

## Distribuisci 2.7.2 e versioni precedenti

ONTAP Select Deploy 2.7.2 e tutte le versioni precedenti includono la versione 2 dell'API REST.



Le versioni 2 e 3 dell'API REST non sono compatibili. Se si esegue l'aggiornamento a Deploy 2.8 o versioni successive da una versione precedente che include la versione 2 dell'API, è necessario aggiornare qualsiasi codice esistente che acceda direttamente all'API, nonché tutti gli script che utilizzano l'interfaccia a riga di comando.

# Caratteristiche operative di base dell'API ONTAP Select Deploy

Sebbene REST stabilisca un insieme comune di tecnologie e best practice, i dettagli di ciascuna API possono variare in base alle scelte progettuali. È necessario conoscere i dettagli e le caratteristiche operative dell'API ONTAP Select Deploy prima di utilizzarla.

## Host dell'hypervisor rispetto al nodo ONTAP Select

Un *host hypervisor* è la piattaforma hardware principale che ospita una macchina virtuale ONTAP Select. Quando una macchina virtuale ONTAP Select viene distribuita e attiva su un host hypervisor, la macchina virtuale viene considerata un *nodo ONTAP Select*. Con la versione 3 dell'API REST Deploy, gli oggetti host e nodo sono separati e distinti. Ciò consente una relazione uno-a-molti, in cui uno o più nodi ONTAP Select possono essere eseguiti sullo stesso host hypervisor.

## Identificatori di oggetti

A ogni istanza o oggetto di risorsa viene assegnato un identificatore univoco al momento della creazione. Questi identificatori sono globalmente univoci all'interno di una specifica istanza di ONTAP Select Deploy. Dopo aver eseguito una chiamata API che crea una nuova istanza di oggetto, il valore ID associato viene restituito al chiamante nel `location` Intestazione della risposta HTTP. È possibile estrarre l'identificatore e utilizzarlo nelle chiamate successive quando si fa riferimento all'istanza della risorsa.



Il contenuto e la struttura interna degli identificatori degli oggetti possono cambiare in qualsiasi momento. Si consiglia di utilizzare gli identificatori solo nelle chiamate API applicabili, quando necessario, quando si fa riferimento agli oggetti associati.

## Identificatori di richiesta

A ogni richiesta API andata a buon fine viene assegnato un identificatore univoco. L'identificatore viene restituito nel `request-id` Intestazione della risposta HTTP associata. È possibile utilizzare un identificatore di richiesta per fare riferimento collettivamente alle attività di una singola transazione API richiesta-risposta specifica. Ad esempio, è possibile recuperare tutti i messaggi di evento per una transazione in base all'ID di richiesta.

## Chiamate sincrone e asincrone

Esistono due modi principali in cui un server esegue una richiesta HTTP ricevuta da un client:

- Sincrono Il server esegue la richiesta immediatamente e risponde con un codice di stato 200, 201 o 204.
- Asincrono: il server accetta la richiesta e risponde con un codice di stato 202. Questo indica che il server ha accettato la richiesta del client e ha avviato un'attività in background per completarla. L'esito positivo o

negativo finale non è immediatamente disponibile e deve essere determinato tramite ulteriori chiamate API.

## Confermare il completamento di un lavoro di lunga durata

In genere, qualsiasi operazione che può richiedere molto tempo per essere completata viene elaborata in modo asincrono utilizzando un'attività in background sul server. Con l'API REST Deploy, ogni attività in background è ancorata a un oggetto Job che tiene traccia dell'attività e fornisce informazioni, come lo stato corrente. Un oggetto Job, incluso il suo identificatore univoco, viene restituito nella risposta HTTP dopo la creazione di un'attività in background.

È possibile interrogare direttamente l'oggetto Job per determinare l'esito positivo o negativo della chiamata API associata. Per ulteriori informazioni, consultare *elaborazione asincrona tramite l'oggetto Job*.

Oltre all'utilizzo dell'oggetto Job, esistono altri modi per determinare il successo o il fallimento di una richiesta, tra cui:

- Messaggi di evento È possibile recuperare tutti i messaggi di evento associati a una specifica chiamata API utilizzando l'ID della richiesta restituito con la risposta originale. I messaggi di evento in genere contengono un'indicazione di successo o fallimento e possono essere utili anche durante il debug di una condizione di errore.
- Stato o status della risorsa Molte risorse mantengono un valore di stato o status che è possibile interrogare per determinare indirettamente il successo o il fallimento di una richiesta.

## Sicurezza

L'API Deploy utilizza le seguenti tecnologie di sicurezza:

- Transport Layer Security: tutto il traffico inviato sulla rete tra il server e il client Deploy viene crittografato tramite TLS. L'utilizzo del protocollo HTTP su un canale non crittografato non è supportato. È supportata la versione 1.2 di TLS.
- Autenticazione HTTP: l'autenticazione di base viene utilizzata per ogni transazione API. Un'intestazione HTTP, che include nome utente e password in una stringa base64, viene aggiunta a ogni richiesta.

## Transazione API di richiesta e risposta per ONTAP Select

Ogni chiamata all'API Deploy viene eseguita come una richiesta HTTP alla macchina virtuale Deploy, che genera una risposta associata al client. Questa coppia richiesta/risposta è considerata una transazione API. Prima di utilizzare l'API Deploy, è necessario acquisire familiarità con le variabili di input disponibili per controllare una richiesta e il contenuto dell'output della risposta.

## Variabili di input che controllano una richiesta API

È possibile controllare il modo in cui viene elaborata una chiamata API tramite i parametri impostati nella richiesta HTTP.

### Intestazioni di richiesta

È necessario includere diverse intestazioni nella richiesta HTTP, tra cui:

- content-type Se il corpo della richiesta include JSON, questa intestazione deve essere impostata su

application/json.

- accetta Se il corpo della risposta includerà JSON, questa intestazione deve essere impostata su application/json.
- autorizzazione L'autenticazione di base deve essere impostata con il nome utente e la password codificati in una stringa base64.

## Corpo della richiesta

Il contenuto del corpo della richiesta varia a seconda della chiamata specifica. Il corpo della richiesta HTTP è costituito da uno dei seguenti elementi:

- Oggetto JSON con variabili di input (ad esempio, il nome di un nuovo cluster)
- Vuoto

## Filtra oggetti

Quando si esegue una chiamata API che utilizza GET, è possibile limitare o filtrare gli oggetti restituiti in base a qualsiasi attributo. Ad esempio, è possibile specificare un valore esatto da ricercare:

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

Oltre alla corrispondenza esatta, sono disponibili altri operatori per restituire un set di oggetti su un intervallo di valori. ONTAP Select supporta gli operatori di filtro mostrati di seguito.

Operatore	Descrizione
=	Uguale a
<	Meno di
>	Maggiore di
≤	Minore o uguale a
≥	Maggiore o uguale a
	O
!	Non uguale a
*	Jolly avido

È anche possibile restituire un set di oggetti in base al fatto che un campo specifico sia impostato o meno utilizzando la parola chiave null o la sua negazione (!null) come parte della query.

## Selezione dei campi oggetto

Per impostazione predefinita, l'emissione di una chiamata API tramite GET restituisce solo gli attributi che identificano in modo univoco l'oggetto o gli oggetti. Questo set minimo di campi funge da chiave per ciascun oggetto e varia in base al tipo di oggetto. È possibile selezionare proprietà aggiuntive dell'oggetto utilizzando il parametro di query fields nei seguenti modi:

- Campi economici Specificare `fields=*` per recuperare i campi oggetto che sono mantenuti nella memoria del server locale o per i quali è richiesta poca elaborazione per accedervi.
- Campi costosi Specificare `fields=**` per recuperare tutti i campi dell'oggetto, compresi quelli che

richiedono un'ulteriore elaborazione del server per accedervi.

- Selezione campo personalizzato Usa `fields=FIELDNAME` per specificare il campo esatto desiderato. Quando si richiedono più campi, i valori devono essere separati da virgole senza spazi.



Come buona pratica, dovresti sempre identificare i campi specifici che desideri. Recupera il set di campi economici o costosi solo quando necessario. La classificazione di campi economici e costosi è determinata da NetApp in base all'analisi interna delle prestazioni. La classificazione di un determinato campo può cambiare in qualsiasi momento.

## Ordina gli oggetti nel set di output

I record in una raccolta di risorse vengono restituiti nell'ordine predefinito definito dall'oggetto. È possibile modificare l'ordine utilizzando il parametro di query `order_by` con il nome del campo e la direzione di ordinamento come segue:

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

Ad esempio, è possibile ordinare il campo tipo in ordine decrescente seguito da id in ordine crescente:

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

Quando si includono più parametri, è necessario separare i campi con una virgola.

## Paginazione

Quando si effettua una chiamata API tramite GET per accedere a una raccolta di oggetti dello stesso tipo, per impostazione predefinita vengono restituiti tutti gli oggetti corrispondenti. Se necessario, è possibile limitare il numero di record restituiti utilizzando il parametro di query `max_records` nella richiesta. Ad esempio:

```
max_records=20
```

Se necessario, è possibile combinare questo parametro con altri parametri di query per restringere il set di risultati. Ad esempio, il seguente comando restituisce fino a 10 eventi di sistema generati dopo l'intervallo di tempo specificato:

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

È possibile inviare più richieste per scorrere gli eventi (o qualsiasi tipo di oggetto). Ogni successiva chiamata API dovrebbe utilizzare un nuovo valore temporale basato sull'evento più recente nell'ultimo set di risultati.

## Interpretare una risposta API

Ogni richiesta API genera una risposta al client. È possibile esaminare la risposta per determinare se è andata a buon fine e recuperare dati aggiuntivi se necessario.

### Codice di stato HTTP

Di seguito sono descritti i codici di stato HTTP utilizzati dall'API REST Deploy.

Codice	Senso	Descrizione
200	OK	Indica il successo delle chiamate che non creano un nuovo oggetto.
201	Creato	Un oggetto è stato creato correttamente; l'intestazione della risposta sulla posizione include l'identificatore univoco per l'oggetto.
202	Accettato	È stato avviato un processo in background di lunga durata per eseguire la richiesta, ma l'operazione non è ancora stata completata.

<b>Codice</b>	<b>Senso</b>	<b>Descrizione</b>
400	Brutta richiesta	L'input della richiesta non è riconosciuto o è inappropriato.
403	Vietato	L'accesso è negato a causa di un errore di autorizzazione.
404	Non trovato	La risorsa a cui si fa riferimento nella richiesta non esiste.
405	Metodo non consentito	Il verbo HTTP nella richiesta non è supportato per la risorsa.
409	Conflitto	Il tentativo di creare un oggetto non è riuscito perché l'oggetto esiste già.
500	Errore interno	Si è verificato un errore interno generale sul server.
501	Non implementato	L'URI è noto ma non è in grado di eseguire la richiesta.

### Intestazioni di risposta

Nella risposta HTTP generata dal server Deploy sono incluse diverse intestazioni, tra cui:

- request-id A ogni richiesta API riuscita viene assegnato un identificatore di richiesta univoco.
- posizione Quando viene creato un oggetto, l'intestazione posizione include l'URL completo del nuovo oggetto, incluso l'identificatore univoco dell'oggetto.

### Corpo della risposta

Il contenuto della risposta associata a una richiesta API varia in base all'oggetto, al tipo di elaborazione e all'esito positivo o negativo della richiesta. Il corpo della risposta viene visualizzato in formato JSON.

- Singolo oggetto. Un singolo oggetto può essere restituito con un set di campi in base alla richiesta. Ad esempio, è possibile utilizzare GET per recuperare proprietà selezionate di un cluster utilizzando l'identificatore univoco.
- Oggetti multipli È possibile restituire più oggetti da una raccolta di risorse. In tutti i casi, viene utilizzato un formato coerente, con num\_records Indica il numero di record e record contenenti un array delle istanze dell'oggetto. Ad esempio, è possibile recuperare tutti i nodi definiti in un cluster specifico.
- Oggetto Job: se una chiamata API viene elaborata in modo asincrono, viene restituito un oggetto Job che ancora l'attività in background. Ad esempio, la richiesta POST utilizzata per distribuire un cluster viene elaborata in modo asincrono e restituisce un oggetto Job.
- Oggetto Errore Se si verifica un errore, viene sempre restituito un oggetto Errore. Ad esempio, si riceverà un errore quando si tenta di creare un cluster con un nome già esistente.
- Vuoto In alcuni casi, non vengono restituiti dati e il corpo della risposta è vuoto. Ad esempio, il corpo della risposta è vuoto dopo aver utilizzato DELETE per eliminare un host esistente.

## Elaborazione asincrona tramite l'oggetto Job per ONTAP Select

Alcune chiamate API Deploy, in particolare quelle che creano o modificano una risorsa, possono richiedere più tempo rispetto ad altre. ONTAP Select Deploy elabora queste richieste di lunga durata in modo asincrono.

## Richieste asincrone descritte utilizzando l'oggetto Job

Dopo aver effettuato una chiamata API eseguita in modo asincrono, il codice di risposta HTTP 202 indica che la richiesta è stata convalidata e accettata correttamente, ma non è ancora stata completata. La richiesta viene elaborata come un'attività in background che continua a essere eseguita dopo la risposta HTTP iniziale al client. La risposta include l'oggetto Job che ancora la richiesta, incluso il suo identificatore univoco.



Per determinare quali chiamate API funzionano in modo asincrono, fare riferimento alla pagina della documentazione online ONTAP Select Deploy.

## Interroga l'oggetto Job associato a una richiesta API

L'oggetto Job restituito nella risposta HTTP contiene diverse proprietà. È possibile interrogare la proprietà di stato per determinare se la richiesta è stata completata correttamente. Un oggetto Job può trovarsi in uno dei seguenti stati:

- In coda
- Corsa
- Successo
- Fallimento

Esistono due tecniche che è possibile utilizzare quando si interroga un oggetto Job per rilevare uno stato terminale per l'attività, ovvero successo o fallimento:

- Richiesta di polling standard Lo stato del lavoro corrente viene restituito immediatamente
- Richiesta di polling lungo Lo stato del lavoro viene restituito solo quando si verifica una delle seguenti situazioni:
  - Lo stato è cambiato più di recente rispetto al valore data-ora fornito nella richiesta di sondaggio
  - Il valore di timeout è scaduto (da 1 a 120 secondi)

Il polling standard e il polling lungo utilizzano la stessa chiamata API per interrogare un oggetto Job. Tuttavia, una richiesta di polling lungo include due parametri di query: `poll_timeout` E `last_modified`.



È consigliabile utilizzare sempre il polling lungo per ridurre il carico di lavoro sulla macchina virtuale Deploy.

## Procedura generale per l'emissione di una richiesta asincrona

Per completare una chiamata API asincrona, è possibile utilizzare la seguente procedura di alto livello:

1. Emettere la chiamata API asincrona.
2. Ricevere una risposta HTTP 202 che indica l'accettazione corretta della richiesta.
3. Estrarre l'identificatore per l'oggetto Job dal corpo della risposta.
4. All'interno di un ciclo, eseguire quanto segue in ogni ciclo:
  - a. Ottieni lo stato attuale del lavoro con una richiesta di sondaggio lungo
  - b. Se il Job è in uno stato non terminale (in coda, in esecuzione), eseguire nuovamente il ciclo.
5. Interrompere quando il lavoro raggiunge uno stato terminale (successo, fallimento).

## **Informazioni sul copyright**

Copyright © 2026 NetApp, Inc. Tutti i diritti riservati. Stampato negli Stati Uniti d'America. Nessuna porzione di questo documento soggetta a copyright può essere riprodotta in qualsiasi formato o mezzo (grafico, elettronico o meccanico, inclusi fotocopie, registrazione, nastri o storage in un sistema elettronico) senza previo consenso scritto da parte del detentore del copyright.

Il software derivato dal materiale sottoposto a copyright di NetApp è soggetto alla seguente licenza e dichiarazione di non responsabilità:

IL PRESENTE SOFTWARE VIENE FORNITO DA NETAPP "COSÌ COM'È" E SENZA QUALSIVOGLIA TIPO DI GARANZIA IMPLICITA O ESPRESSA FRA CUI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, GARANZIE IMPLICITE DI COMMERCIALITÀ E IDONEITÀ PER UNO SCOPO SPECIFICO, CHE VENGONO DECLINATE DAL PRESENTE DOCUMENTO. NETAPP NON VERRÀ CONSIDERATA RESPONSABILE IN ALCUN CASO PER QUALSIVOGLIA DANNO DIRETTO, INDIRETTO, ACCIDENTALE, SPECIALE, ESEMPLARE E CONSEGUENZIALE (COMPRESI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, PROCUREMENT O SOSTITUZIONE DI MERCI O SERVIZI, IMPOSSIBILITÀ DI UTILIZZO O PERDITA DI DATI O PROFITTI OPPURE INTERRUZIONE DELL'ATTIVITÀ AZIENDALE) CAUSATO IN QUALSIVOGLIA MODO O IN RELAZIONE A QUALUNQUE TEORIA DI RESPONSABILITÀ, SIA ESSA CONTRATTUALE, RIGOROSA O DOVUTA A INSOLVENZA (COMPRESA LA NEGLIGENZA O ALTRO) INSORTA IN QUALSIASI MODO ATTRAVERSO L'UTILIZZO DEL PRESENTE SOFTWARE ANCHE IN PRESENZA DI UN PREAVVISO CIRCA L'EVENTUALITÀ DI QUESTO TIPO DI DANNI.

NetApp si riserva il diritto di modificare in qualsiasi momento qualunque prodotto descritto nel presente documento senza fornire alcun preavviso. NetApp non si assume alcuna responsabilità circa l'utilizzo dei prodotti o materiali descritti nel presente documento, con l'eccezione di quanto concordato espressamente e per iscritto da NetApp. L'utilizzo o l'acquisto del presente prodotto non comporta il rilascio di una licenza nell'ambito di un qualche diritto di brevetto, marchio commerciale o altro diritto di proprietà intellettuale di NetApp.

Il prodotto descritto in questa guida può essere protetto da uno o più brevetti degli Stati Uniti, esteri o in attesa di approvazione.

**LEGENDA PER I DIRITTI SOTTOPOSTI A LIMITAZIONE:** l'utilizzo, la duplicazione o la divulgazione da parte degli enti governativi sono soggetti alle limitazioni indicate nel sottoparagrafo (b)(3) della clausola Rights in Technical Data and Computer Software del DFARS 252.227-7013 (FEB 2014) e FAR 52.227-19 (DIC 2007).

I dati contenuti nel presente documento riguardano un articolo commerciale (secondo la definizione data in FAR 2.101) e sono di proprietà di NetApp, Inc. Tutti i dati tecnici e il software NetApp forniti secondo i termini del presente Contratto sono articoli aventi natura commerciale, sviluppati con finanziamenti esclusivamente privati. Il governo statunitense ha una licenza irrevocabile limitata, non esclusiva, non trasferibile, non cedibile, mondiale, per l'utilizzo dei Dati esclusivamente in connessione con e a supporto di un contratto governativo statunitense in base al quale i Dati sono distribuiti. Con la sola esclusione di quanto indicato nel presente documento, i Dati non possono essere utilizzati, divulgati, riprodotti, modificati, visualizzati o mostrati senza la previa approvazione scritta di NetApp, Inc. I diritti di licenza del governo degli Stati Uniti per il Dipartimento della Difesa sono limitati ai diritti identificati nella clausola DFARS 252.227-7015(b) (FEB 2014).

## **Informazioni sul marchio commerciale**

NETAPP, il logo NETAPP e i marchi elencati alla pagina <http://www.netapp.com/TM> sono marchi di NetApp, Inc. Gli altri nomi di aziende e prodotti potrebbero essere marchi dei rispettivi proprietari.