Concetti

ONTAP Select

NetApp
November 08, 2026

This PDF was generated from https://docs.netapp.com/it-it/ontap-select-9191/concept_api_rest.html on November 08, 2026. Always check docs.netapp.com for the latest.

Sommario

Concetti . . . 1
- Scopri i servizi web REST per distribuire e gestire i cluster ONTAP Select . . . 1
  - Architettura e vincoli classici . . . 1
  - Risorse e rappresentazione dello stato . . . 1
  - Endpoint URI . . . 1
  - messaggi HTTP . . . 1
  - Formattazione JSON . . . 2
- Scopri come accedere all'API ONTAP Select Deploy . . . 2
  - Distribuisci l'interfaccia utente nativa dell'utilità . . . 2
  - Pagina della documentazione online di ONTAP Select Deploy . . . 2
  - Programma personalizzato . . . 2
- Caratteristiche operative di base dell'API ONTAP Select Deploy . . . 2
  - Host hypervisor rispetto al nodo ONTAP Select . . . 2
  - Identificatori degli oggetti . . . 3
  - Identificativi della richiesta . . . 3
  - Chiamate sincrone e asincrone . . . 3
  - Confermare il completamento di un lavoro di lunga durata . . . 3
  - 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 tramite l'oggetto Job . . . 7
  - Interroga l'oggetto Job associato a una richiesta API . . . 7
  - Procedura generale per l'invio di una richiesta asincrona . . . 8

Concetti

Scopri i servizi web REST per distribuire e gestire i cluster ONTAP Select

Representational State Transfer (REST) è uno stile per la creazione di applicazioni web distribuite. Applicato alla progettazione di un'API per servizi web, definisce un insieme di tecnologie e best practice per esporre risorse basate su server e gestirne gli stati. Utilizza protocolli e standard di settore per fornire una base flessibile per la distribuzione e la gestione dei cluster ONTAP Select.

Architettura e vincoli classici

REST è stato formalmente articolato da Roy Fielding nella sua tesi di dottorato "dissertazione" presso UC Irvine nel 2000. Definisce uno stile architetturale attraverso un insieme di vincoli, che collettivamente hanno migliorato le applicazioni web-based e i protocolli sottostanti. I vincoli stabiliscono un'applicazione di servizi web RESTful basata su un'architettura client/server utilizzando un protocollo di comunicazione stateless.

Risorse e rappresentazione dello stato

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 REST web services è identificare le risorse.
- Definizione degli stati delle risorse e delle relative operazioni di stato. Le risorse si trovano sempre in uno di un numero finito di stati. Gli stati, così come le operazioni associate utilizzate per effettuare i cambiamenti di stato, devono essere definiti in modo chiaro.

I messaggi vengono scambiati tra client e server per accedere alle risorse e modificarne lo stato, secondo il modello generico CRUD (Create, Read, Update, and Delete).

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 ogni 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

L'Hypertext Transfer Protocol (HTTP) è il protocollo utilizzato dal client e dal server dei servizi web per scambiare messaggi di richiesta e risposta relativi alle risorse. Nell'ambito della 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 stateless. Pertanto, per associare un insieme di richieste e risposte correlate in un'unica transazione, è necessario includere informazioni aggiuntive nelle intestazioni HTTP che accompagnano i flussi di dati di

richiesta/risposta.

Formattazione JSON

Sebbene le informazioni possano essere strutturate e trasferite tra client e 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 di settore per la rappresentazione di semplici strutture dati in testo semplice e viene utilizzato per trasferire informazioni sullo stato che descrivono le risorse.

Scopri 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.

> L'API REST inclusa in ONTAP Select Deploy ha un numero di versione assegnato. Il numero di versione dell'API è indipendente dal numero di rilascio di Deploy. L'utility di amministrazione ONTAP Select 9.19.1 Deploy include la versione 3 dell'API REST.

Distribuisci l'interfaccia utente nativa dell'utilità

Il metodo principale per accedere all'API è tramite l'interfaccia utente web di 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'utility Deploy.

Pagina della documentazione online di ONTAP Select Deploy

La pagina della documentazione online di ONTAP Select Deploy offre un punto di accesso alternativo quando si utilizza un browser. Oltre a consentire l'esecuzione diretta di singole chiamate API, la pagina include anche una descrizione dettagliata dell'API, compresi 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 di Deploy utilizzando diversi linguaggi di programmazione e strumenti. Tra le scelte più popolari figurano Python, Java e cURL. Un programma, uno script o uno strumento che utilizza l'API funge da client per i servizi web REST. L'utilizzo di un linguaggio di programmazione consente di comprendere meglio l'API e offre l'opportunità di automatizzare le distribuzioni di ONTAP Select.

Caratteristiche operative di base dell'API ONTAP Select Deploy

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

Host hypervisor rispetto al nodo ONTAP Select

Un *hypervisor host* è la piattaforma hardware principale che ospita una macchina virtuale ONTAP Select.

Quando una macchina virtuale ONTAP Select viene distribuita e attivata su un host hypervisor, la macchina virtuale è considerata un *nodo ONTAP Select*. Con la versione 3 della REST API Deploy, gli oggetti host e nodo sono separati e distinti. Questo consente una relazione uno-a-molti, dove uno o più nodi ONTAP Select possono essere eseguiti sullo stesso host hypervisor.

Identificatori degli oggetti

A ogni istanza o oggetto risorsa viene assegnato un identificatore univoco al momento della creazione. Questi identificatori sono univoci a livello globale all'interno di una specifica istanza di ONTAP Select Deploy. Dopo aver effettuato una chiamata API che crea una nuova istanza di oggetto, il valore id associato viene restituito al chiamante nell' `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. È opportuno utilizzare gli identificatori solo nelle chiamate API pertinenti, quando necessario, per fare riferimento agli oggetti associati.

Identificativi della richiesta

A ogni richiesta API andata a buon fine viene assegnato un identificatore univoco. L'identificatore viene restituito nell' `request-id`intestazione della risposta HTTP associata. È possibile utilizzare un identificatore di richiesta per fare riferimento collettivamente alle attività di una singola specifica transazione richiesta-risposta API. Ad esempio, è possibile recuperare tutti i messaggi di evento per una transazione in base all'ID della 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 completare la richiesta. L'esito finale, positivo o negativo, 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 richieda molto tempo per essere completata viene elaborata in modo asincrono tramite un'attività in background sul server. Con la Deploy REST API, ogni attività in background è associata a un oggetto Job che ne tiene traccia 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. Consultare *elaborazione asincrona tramite l'oggetto Job* per ulteriori informazioni.

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

- Messaggi di evento Puoi 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 per il debug di una condizione di errore.

- Stato o stato della risorsa Diverse risorse mantengono un valore di stato o di stato 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:

- Sicurezza del livello di trasporto (TLS): tutto il traffico inviato sulla rete tra il server Deploy e il client è crittografato tramite TLS. L'utilizzo del protocollo HTTP su un canale non crittografato non è supportato. È supportata la versione 1.2 di TLS.
- L'autenticazione HTTP Basic viene utilizzata per ogni transazione API. A ogni richiesta viene aggiunta un'intestazione HTTP che include nome utente e password in una stringa base64.

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 per il client. Questa coppia richiesta/risposta è considerata una transazione API. Prima di utilizzare l'API Deploy, è necessario familiarizzare con le variabili di input disponibili per controllare una richiesta e con 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 della 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.
- accept Se il corpo della risposta include JSON, questa intestazione deve essere impostata su application/json.
- L'autenticazione Basic deve essere impostata con nome utente e 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 effettua 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 confrontare:

`<field>=<query value>`

Oltre alla corrispondenza esatta, sono disponibili altri operatori per restituire un insieme 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 inoltre possibile restituire un insieme 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, 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 ulteriori proprietà 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 locale del server o che richiedono poca elaborazione per l'accesso.
- Campi costosi Specifica `fields=**` per recuperare tutti i campi dell'oggetto, inclusi quelli che richiedono un'elaborazione aggiuntiva da parte del server per l'accesso.
- Selezione personalizzata dei campi Usa `fields=FIELDNAME` per specificare esattamente il campo desiderato. Quando si richiedono più campi, i valori devono essere separati da virgole senza spazi.



Come best practice, è sempre consigliabile identificare i campi specifici di cui si ha bisogno. È opportuno recuperare l'insieme dei campi economici o costosi solo quando necessario. La classificazione in campi economici e costosi è determinata da NetApp in base ad analisi interne 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 dal campo 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, la seguente query restituisce fino a 10 eventi di sistema generati dopo l'ora specificata: `time⇒ 2019-04-04T15:41:29.140265Z&max_records=10`

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

Interpretare una risposta API

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

codice di stato HTTP

Di seguito vengono descritti i codici di stato HTTP utilizzati dalla Deploy REST API.

Codice	Significato	Descrizione
200	OK	Indica il successo delle chiamate che non creano un nuovo oggetto.
201	Creato	L'oggetto è stato creato correttamente; l'intestazione della risposta relativa alla posizione include l'identificativo univoco dell'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.
400	Richiesta non valida	L'input della richiesta non è riconosciuto o non è appropriato.
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 generico 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:

- ID richiesta A ogni richiesta API andata a buon fine viene assegnato un identificatore di richiesta univoco.

- location Quando viene creato un oggetto, l'intestazione location include l'URL completo del nuovo oggetto, compreso l'identificativo univoco dell'oggetto.

Corpo di 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.

- Oggetto singolo Un singolo oggetto può essere restituito con un insieme di campi in base alla richiesta. Ad esempio, è possibile utilizzare GET per recuperare proprietà selezionate di un cluster utilizzando l'identificativo univoco.
- Oggetti multipli Più oggetti da una raccolta di risorse possono essere restituiti. In tutti i casi, viene utilizzato un formato coerente, con `num_records` che indica il numero di record e i record che contengono un array delle istanze degli oggetti. Ad esempio, è possibile recuperare tutti i nodi definiti in uno specifico cluster.
- Oggetto Job Se una chiamata API viene elaborata in modo asincrono, viene restituito un oggetto Job che funge da punto di ancoraggio per 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 Error: Se si verifica un errore, viene sempre restituito un oggetto Error. Ad esempio, si riceverà un errore quando si tenta di creare un cluster con un nome già esistente.
- 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 di Deploy, in particolare quelle che creano o modificano una risorsa, possono richiedere più tempo per essere completate rispetto ad altre chiamate. ONTAP Select Deploy elabora queste richieste di lunga durata in modo asincrono.

Richieste asincrone descritte tramite l'oggetto Job

Dopo aver effettuato una chiamata API che viene 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 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, compreso il suo identificatore univoco.



Dovresti consultare la pagina della documentazione online di ONTAP Select Deploy per determinare quali chiamate API operano in modo asincrono.

Interroga l'oggetto Job associato a una richiesta API

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

- In coda
- In esecuzione

- Successo
- Fallimento

Esistono due tecniche che è possibile utilizzare per interrogare un oggetto Job e rilevare lo stato finale dell'attività, ovvero successo o fallimento:

- Richiesta di polling standard Lo stato attuale del job viene restituito immediatamente
- Richiesta di long polling Lo stato del job viene restituito solo quando si verifica una delle seguenti condizioni:
  - Lo stato è cambiato più recentemente 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 long polling utilizzano la stessa chiamata API per interrogare un oggetto Job. Tuttavia, una richiesta di long polling include due parametri di query: `poll_timeout` e `last_modified`.



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

Procedura generale per l'invio di una richiesta asincrona

È possibile utilizzare la seguente procedura high-level per completare una chiamata API asincrona:

1. Effettua la chiamata API asincrona.
2. Ricevi una risposta HTTP 202 che indica l'accettazione della richiesta.
3. Estrai l'identificativo dell'oggetto Job dal corpo della risposta.
4. All'interno di un ciclo, eseguire le seguenti operazioni in ogni ciclo:
   a. Ottieni lo stato attuale del Job con una richiesta di long-poll
   b. Se il lavoro si trova in uno stato non terminale (in coda, in esecuzione), ripetere il ciclo.
5. Interrompere l'esecuzione quando il Job 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 COMMERCIABILITÀ 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 CONSEQUENZIALE (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.