Concetti

ONTAP Select

NetApp
January 31, 2025

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

Sommario

Concetti 1
Base REST per i web Services 1
Come accedere all’API di implementazione 2
Implementare la versione delle API 2
Caratteristiche operative di base 3
Transazione API di richiesta e risposta 4
Elaborazione asincrona utilizzando l’oggetto lavoro 8

Concetti

Base REST per i web Services

Representational state Transfer (REST) è uno stile per la creazione di applicazioni web distribuite. Quando viene applicato alla progettazione di un'API di servizi Web, stabilisce un insieme di tecnologie e Best practice per esporre le risorse basate su server e gestirne gli stati. Utilizza protocolli e standard mainstream per fornire una base flessibile per l'implementazione e la gestione dei cluster ONTAP Select.

Architettura e limiti classici

REST è stato formalmente articolato da Roy Fielding nel suo dottorato "dissertazione" Presso UC Irvine nel 2000. Definisce uno stile architettonico attraverso una serie di vincoli, che collettivamente hanno migliorato le applicazioni basate sul 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 rappresentazione dello stato

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

- Identificazione delle risorse di sistema o basate su server
  Ogni sistema utilizza e gestisce le risorse. Una risorsa può essere un file, una transazione di business, un processo o un'entità amministrativa. Una delle prime attività nella progettazione di un'applicazione basata sui servizi web REST è quella di identificare le risorse.
- Definizione degli stati delle risorse e delle operazioni di stato associate
  Le risorse si trovano sempre in un numero limitato 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 in base al modello generico CRUD (Create, Read, Update e Delete).

Endpoint URI

Ogni risorsa REST deve essere definita e resa disponibile utilizzando uno schema di indirizzamento ben definito. Gli endpoint in cui sono situate e identificate le risorse utilizzano un URI (Uniform Resource Identifier). L'URI fornisce un framework generale per la creazione di un nome univoco per ogni risorsa nella rete. L'URL (Uniform Resource Locator) 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

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 azioni di gestione dello stato corrispondenti.

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 portate 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 l'API DI DISTRIBUZIONE REST) è JavaScript Object Notation (JSON). JSON è uno standard di settore per la rappresentazione di semplici strutture di dati in testo normale e viene utilizzato per trasferire informazioni di stato che descrivono le risorse.

Come accedere all'API di implementazione

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

Implementare l'interfaccia utente nativa dell'utility

Il modo 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 alla progettazione dell'interfaccia utente. È possibile accedere all'API anche tramite l'interfaccia della riga di comando dell'utilità di implementazione.

Pagina della documentazione online di ONTAP Select Deploy

La pagina della documentazione online di ONTAP Select Deploy fornisce un access point alternativo quando si utilizza un browser. Oltre a fornire un modo per eseguire direttamente le 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 di distribuzione utilizzando uno dei diversi linguaggi e tool di programmazione. Le scelte più popolari includono Python, Java e CURL. Un programma, uno script o uno strumento che utilizza l'API agisce come un 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 implementazioni di ONTAP Select.

Implementare la versione delle API

AIL'API REST inclusa nella distribuzione ONTAP Select viene assegnato un numero di versione. Il numero di versione dell'API è indipendente dal numero di release di implementazione. È necessario conoscere la versione dell'API inclusa nella release di deploy e il modo in cui questo potrebbe influire sull'utilizzo dell'API.

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

Implementare 2.8 e versioni successive

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

Implementare 2.7.2 e versioni precedenti

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

Le versioni 2 e 3 dell'API REST non sono compatibili. Se si esegue l'aggiornamento per la distribuzione della versione 2.8 o successiva da una release precedente che include la versione 2 dell'API, è necessario aggiornare qualsiasi codice esistente che acceda direttamente all'API e qualsiasi script che utilizzi l'interfaccia della riga di comando.

Caratteristiche operative di base

Mentre REST stabilisce un insieme comune di tecnologie e Best practice, i dettagli di ciascuna API possono variare in base alle scelte di progettazione. Prima di utilizzare l'API, è necessario conoscere i dettagli e le caratteristiche operative dell'API di implementazione di ONTAP Select.

Host 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 implementata e attiva su un host hypervisor, la macchina virtuale viene considerata un *nodo ONTAP Select*. Con la versione 3 dell'API REST di implementazione, 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 univoci a livello globale 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 in `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 di oggetti possono cambiare in qualsiasi momento. È necessario utilizzare gli identificatori delle chiamate API applicabili solo se necessario quando si fa riferimento agli oggetti associati.

Richiedi identificatori

A ogni richiesta API riuscita viene assegnato un identificatore univoco. L'identificatore viene restituito in `request-id` Intestazione della risposta HTTP associata. È possibile utilizzare un identificatore di richiesta per fare riferimento collettivamente alle attività di una singola transazione richiesta-risposta API specifica. 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 immediatamente la richiesta 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. Indica che il server ha accettato la richiesta del client e ha avviato un'attività in background per completare la richiesta. Il successo o l'errore finale non sono immediatamente disponibili e devono essere determinati tramite chiamate API aggiuntive.

Confermare il completamento di un lavoro a esecuzione prolungata

Generalmente, 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 di distribuzione, ogni attività in background è ancorata da un Oggetto lavoro che tiene traccia dell'attività e fornisce informazioni, ad esempio lo stato corrente. Un oggetto lavoro,
Dopo la creazione di un'attività in background, nella risposta HTTP viene restituito il relativo identificatore univoco.

È possibile eseguire query direttamente sull'oggetto Job per determinare il successo o l'errore della chiamata API associata.
Per ulteriori informazioni, fare riferimento a *elaborazione asincrona mediante l'oggetto Job*.

Oltre a utilizzare l'oggetto lavoro, esistono altri modi per determinare il successo o il fallimento di un richieste che includono:

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

Sicurezza

L'API di implementazione utilizza le seguenti tecnologie di sicurezza:

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

Transazione API di richiesta e risposta

Ogni chiamata API di implementazione viene eseguita come richiesta HTTP alla macchina virtuale di implementazione, che genera una risposta associata al client. Questa coppia di richieste/risposte è considerata una transazione API. Prima di utilizzare l'API di distribuzione, è necessario conoscere 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 la modalità di elaborazione di una chiamata API tramite i parametri impostati nella richiesta HTTP.

Intestazioni delle richieste

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

- tipo di contenuto
  Se il corpo della richiesta include JSON, questa intestazione deve essere impostata su application/json.
- accetta
  Se il corpo della risposta include 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 in base alla 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 associare:

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

Oltre a una corrispondenza esatta, sono disponibili altri operatori per restituire un set di oggetti su un intervallo di valori. ONTAP Select supporta gli operatori di filtraggio indicati di seguito.

Operatore	Descrizione
=	Uguale a.
<	Inferiore a.
>	Maggiore di
⇐	Minore o uguale a.
>=	Maggiore o uguale a.
|	Oppure
!	Non uguale a.
*	Goloso carattere jolly

È inoltre possibile restituire un insieme di oggetti in base all'impostazione o meno di un campo specifico utilizzando la parola chiave Null o la relativa negazione (!null) come parte della query.

Selezione dei campi oggetto

Per impostazione predefinita, l'emissione di una chiamata API utilizzando GET restituisce solo gli attributi che identificano in modo univoco lo o gli oggetti. Questo insieme 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 dei campi nei seguenti modi:

- Campi economici
  Specificare `fields=*` per recuperare i campi oggetto gestiti nella memoria del server locale o che richiedono un'elaborazione ridotta per l'accesso.
- Campi costosi
  Specificare `fields=**` per recuperare tutti i campi oggetto, inclusi quelli che richiedono un'ulteriore elaborazione del server per l'accesso.
- Selezione di campi personalizzati
  Utilizzare `fields=FIELDNAME` per specificare il campo esatto desiderato. Quando si richiedono più campi, i valori devono essere separati utilizzando virgole senza spazi.

Come Best practice, devi sempre identificare i campi specifici che desideri. È necessario recuperare l'insieme di campi economici o costosi solo quando necessario. La classificazione economica e costosa è determinata da NetApp in base all'analisi interna delle performance. La classificazione di un dato campo può cambiare in qualsiasi momento.

Ordinare gli oggetti nel set di output

I record di 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.

Impaginazione

Quando si esegue una chiamata API utilizzando GET per accedere a un insieme di oggetti dello stesso tipo, vengono restituiti tutti gli oggetti corrispondenti per impostazione predefinita. Se necessario, è possibile limitare il numero di record restituiti utilizzando il parametro di query max_records con la 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, quanto segue restituisce fino a 10 eventi di sistema generati dopo il 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 deve utilizzare un nuovo valore temporale basato sull'ultimo evento dell'ultimo set di risultati.

Interpretare una risposta API

Ogni richiesta API genera una risposta al client. È possibile esaminare la risposta per determinare se ha avuto esito positivo e ha recuperato ulteriori dati in base alle necessità.

Codice di stato HTTP

I codici di stato HTTP utilizzati dall'API DI DISTRIBUZIONE REST sono descritti di seguito.

Codice	Significato	Descrizione
200	OK	Indica che le chiamate che non creano un nuovo oggetto sono riuscite.
201	Creato	Un oggetto è stato creato correttamente; l'intestazione della risposta di posizione include l'identificatore univoco dell'oggetto.
202	Accettato	Per eseguire la richiesta è stato avviato un processo in background a esecuzione prolungata, ma l'operazione non è ancora stata completata.
400	Richiesta errata	L'input della richiesta non viene riconosciuto o non è appropriato.
403	Vietato	Accesso 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	Tentativo di creazione di un oggetto non riuscito perché l'oggetto esiste già.
500	Errore interno	Si è verificato un errore interno generale nel server.
501	Non implementato	L'URI è noto ma non è in grado di eseguire la richiesta.

Intestazioni delle risposte

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

- id richiesta
  A ogni richiesta API riuscita viene assegnato un identificatore univoco della richiesta.
- posizione
  Quando viene creato un oggetto, l'intestazione della posizione include l'URL completo del nuovo oggetto, incluso l'identificatore 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 rendering del corpo di risposta viene eseguito in JSON.

- Oggetto singolo
  È possibile restituire un singolo oggetto con un set di campi in base alla richiesta. AD esempio, È possibile utilizzare GET per recuperare le proprietà selezionate di un cluster utilizzando l'identificatore univoco.
- Oggetti multipli
  È possibile restituire più oggetti di una raccolta di risorse. In tutti i casi, viene utilizzato un formato coerente, con `num_records` indica il numero di record e record che contengono una matrice delle istanze dell'oggetto. Ad esempio, è possibile recuperare tutti i nodi definiti in un cluster specifico.
- Oggetto lavoro
  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 implementare un cluster viene elaborata in modo asincrono e restituisce un oggetto Job.
- Oggetto di errore
  Se si verifica un errore, viene sempre restituito un oggetto Error. Ad esempio, quando si tenta di creare un

cluster con un nome già esistente, viene visualizzato un messaggio di errore.

- Vuoto
  In alcuni casi, non viene restituito alcun dato 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 utilizzando l'oggetto lavoro

Alcune delle chiamate API di implementazione, in particolare quelle che creano o modificano una risorsa, possono richiedere più tempo per il completamento rispetto ad altre chiamate. ONTAP Select Deploy elabora queste richieste a esecuzione prolungata 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 ancora 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, incluso il relativo identificatore univoco.

Fare riferimento alla pagina della documentazione online di ONTAP Select Deploy per determinare quali chiamate API funzionano in modo asincrono.

Eseguire una query sull'oggetto Job associato a una richiesta API

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

- In coda
- In esecuzione
- Successo
- Guasto

Esistono due tecniche che è possibile utilizzare quando si esegue il polling di un oggetto Job per rilevare lo stato di un terminale per l'attività, ovvero riuscito o non riuscito:

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

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

Per ridurre il carico di lavoro sulla macchina virtuale di implementazione, è necessario utilizzare sempre il polling lungo.

Procedura generale per l’emissione di una richiesta asincrona

È possibile utilizzare la seguente procedura di alto livello per completare una chiamata API asincrona:

1. Eseguire la chiamata API asincrona.
2. Ricevere una risposta HTTP 202 che indichi la corretta accettazione della richiesta.
3. Estrarre l’identificatore per l’oggetto Job dal corpo della risposta.
4. All’interno di un loop, eseguire le seguenti operazioni in ogni ciclo:
    a. Ottenere lo stato corrente del lavoro con una richiesta di polling lungo
    b. Se il job si trova in uno stato non terminale (in coda, in esecuzione), eseguire nuovamente il loop.
5. Interrompere quando il lavoro raggiunge uno stato terminale (successo, errore).

Informazioni sul copyright

Copyright © 2025 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.