



## **Database open source**

NetApp Solutions

NetApp  
April 26, 2024

This PDF was generated from [https://docs.netapp.com/it-it/netapp-solutions/databases/aws\\_postgres\\_fsx\\_ec2\\_hadr.html](https://docs.netapp.com/it-it/netapp-solutions/databases/aws_postgres_fsx_ec2_hadr.html) on April 26, 2024. Always check docs.netapp.com for the latest.

# Sommario

- Database open source ..... 1
  - TR-4956: Implementazione automatizzata di alta disponibilità PostgreSQL e disaster recovery in AWS
- FSX/EC2 ..... 1
- TR-4722: Database MySQL su Best practice NetApp ONTAP ..... 10

# Database open source

## TR-4956: Implementazione automatizzata di alta disponibilità PostgreSQL e disaster recovery in AWS FSX/EC2

Allen Cao, Niyaz Mohamed, NetApp

### Scopo

PostgreSQL è un database open-source ampiamente utilizzato, classificato al quarto posto tra i primi dieci motori di database più diffusi "[MOTORI DB](#)". Da un lato, PostgreSQL deriva la sua popolarità dal suo modello open-source senza licenza, pur possedendo funzionalità sofisticate. D'altro canto, poiché è open source, mancano indicazioni dettagliate sull'implementazione di database di livello produzione nell'area dell'alta disponibilità e del disaster recovery (ha/DR), in particolare nel cloud pubblico. In generale, può essere difficile configurare un tipico sistema ha/DR PostgreSQL con standby a caldo e a caldo, replica in streaming e così via. Il test dell'ambiente ha/DR promuovendo il sito di standby e quindi il ritorno al primario può interrompere la produzione. Esistono problemi di performance ben documentati sul primario quando i carichi di lavoro di lettura vengono implementati nello streaming hot standby.

In questa documentazione, dimostreremo come eliminare una soluzione di streaming ha/DR PostgreSQL a livello di applicazione e creare una soluzione ha/DR PostgreSQL basata sullo storage AWS FSX ONTAP e sulle istanze di calcolo EC2 utilizzando la replica a livello di storage. La soluzione crea un sistema più semplice e paragonabile e offre risultati equivalenti rispetto alla replica in streaming a livello di applicazione PostgreSQL tradizionale per ha/DR.

Questa soluzione si basa su una tecnologia di replica a livello di storage NetApp SnapMirror collaudata e matura, disponibile nello storage cloud FSX ONTAP nativo di AWS per PostgreSQL ha/DR. È semplice da implementare con un toolkit di automazione fornito dal team delle soluzioni NetApp. Offre funzionalità simili eliminando la complessità e il trascinamento delle performance sul sito primario con la soluzione ha/DR basata su streaming a livello applicativo. La soluzione può essere facilmente implementata e testata senza influire sul sito primario attivo.

Questa soluzione risolve i seguenti casi di utilizzo:

- Implementazione ha/DR di livello produzione per PostgreSQL nel cloud AWS pubblico
- Test e convalida di un carico di lavoro PostgreSQL nel cloud AWS pubblico
- Test e convalida di una strategia ha/DR PostgreSQL basata sulla tecnologia di replica SnapMirror di NetApp

### Pubblico

Questa soluzione è destinata alle seguenti persone:

- Il DBA interessato all'implementazione di PostgreSQL con ha/DR nel cloud AWS pubblico.
- L'architetto della soluzione di database che è interessato a testare i workload PostgreSQL nel cloud pubblico AWS.
- L'amministratore dello storage interessato all'implementazione e alla gestione delle istanze PostgreSQL distribuite nello storage AWS FSX.

- Il proprietario dell'applicazione interessato a creare un ambiente PostgreSQL in AWS FSX/EC2.

## Ambiente di test e convalida della soluzione

Il test e la convalida di questa soluzione sono stati eseguiti in un ambiente AWS FSX e EC2 che potrebbe non corrispondere all'ambiente di implementazione finale. Per ulteriori informazioni, vedere la sezione [\[Key Factors for Deployment Consideration\]](#).

### Architettura

#### Componenti hardware e software

Hardware		
Storage FSX ONTAP	Versione corrente	Due coppie FSX ha nello stesso VPC e nella stessa zona di disponibilità dei cluster ha primari e di standby
Istanza EC2 per il calcolo	t2.xlarge/4vCPU/16G	Due EC2 T2 xlarge come istanze di calcolo primarie e di standby
Controller Ansible	CentOS VM/4vCPU/8G on-premise	Una macchina virtuale per ospitare il controller di automazione Ansible on-premise o nel cloud
Software		
RedHat Linux	RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2	Implementazione dell'abbonamento a RedHat per il test
CentOS Linux	CentOS Linux release 8.2.2004 (Core)	Hosting del controller Ansible implementato in laboratori on-premise
PostgreSQL	Versione 14.5	L'automazione estrae l'ultima versione disponibile di PostgreSQL dal postgresql.org yum repo
Ansible	Versione 2.10.3	Prerequisiti per le raccolte e le librerie richieste installate con il manuale dei requisiti

#### Fattori chiave per l'implementazione

- **Backup, ripristino e ripristino del database PostgreSQL.** Un database PostgreSQL supporta una serie di metodi di backup, ad esempio un backup logico con `pg_dump`, un backup fisico online con `pg_basebackup` o un comando di backup del sistema operativo di livello inferiore e snapshot coerenti a livello di storage. Questa soluzione utilizza le snapshot dei gruppi di coerenza NetApp per i dati del database PostgreSQL e il backup, ripristino e ripristino dei volumi WAL nel sito di standby. Le snapshot dei volumi del gruppo di coerenza NetApp sequenziali i/o man mano che vengono scritte nello storage e proteggono l'integrità dei file di dati del database.
- **Istanze di calcolo EC2.** In questi test e convalide, abbiamo utilizzato il tipo di istanza AWS EC2 t2.xlarge per l'istanza di calcolo del database PostgreSQL. NetApp consiglia di utilizzare un'istanza EC2 di tipo M5 come istanza di calcolo per PostgreSQL in fase di implementazione, poiché è ottimizzata per i carichi di

lavoro del database. L'istanza di calcolo in standby deve sempre essere implementata nella stessa zona del file system passivo (standby) implementato per il cluster FSX ha.

- **Implementazione di cluster ha storage FSX a singola o multi-zona.** in questi test e convalide, abbiamo implementato un cluster ha FSX in una singola zona di disponibilità AWS. Per l'implementazione in produzione, NetApp consiglia di implementare una coppia FSX ha in due diverse zone di disponibilità. Una coppia ha in standby per il disaster recovery per la business continuity può essere impostata in una regione diversa se è richiesta una distanza specifica tra il primario e lo standby. Un cluster FSX ha viene fornito in maniera ininterrotta in una coppia ha con mirroring sincronizzato in una coppia di file system Active-passive per fornire ridondanza a livello di storage.
- **Posizionamento dei dati e dei log di PostgreSQL.** le implementazioni tipiche di PostgreSQL condividono la stessa directory principale o volumi per i file di dati e di log. Nei test e nelle convalide, abbiamo separato i dati PostgreSQL e i log in due volumi separati per le performance. Nella directory dei dati viene utilizzato un soft link per indicare la directory di log o il volume che ospita i log WAL di PostgreSQL e i log WAL archiviati.
- **Timer di ritardo all'avvio del servizio PostgreSQL.** questa soluzione utilizza volumi montati su NFS per memorizzare il file di database PostgreSQL e i file di log WAL. Durante il riavvio di un host di database, il servizio PostgreSQL potrebbe tentare di avviarsi mentre il volume non è montato. Ciò comporta un errore di avvio del servizio di database. Per un corretto avvio del database PostgreSQL, è necessario un timer di 10 - 15 secondi.
- **RPO/RTO per la business continuity.** la replica dei dati FSX dal primario allo standby per il DR si basa su ASYNC, il che significa che l'RPO dipende dalla frequenza dei backup Snapshot e della replica SnapMirror. Una maggiore frequenza di copia Snapshot e replica SnapMirror riduce l'RPO. Di conseguenza, esiste un equilibrio tra la potenziale perdita di dati in caso di disastro e il costo incrementale dello storage. Abbiamo stabilito che la copia Snapshot e la replica SnapMirror possono essere implementate a intervalli di soli 5 minuti per RPO, mentre PostgreSQL può in genere essere ripristinato nel sito di standby del DR in meno di un minuto per RTO.
- **Backup del database.** dopo l'implementazione o la migrazione di un database PostgreSQL nello storage AWS FSX da un data center on-premises, i dati vengono sottoposti a mirroring con sincronizzazione automatica nella coppia FSX ha per la protezione. I dati vengono ulteriormente protetti con un sito di standby replicato in caso di disastro. Per la conservazione a lungo termine del backup o la protezione dei dati, NetApp consiglia di utilizzare l'utilità PostgreSQL pg\_basebackup integrata per eseguire un backup completo del database che può essere trasferito sullo storage BLOB S3.

## Implementazione della soluzione

L'implementazione di questa soluzione può essere completata automaticamente utilizzando il toolkit di automazione basato su Ansible di NetApp seguendo le istruzioni dettagliate riportate di seguito.

1. Leggere le istruzioni del toolkit di automazione readme.MD "[na\\_postgresql\\_aws\\_deploy\\_hadr](#)".
2. Guarda il video seguente.

### Installazione e protezione automatica di PostgreSQL

1. Configurare i file dei parametri richiesti (hosts, host\_vars/host\_name.yml, fsx\_vars.yml) immettendo i parametri specifici dell'utente nel modello nelle relative sezioni. Quindi, utilizzare il pulsante Copy per copiare i file sull'host del controller Ansible.

### Prerequisiti per l'implementazione automatica

L'implementazione richiede i seguenti prerequisiti.

1. È stato impostato un account AWS e sono stati creati i segmenti VPC e di rete necessari all'interno dell'account AWS.
2. Dalla console AWS EC2, è necessario implementare due istanze EC2 Linux, una come server primario PostgreSQL DB sul primario e una sul sito di DR di standby. Per la ridondanza di calcolo nei siti DR primari e di standby, implementare due istanze EC2 Linux aggiuntive come server PostgreSQL DB in standby. Per ulteriori informazioni sulla configurazione dell'ambiente, vedere il diagramma dell'architettura nella sezione precedente. Esaminare anche il ["Guida utente per istanze Linux"](#) per ulteriori informazioni.
3. Dalla console AWS EC2, implementare due cluster ha di storage FSX ONTAP per ospitare i volumi di database PostgreSQL. Se non si ha familiarità con l'implementazione dello storage FSX, consultare la documentazione ["Creazione di FSX per file system ONTAP"](#) per istruzioni dettagliate.
4. Creare una macchina virtuale CentOS Linux per ospitare il controller Ansible. Il controller Ansible può essere collocato on-premise o nel cloud AWS. Se si trova on-premise, è necessario disporre della connettività SSH per VPC, istanze EC2 Linux e cluster di storage FSX.
5. Impostare il controller Ansible come descritto nella sezione "impostazione del nodo di controllo Ansible per le implementazioni CLI su RHEL/CentOS" dalla risorsa ["Introduzione all'automazione delle soluzioni NetApp"](#).
6. Clonare una copia del toolkit di automazione dal sito pubblico NetApp GitHub.

```
git clone https://github.com/NetApp-
Automation/na_postgresql_aws_deploy_hadr.git
```

1. Dalla directory root del toolkit, eseguire i playbook prerequisiti per installare le raccolte e le librerie richieste per il controller Ansible.

```
ansible-playbook -i hosts requirements.yml
```

```
ansible-galaxy collection install -r collections/requirements.yml --force
--force-with-deps
```

1. Recuperare i parametri dell'istanza EC2 FSX richiesti per il file di variabili host DB `host_vars/*` e il file delle variabili globali `fsx_vars.yml` configurazione.

## Configurare il file hosts

Inserire i nomi host delle istanze primaria di FSX ONTAP per la gestione del cluster e EC2 nel file hosts.

```
# Primary FSx cluster management IP address
[fsx_ontap]
172.30.15.33
```

```
# Primary PostgreSQL DB server at primary site where database is
initialized at deployment time
[postgresql]
psql_01p ansible_ssh_private_key_file=psql_01p.pem
```

```
# Primary PostgreSQL DB server at standby site where postgresql service is
installed but disabled at deployment
# Standby DB server at primary site, to setup this server comment out
other servers in [dr_postgresql]
# Standby DB server at standby site, to setup this server comment out
other servers in [dr_postgresql]
[dr_postgresql] --
psql_01s ansible_ssh_private_key_file=psql_01s.pem
#psql_01ps ansible_ssh_private_key_file=psql_01ps.pem
#psql_01ss ansible_ssh_private_key_file=psql_01ss.pem
```

**Configurare il file `host_name.yml` nella cartella `host_vars`**

```
# Add your AWS EC2 instance IP address for the respective PostgreSQL
server host
ansible_host: "10.61.180.15"

# "{{groups.postgresql[0]}}" represents first PostgreSQL DB server as
defined in PostgreSQL hosts group [postgresql]. For concurrent multiple
PostgreSQL DB servers deployment, [0] will be incremented for each
additional DB server. For example, "{{groups.postgresql[1]}}" represents
DB server 2, "{{groups.postgresql[2]}}" represents DB server 3 ... As a
good practice and the default, two volumes are allocated to a PostgreSQL
DB server with corresponding /pgdata, /pglogs mount points, which store
PostgreSQL data, and PostgreSQL log files respectively. The number and
naming of DB volumes allocated to a DB server must match with what is
defined in global fsx_vars.yml file by src_db_vols, src_archivelog_vols
parameters, which dictates how many volumes are to be created for each DB
server. aggr_name is aggr1 by default. Do not change. lif address is the
NFS IP address for the SVM where PostgreSQL server is expected to mount
its database volumes. Primary site servers from primary SVM and standby
servers from standby SVM.
host_datastores_nfs:
  - {vol_name: "{{groups.postgresql[0]}}_pgdata", aggr_name: "aggr1", lif:
    "172.21.94.200", size: "100"}
  - {vol_name: "{{groups.postgresql[0]}}_pglogs", aggr_name: "aggr1", lif:
    "172.21.94.200", size: "100"}

# Add swap space to EC2 instance, that is equal to size of RAM up to 16G
max. Determine the number of blocks by dividing swap size in MB by 128.
swap_blocks: "128"

# Postgresql user configurable parameters
psql_port: "5432"
buffer_cache: "8192MB"
archive_mode: "on"
max_wal_size: "5GB"
client_address: "172.30.15.0/24"
```

### Configurare il file globale fsx\_vars.yml nella cartella vars

```
#####
##### PostgreSQL HADR global user configuration variables #####
##### Consolidate all variables from FSx, Linux, and postgresql #####
#####

#####
### Ontap env specific config variables ###
```



```
#####

#####
#####
# Variables for SnapMirror Peering
#####
#####

#Passphrase for cluster peering authentication
passphrase: "xxxxxxx"

#Please enter destination or standby FSx cluster name
dst_cluster_name: "FsxId0cf8e0bccb14805e8"

#Please enter destination or standby FSx cluster management IP
dst_cluster_ip: "172.30.15.90"

#Please enter destination or standby FSx cluster inter-cluster IP
dst_inter_ip: "172.30.15.13"

#Please enter destination or standby SVM name to create mirror
relationship
dst_vserver: "dr"

#Please enter destination or standby SVM management IP
dst_vserver_mgmt_lif: "172.30.15.88"

#Please enter destination or standby SVM NFS lif
dst_nfs_lif: "172.30.15.88"

#Please enter source or primary FSx cluster name
src_cluster_name: "FsxId0cf8e0bccb14805e8"

#Please enter source or primary FSx cluster management IP
src_cluster_ip: "172.30.15.20"

#Please enter source or primary FSx cluster inter-cluster IP
src_inter_ip: "172.30.15.5"

#Please enter source or primary SVM name to create mirror relationship
src_vserver: "prod"

#Please enter source or primary SVM management IP
src_vserver_mgmt_lif: "172.30.15.115"

#####
#####
```

```

# Variable for PostgreSQL Volumes, lif - source or primary FSx NFS lif
address
#####

src_db_vols:
  - {vol_name: "{{groups.postgresql[0]}}_pgdata", aggr_name: "aggr1", lif:
    "172.21.94.200", size: "100"}

src_archivelog_vols:
  - {vol_name: "{{groups.postgresql[0]}}_pglogs", aggr_name: "aggr1", lif:
    "172.21.94.200", size: "100"}

#Names of the Nodes in the ONTAP Cluster
nfs_export_policy: "default"

#####
### Linux env specific config variables ###
#####

#NFS Mount points for PostgreSQL DB volumes
mount_points:
  - "/pgdata"
  - "/pglogs"

#RedHat subscription username and password
redhat_sub_username: "xxxxx"
redhat_sub_password: "xxxxx"

#####
### DB env specific install and config variables ###
#####
#The latest version of PostgreSQL RPM is pulled/installed and config file
is deployed from a preconfigured template
#Recovery type and point: default as all logs and promote and leave all
PITR parameters blank

```

## Implementazione PostgreSQL e configurazione ha/DR

Le seguenti attività implementano il servizio del server DB PostgreSQL e inizializzano il database nel sito primario sull'host del server DB EC2 primario. Un host del server DB EC2 primario in standby viene quindi configurato nel sito di standby. Infine, la replica del volume DB viene configurata dal cluster FSX del sito primario al cluster FSX del sito di standby per il disaster recovery.

1. Creare volumi DB sul cluster FSX primario e impostare postgresql sull'host dell'istanza EC2 primario.

```
ansible-playbook -i hosts postgresql_deploy.yml -u ec2-user --private-key psql_01p.pem -e @vars/fsx_vars.yml
```

2. Impostare l'host di istanza EC2 DR di standby.

```
ansible-playbook -i hosts postgresql_standby_setup.yml -u ec2-user --private-key psql_01s.pem -e @vars/fsx_vars.yml
```

3. Impostare il peering del cluster FSX ONTAP e la replica del volume del database.

```
ansible-playbook -i hosts fsx_replication_setup.yml -e @vars/fsx_vars.yml
```

4. Consolida i passaggi precedenti in un'implementazione PostgreSQL e un'installazione ha/DR in un'unica fase.

```
ansible-playbook -i hosts postgresql_hadr_setup.yml -u ec2-user -e @vars/fsx_vars.yml
```

5. Per configurare un host PostgreSQL DB di standby sul sito primario o in standby, commentare tutti gli altri server nella sezione del file hosts [dr\_postgresql] ed eseguire il playbook postgresql\_standby\_setup.yml con il rispettivo host di destinazione (come ad esempio psql\_01ps o istanza di calcolo EC2 di standby sul sito primario). Assicurarsi che un file di parametri host, ad esempio psql\_01ps.yml è configurato in host\_vars directory.

```
[dr_postgresql] --  
#psql_01s ansible_ssh_private_key_file=psql_01s.pem  
psql_01ps ansible_ssh_private_key_file=psql_01ps.pem  
#psql_01ss ansible_ssh_private_key_file=psql_01ss.pem
```

```
ansible-playbook -i hosts postgresql_standby_setup.yml -u ec2-user --private-key psql_01ps.pem -e @vars/fsx_vars.yml
```

## Backup e replica dello snapshot del database PostgreSQL su un sito in standby

Il backup e la replica dello snapshot del database PostgreSQL nel sito di standby possono essere controllati ed eseguiti sul controller Ansible con un intervallo definito dall'utente. Abbiamo convalidato che l'intervallo può essere di soli 5 minuti. Pertanto, in caso di guasto nel sito primario, si verificano 5 minuti di potenziale perdita di dati se il guasto si verifica immediatamente prima del successivo backup di snapshot pianificato.

```
*/15 * * * * /home/admin/na_postgresql_aws_deploy_hadr/data_log_snap.sh
```

## Failover al sito di standby per DR

Per testare il sistema ha/DR PostgreSQL come esercizio di DR, eseguire il failover e il ripristino del database PostgreSQL sull'istanza primaria di standby EC2 DB sul sito di standby eseguendo il seguente manuale. In uno scenario di disaster recovery, eseguire lo stesso per un failover effettivo al sito di DR.

```
ansible-playbook -i hosts postgresql_failover.yml -u ec2-user --private-key psql_01s.pem -e @vars/fsx_vars.yml
```

## Risincronizzare volumi DB replicati dopo il test di failover

Eseguire la risincronizzazione dopo il test di failover per ristabilire la replica SnapMirror del volume di database.

```
ansible-playbook -i hosts postgresql_standby_resync.yml -u ec2-user --private-key psql_01s.pem -e @vars/fsx_vars.yml
```

## Failover dal server DB EC2 primario al server DB EC2 in standby a causa di un guasto dell'istanza di calcolo EC2

NetApp consiglia di eseguire il failover manuale o di utilizzare cluster-ware del sistema operativo ben consolidati che potrebbero richiedere una licenza.

## Dove trovare ulteriori informazioni

Per ulteriori informazioni sulle informazioni descritte in questo documento, consultare i seguenti documenti e/o siti Web:

- Amazon FSX per NetApp ONTAP

["https://aws.amazon.com/fsx/netapp-ontap/"](https://aws.amazon.com/fsx/netapp-ontap/)

- Amazon EC2

[https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bcd9843&sc\\_channel=ps&s\\_kwid=AL!4422!3!467723097970!e!!g!!aws%20ec2&ef\\_id=Cj0KCQiA54KfBhCKARIsAJzSrdqWQrghn6I71jiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw\\_wcB:G:s&s\\_kwid=AL!4422!3!467723097970!e!!g!!aws%20ec2](https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bcd9843&sc_channel=ps&s_kwid=AL!4422!3!467723097970!e!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARIsAJzSrdqWQrghn6I71jiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwid=AL!4422!3!467723097970!e!!g!!aws%20ec2)

- Automazione delle soluzioni NetApp

["https://docs.netapp.com/us-en/netapp-solutions/automation/automation\\_introduction.html"](https://docs.netapp.com/us-en/netapp-solutions/automation/automation_introduction.html)

## TR-4722: Database MySQL su Best practice NetApp ONTAP

Anup Bharti, Manohar Kulkarni, Jeffrey Steiner NetApp

MySQL e le sue varianti, tra cui MariaDB e Percona, sono ampiamente utilizzate per molte applicazioni aziendali. Variano da siti di social networking a livello globale e grandi sistemi di e-commerce a sistemi di hosting per PMI contenenti migliaia di istanze di database. Questo documento descrive i requisiti di configurazione e fornisce istruzioni sull'ottimizzazione e la configurazione dello storage per la distribuzione di MySQL sul software di gestione dati NetApp® ONTAP®. Per determinare se l'ambiente, le configurazioni e le versioni specificate in questo report supportano il tuo ambiente, consulta l'Interoperability Matrix Tool (IMT).

["TR-4722: Database MySQL su Best practice NetApp ONTAP"](#)

## Informazioni sul copyright

Copyright © 2024 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 COMMERCIALIZZABILITÀ 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.