Gestisci Trident Protect

Trident

NetApp
July 01, 2026

This PDF was generated from https://docs.netapp.com/it-it/trident-2602/trident-protect/manage-authorization-access-control.html on July 01, 2026. Always check docs.netapp.com for the latest.

Sommario

Gestisci Trident Protect

Gestisci l'autorizzazione e il controllo degli accessi di Trident Protect

Trident Protect utilizza il modello Kubernetes di controllo degli accessi in base al ruolo (RBAC). Per impostazione predefinita, Trident Protect fornisce un singolo namespace di sistema e il relativo account di servizio predefinito. Se hai un'organizzazione con molti utenti o esigenze di sicurezza specifiche, puoi utilizzare le funzionalità RBAC di Trident Protect per ottenere un controllo più granulare sull'accesso a risorse e namespace.

L'amministratore del cluster ha sempre accesso alle risorse nello spazio dei nomi predefinito `trident-protect` e può accedere anche alle risorse in tutti gli altri spazi dei nomi. Per controllare l'accesso alle risorse e alle applicazioni, è necessario creare spazi dei nomi aggiuntivi e aggiungere risorse e applicazioni a tali spazi dei nomi.

Si noti che nessun utente può creare CR di gestione dei dati applicativi nello spazio dei nomi predefinito `trident-protect`. È necessario creare CR di gestione dei dati applicativi in uno spazio dei nomi dell'applicazione (come best practice, creare CR di gestione dei dati applicativi nello stesso spazio dei nomi dell'applicazione associata).

Solo gli amministratori devono avere accesso agli oggetti risorsa personalizzati privilegiati di Trident Protect, che includono:

- **AppVault**: Richiede i dati delle credenziali del bucket
- **AutoSupportBundle**: Raccoglie metriche, log e altri dati sensibili di Trident Protect
- **AutoSupportBundleSchedule**: Gestisce le pianificazioni di raccolta dei log

Come best practice, utilizzare RBAC per limitare l'accesso agli oggetti privilegiati agli amministratori.

Per ulteriori informazioni su come RBAC regola l'accesso alle risorse e ai namespace, consultare la "Documentazione Kubernetes RBAC".

Per informazioni sugli account di servizio, consultare il "Documentazione dell'account di servizio Kubernetes".

Esempio: Gestisci l'accesso per due gruppi di utenti

Ad esempio, un'organizzazione ha un amministratore del cluster, un gruppo di utenti di ingegneria e un gruppo di utenti di marketing. L'amministratore del cluster completerà le seguenti operazioni per creare un ambiente in cui il gruppo di ingegneria e il gruppo di marketing abbiano accesso solo alle risorse assegnate ai rispettivi spazi dei nomi.

Passaggio 1: crea uno spazio dei nomi per contenere le risorse per ciascun gruppo

La creazione di un namespace consente di separare logicamente le risorse e di controllare meglio chi ha accesso a tali risorse.

Passaggi

1. Crea uno spazio dei nomi per il gruppo di ingegneria:

```
kubectl create ns engineering-ns
```

2. Crea uno spazio dei nomi per il gruppo marketing:

```
kubectl create ns marketing-ns
```

Passaggio 2: crea nuovi account di servizio per interagire con le risorse in ogni namespace

Ogni nuovo namespace che crei è dotato di un account di servizio predefinito, ma dovresti creare un account di servizio per ogni gruppo di utenti così da poter suddividere ulteriormente i privilegi tra i gruppi in futuro, se necessario.

Passaggi

1. Crea un account di servizio per il gruppo di ingegneria:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: eng-user
  namespace: engineering-ns
```

2. Crea un account di servizio per il gruppo marketing:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: mkt-user
  namespace: marketing-ns
```

Passaggio 3: crea un segreto per ogni nuovo account di servizio

Un secret dell'account di servizio viene utilizzato per autenticarsi con l'account di servizio e può essere facilmente eliminato e ricreato se compromesso.

Passaggi

1. Crea un segreto per l'account del servizio di ingegneria:

```
apiVersion: v1
kind: Secret
metadata:
  annotations:
    kubernetes.io/service-account.name: eng-user
  name: eng-user-secret
  namespace: engineering-ns
type: kubernetes.io/service-account-token
```

2. Crea un segreto per l'account del servizio marketing:

```
apiVersion: v1
kind: Secret
metadata:
  annotations:
    kubernetes.io/service-account.name: mkt-user
  name: mkt-user-secret
  namespace: marketing-ns
type: kubernetes.io/service-account-token
```

Passaggio 4: crea un oggetto RoleBinding per associare l'oggetto ClusterRole a ciascun nuovo account di servizio

Un oggetto ClusterRole predefinito viene creato quando si installa Trident Protect. È possibile associare questo ClusterRole all'account di servizio creando e applicando un oggetto RoleBinding.

Passaggi

1. Associa il ClusterRole all'account del servizio di ingegneria:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: engineering-ns-tenant-rolebinding
  namespace: engineering-ns
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: trident-protect-tenant-cluster-role
subjects:
- kind: ServiceAccount
  name: eng-user
  namespace: engineering-ns
```

2. Associa il ClusterRole all'account del servizio marketing:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: marketing-ns-tenant-rolebinding
  namespace: marketing-ns
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: trident-protect-tenant-cluster-role
subjects:
- kind: ServiceAccount
  name: mkt-user
  namespace: marketing-ns
```

Passaggio 5: testare le autorizzazioni

Verificare che le autorizzazioni siano corrette.

Passaggi

1. Confermare che gli utenti di ingegneria possano accedere alle risorse di ingegneria:

```
kubectl auth can-i --as=system:serviceaccount:engineering-ns:eng-user
get applications.protect.trident.netapp.io -n engineering-ns
```

2. Confermare che gli utenti di ingegneria non possano accedere alle risorse di marketing:

```
kubectl auth can-i --as=system:serviceaccount:engineering-ns:eng-user
get applications.protect.trident.netapp.io -n marketing-ns
```

Passaggio 6: Concedi l'accesso agli oggetti AppVault

Per eseguire attività di gestione dei dati quali backup e snapshot, l'amministratore del cluster deve concedere l'accesso agli oggetti AppVault ai singoli utenti.

Passaggi

1. Crea e applica un file YAML di combinazione tra AppVault e secret che garantisce a un utente l'accesso a un AppVault. Ad esempio, la seguente CR concede l'accesso a un AppVault all'utente `eng-user`:

```
apiVersion: v1
data:
  accessKeyID: <ID_value>
  secretAccessKey: <key_value>
kind: Secret
metadata:
  name: appvault-for-eng-user-only-secret
  namespace: trident-protect
type: Opaque
---
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: appvault-for-eng-user-only
  namespace: trident-protect # Trident Protect system namespace
spec:
  providerConfig:
    azure:
      accountName: ""
      bucketName: ""
      endpoint: ""
    gcp:
      bucketName: ""
      projectID: ""
    s3:
      bucketName: testbucket
      endpoint: 192.168.0.1:30000
      secure: "false"
      skipCertValidation: "true"
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        key: accessKeyID
        name: appvault-for-eng-user-only-secret
    secretAccessKey:
      valueFromSecret:
        key: secretAccessKey
        name: appvault-for-eng-user-only-secret
  providerType: GenericS3
```

2. Crea e applica un Role CR per consentire agli amministratori del cluster di concedere l'accesso a risorse specifiche in uno spazio dei nomi. Ad esempio:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: eng-user-appvault-reader
  namespace: trident-protect
rules:
- apiGroups:
  - protect.trident.netapp.io
  resourceNames:
  - appvault-for-enguser-only
  resources:
  - appvaults
  verbs:
  - get
```

3. Crea e applica una RoleBinding CR per associare i permessi all'utente eng-user. Ad esempio:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: eng-user-read-appvault-binding
  namespace: trident-protect
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: eng-user-appvault-reader
subjects:
- kind: ServiceAccount
  name: eng-user
  namespace: engineering-ns
```

4. Verificare che le autorizzazioni siano corrette.

   a. Tentativo di recuperare le informazioni sugli oggetti AppVault per tutti gli spazi dei nomi:

   ```
   kubectl get appvaults -n trident-protect
   --as=system:serviceaccount:engineering-ns:eng-user
   ```

   Dovresti vedere un output simile al seguente:

```
Error from server (Forbidden): appvaults.protect.trident.netapp.io is
forbidden: User "system:serviceaccount:engineering-ns:eng-user"
cannot list resource "appvaults" in API group
"protect.trident.netapp.io" in the namespace "trident-protect"
```

b. Verificare se l'utente riesce a ottenere le informazioni AppVault a cui ora ha il permesso di accedere:

```
kubectl auth can-i --as=system:serviceaccount:engineering-ns:eng-user
get appvaults.protect.trident.netapp.io/appvault-for-eng-user-only -n
trident-protect
```

Dovresti vedere un output simile al seguente:

```
yes
```

Risultato

Gli utenti a cui hai concesso le autorizzazioni AppVault devono essere in grado di utilizzare oggetti AppVault autorizzati per le operazioni di gestione dei dati dell'applicazione e non devono essere in grado di accedere a nessuna risorsa al di fuori degli spazi dei nomi assegnati o di creare nuove risorse a cui non hanno accesso.

Monitorare le risorse di Trident Protect

È possibile utilizzare gli strumenti open source kube-state-metrics, Prometheus e Alertmanager per monitorare lo stato di salute delle risorse protette da Trident Protect.

Il servizio kube-state-metrics genera metriche dalla comunicazione API di Kubernetes. Utilizzarlo con Trident Protect espone informazioni utili sullo stato delle risorse nel tuo ambiente.

Prometheus è un toolkit in grado di acquisire i dati generati da kube-state-metrics e presentarli come informazioni facilmente leggibili su questi oggetti. Insieme, kube-state-metrics e Prometheus offrono un modo per monitorare l'integrità e lo stato delle risorse che gestisci con Trident Protect.

Alertmanager è un servizio che acquisisce gli avvisi inviati da strumenti come Prometheus e li instrada verso destinazioni che configuri.

Le configurazioni e le istruzioni incluse in questi passaggi sono solo esempi; è necessario personalizzarle per adattarle al proprio ambiente. Fare riferimento alla seguente documentazione ufficiale per istruzioni e supporto specifici:

- "documentazione kube-state-metrics"
- "Documentazione di Prometheus"
- "Documentazione di Alertmanager"

Passaggio 1: installa gli strumenti di monitoraggio

Per abilitare il monitoraggio delle risorse in Trident Protect, è necessario installare e configurare kube-state-metrics, Prometheus e Alertmanager.

Installa kube-state-metrics

Puoi installare kube-state-metrics usando Helm.

Passaggi

1. Aggiungi il grafico Helm kube-state-metrics. Ad esempio:

```
helm repo add prometheus-community https://prometheus-
community.github.io/helm-charts
helm repo update
```

2. Applica il CRD ServiceMonitor di Prometheus al cluster:

```
kubectl apply -f https://raw.githubusercontent.com/prometheus-
operator/prometheus-operator/main/example/prometheus-operator-
crd/monitoring.coreos.com_servicemonitors.yaml
```

3. Crea un file di configurazione per il grafico Helm (ad esempio, `metrics-config.yaml`). Puoi personalizzare la seguente configurazione di esempio in base al tuo ambiente:

metrics-config.yaml: configurazione Helm chart kube-state-metrics

```
---
extraArgs:
  # Collect only custom metrics
  - --custom-resource-state-only=true

customResourceState:
  enabled: true
  config:
    kind: CustomResourceStateMetrics
    spec:
      resources:
      - groupVersionKind:
          group: protect.trident.netapp.io
          kind: "Backup"
          version: "v1"
        labelsFromPath:
          backup_uid: [metadata, uid]
          backup_name: [metadata, name]
          creation_time: [metadata, creationTimestamp]
        metrics:
        - name: backup_info
          help: "Exposes details about the Backup state"
          each:
            type: Info
            info:
              labelsFromPath:
                appVaultReference: ["spec", "appVaultRef"]
                appReference: ["spec", "applicationRef"]
rbac:
  extraRules:
  - apiGroups: ["protect.trident.netapp.io"]
    resources: ["backups"]
    verbs: ["list", "watch"]

# Collect metrics from all namespaces
namespaces: ""

# Ensure that the metrics are collected by Prometheus
prometheus:
  monitor:
    enabled: true
```

4. Installa kube-state-metrics distribuendo il grafico Helm. Ad esempio:

```
helm install custom-resource -f metrics-config.yaml prometheus-
community/kube-state-metrics --version 5.21.0
```

5. Configura kube-state-metrics per generare metriche per le risorse personalizzate utilizzate da Trident Protect seguendo le istruzioni nel "documentazione delle risorse personalizzate kube-state-metrics".

Installa Prometheus

È possibile installare Prometheus seguendo le istruzioni nel "Documentazione di Prometheus".

Installa Alertmanager

È possibile installare Alertmanager seguendo le istruzioni nel "Documentazione di Alertmanager".

Passaggio 2: configura gli strumenti di monitoraggio affinché funzionino insieme

Dopo aver installato gli strumenti di monitoraggio, è necessario configurarli affinché funzionino insieme.

Passaggi

1. Integra kube-state-metrics con Prometheus. Modifica il file di configurazione Prometheus (`prometheus.yaml` e aggiungi le informazioni sul servizio kube-state-metrics. Ad esempio:

   prometheus.yaml: integrazione del servizio kube-state-metrics con Prometheus

```
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: prometheus-config
  namespace: trident-protect
data:
  prometheus.yaml: |
    global:
      scrape_interval: 15s
    scrape_configs:
      - job_name: 'kube-state-metrics'
        static_configs:
          - targets: ['kube-state-metrics.trident-protect.svc:8080']
```

2. Configura Prometheus per indirizzare gli avvisi ad Alertmanager. Modifica il file di configurazione di Prometheus (`prometheus.yaml` e aggiungi la seguente sezione:

prometheus.yaml: Invia avvisi ad Alertmanager

```
alerting:
  alertmanagers:
    - static_configs:
        - targets:
            - alertmanager.trident-protect.svc:9093
```

Risultato

Prometheus ora può raccogliere metriche da kube-state-metrics e inviare avvisi ad Alertmanager. Ora sei pronto per configurare quali condizioni attivano un avviso e dove devono essere inviati gli avvisi.

Passaggio 3: Configura gli avvisi e le destinazioni degli avvisi

Dopo aver configurato gli strumenti affinché funzionino insieme, è necessario configurare quale tipo di informazioni attiva gli avvisi e dove devono essere inviati.

Esempio di avviso: errore di backup

L'esempio seguente definisce un avviso critico che viene attivato quando lo stato della risorsa personalizzata di backup è impostato su `Error` per 5 secondi o più. È possibile personalizzare questo esempio in base al proprio ambiente e includere questo frammento YAML nel file di configurazione `prometheus.yaml`:

rules.yaml: Definisci un avviso Prometheus per i backup non riusciti

```
rules.yaml: |
  groups:
    - name: fail-backup
        rules:
          - alert: BackupFailed
            expr: kube_customresource_backup_info{status="Error"}
            for: 5s
            labels:
              severity: critical
            annotations:
              summary: "Backup failed"
              description: "A backup has failed."
```

Configura Alertmanager per inviare avvisi ad altri canali

È possibile configurare Alertmanager per inviare notifiche ad altri canali, come e-mail, PagerDuty, Microsoft Teams o altri servizi di notifica, specificando la rispettiva configurazione nel `alertmanager.yaml` file.

L'esempio seguente configura Alertmanager per inviare notifiche a un canale Slack. Per personalizzare questo esempio in base al tuo ambiente, sostituisci il valore della `api_url` chiave con l'URL del webhook Slack utilizzato nel tuo ambiente:

alertmanager.yaml: invia avvisi a un canale Slack

```
data:
  alertmanager.yaml: |
    global:
      resolve_timeout: 5m
    route:
      receiver: 'slack-notifications'
    receivers:
      - name: 'slack-notifications'
        slack_configs:
          - api_url: '<your-slack-webhook-url>'
            channel: '#failed-backups-channel'
            send_resolved: false
```

Generare un bundle di supporto Trident Protect

Trident Protect consente agli amministratori di generare bundle che includono informazioni utili a NetApp Support, tra cui log, metriche e informazioni sulla topologia dei cluster e delle app gestite. Se si è connessi a Internet, è possibile caricare i bundle di supporto sul NetApp Support Site (NSS) utilizzando un file custom resource (CR).

Crea un bundle di supporto utilizzando un CR

Passaggi

1. Creare il file custom resource (CR) e assegnargli un nome (ad esempio `trident-protect-support-bundle.yaml`).
2. Configura i seguenti attributi:
   - **metadata.name**: (*Obbligatorio*) Il nome di questa risorsa personalizzata; scegli un nome univoco e sensato per il tuo ambiente.
   - **spec.triggerType**: (*Required*) Determina se il bundle di supporto viene generato immediatamente o programmato. La generazione del bundle programmato avviene alle 12AM UTC. Valori possibili:
     - In programma
     - Manuale
   - **spec.uploadEnabled**: (*Opzionale*) Controlla se il bundle di supporto deve essere caricato sul NetApp Support Site dopo essere stato generato. Se non specificato, il valore predefinito è `false`. Valori possibili:
     - true
     - false (predefinito)
   - **spec.dataWindowStart**: (*Opzionale*) Una stringa di data nel formato RFC 3339 che specifica la data e l'ora in cui deve iniziare la finestra dei dati inclusi nel support bundle. Se non specificato, il valore predefinito è 24 ore fa. La data più remota della finestra che si può specificare è 7 giorni fa.

     Esempio YAML:

```
---
apiVersion: protect.trident.netapp.io/v1
kind: AutoSupportBundle
metadata:
  name: trident-protect-support-bundle
spec:
  triggerType: Manual
  uploadEnabled: true
  dataWindowStart: 2024-05-05T12:30:00Z
```

3. Dopo aver popolato il `trident-protect-support-bundle.yaml` file con i valori corretti, applica la CR:

```
kubectl apply -f trident-protect-support-bundle.yaml -n trident-protect
```

Crea un bundle di supporto utilizzando la CLI

Passaggi

1. Crea il bundle di supporto, sostituendo i valori tra parentesi con le informazioni del tuo ambiente. Il `trigger-type` determina se il bundle viene creato immediatamente o se l'orario di creazione è

dettato dalla pianificazione, e può essere `Manual` o `Scheduled`. L'impostazione predefinita è `Manual`.

Ad esempio:

```
tridentctl-protect create autosupportbundle <my-bundle-name>
--trigger-type <trigger-type> -n trident-protect
```

Monitora e recupera il pacchetto di supporto

Dopo aver creato un support bundle con uno dei due metodi, puoi monitorare l'avanzamento della generazione e recuperarlo sul tuo sistema locale.

Passaggi

1. Attendere che `status.generationState` raggiunga lo stato `Completed`. È possibile monitorare l'avanzamento della generazione con il seguente comando:

   ```
   kubectl get autosupportbundle trident-protect-support-bundle -n trident-protect
   ```

2. Recupera il bundle di supporto nel sistema locale. Ottieni il comando di copia dal bundle completato AutoSupport:

   ```
   kubectl describe autosupportbundle trident-protect-support-bundle -n trident-protect
   ```

   Individua il comando `kubectl cp` dall'output ed eseguilo, sostituendo l'argomento destination con la tua directory locale preferita.

Aggiorna Trident Protect

Puoi aggiornare Trident Protect all'ultima versione per sfruttare le nuove funzionalità o correzioni di bug.

- Quando si esegue l'aggiornamento dalla versione 24.10, gli snapshot in esecuzione durante l'aggiornamento potrebbero non riuscire. Questo errore non impedisce la creazione di snapshot futuri, sia manuali che pianificati. Se uno snapshot non riesce durante l'aggiornamento, è possibile crearne uno nuovo manualmente per garantire che l'applicazione sia protetta.

  Per evitare potenziali errori, è possibile disabilitare tutte le pianificazioni degli Snapshot prima dell'aggiornamento e riabilitarle in seguito. Tuttavia, questo comporta la mancata acquisizione di eventuali Snapshot pianificati durante il periodo di aggiornamento.

- Per le installazioni con registro privato, assicurati che il chart e le immagini Helm richiesti per la versione di destinazione siano disponibili nel tuo registro privato e verifica che i valori Helm personalizzati siano compatibili con la nuova versione del chart. Per ulteriori informazioni, fai riferimento a "Installa Trident Protect da un registro privato".

Passaggio 1: seleziona una versione

Le versioni di Trident Protect seguono una convenzione di naming basata sulla data `YY.MM`, dove "YY" sono le ultime due cifre dell'anno e "MM" è il mese. Le versioni Dot seguono una convenzione `YY.MM.X`, dove "X" è il livello di patch. Selezionerai la versione a cui eseguire l'aggiornamento in base alla versione da cui stai aggiornando.

- È possibile eseguire un aggiornamento diretto a qualsiasi release di destinazione che rientri in una finestra di quattro release della versione installata. Ad esempio, è possibile eseguire un aggiornamento diretto dalla 24.10 (o da qualsiasi 24.10 dot release) alla 25.10.
- Se stai eseguendo l'aggiornamento da una versione al di fuori della finestra di quattro release, esegui un aggiornamento in più fasi. Usare le istruzioni di aggiornamento per la "versione precedente" da cui si esegue l'aggiornamento per passare alla release più recente che rientra nella finestra di quattro release. Ad esempio, se stai utilizzando la versione 24.10 e desideri eseguire l'aggiornamento alla versione 26.02:
  a. Primo upgrade dal 24.10 al 25.02.
  b. Quindi esegui l'aggiornamento da 25.02 a 26.02.

Passaggio 2: Aggiornare Trident Protect

Per aggiornare Trident Protect, eseguire i seguenti passaggi.

Passaggi

1. Aggiorna il repository Helm di Trident:

```
helm repo update
```

2. Aggiorna i CRD di Trident Protect:

   Questo passaggio è necessario se si esegue l'aggiornamento da una versione precedente alla 25.06, poiché i CRD sono ora inclusi nel chart Helm di Trident Protect.

   a. Esegui questo comando per spostare la gestione dei CRD da `trident-protect-crds` a `trident-protect`:

```
kubectl get crd | grep protect.trident.netapp.io | awk '{print $1}' |
xargs -I {} kubectl patch crd {} --type merge -p '{"metadata":
{"annotations":{"meta.helm.sh/release-name": "trident-protect"}}}'
```

b. Esegui questo comando per eliminare il segreto Helm per il `trident-protect-crds` chart:

> Non disinstallare il `trident-protect-crds` chart tramite Helm, poiché ciò potrebbe rimuovere i tuoi CRD e qualsiasi dato correlato.

```
kubectl delete secret -n trident-protect -l name=trident-protect-
crds,owner=helm
```

3. Aggiorna Trident Protect:

```
helm upgrade trident-protect netapp-trident-protect/trident-protect
--version 100.2602.0 --namespace trident-protect
```

> È possibile configurare il livello di registrazione durante l'aggiornamento aggiungendo `--set logLevel=debug` al comando di aggiornamento. Il livello di registrazione predefinito è `warn`. La registrazione di debug è consigliata per la risoluzione dei problemi, in quanto aiuta NetApp a diagnosticare i problemi senza richiedere modifiche al livello di registrazione o la riproduzione dei problemi.

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.