Instalar o Trident no cluster Red Hat OpenShift e criar objetos de armazenamento

NetApp virtualization solutions

NetApp
August 18, 2025

This PDF was generated from https://docs.netapp.com/pt-br/netapp-solutions-virtualization/openshift/osv-trident-install.html on August 18, 2025. Always check docs.netapp.com for the latest.

Índice

Instalar o Trident no cluster Red Hat OpenShift e criar objetos de armazenamento . . . 1
Demonstração em vídeo . . . 6
Configuração do Trident para cluster OpenShift local . . . 6
Configuração Trident para cluster ROSA usando armazenamento FSxN . . . 11
Criando uma classe de instantâneo de volume Trident . . . 12
Definindo padrões com Trident Storage e Snapshot Class . . . 13

Instalar o Trident no cluster Red Hat OpenShift e criar objetos de armazenamento

Instale o Trident usando o Red Hat Certified Trident Operator em clusters OpenShift e prepare os nós de trabalho para acesso em bloco. Crie objetos de classe de armazenamento e backend Trident para armazenamento ONTAP e FSxN para permitir o provisionamento dinâmico de volume para contêineres e VMs.

> Se você precisar criar VMs no OpenShift Virtualization, o Trident deverá ser instalado e os objetos de backend e os objetos de classe de armazenamento deverão ser criados no OpenShift Cluster antes que o OpenShift Virtualization seja instalado no cluster (local e ROSA). A classe de armazenamento padrão e a classe de snapshot de volume padrão devem ser definidas para o armazenamento Trident e a classe de snapshot no cluster. Somente quando isso estiver configurado, o OpenShift Virtualization poderá disponibilizar as imagens douradas localmente para criação de VM usando modelos.

> Se o operador OpenShift Virtualization estiver instalado antes da instalação do Trident, você poderá usar o comando a seguir para excluir as imagens douradas criadas usando uma classe de armazenamento diferente e, em seguida, deixar que o OpenShift Virtualization crie as imagens douradas usando a classe de armazenamento Trident , garantindo que os padrões das classes Trident Storage e Volume Snapshot estejam definidos.

```
oc delete dv,VolumeSnapshot -n openshift-virtualization-os-images
--selector=cdi.kubevirt.io/dataImportCron
```

> Para obter arquivos yaml de exemplo para criar objetos trident para armazenamento FSxN para clusters ROSA e para obter o arquivo yaml de exemplo para VolumeSnapshotClass, role para baixo nesta página.

Instalando o Trident

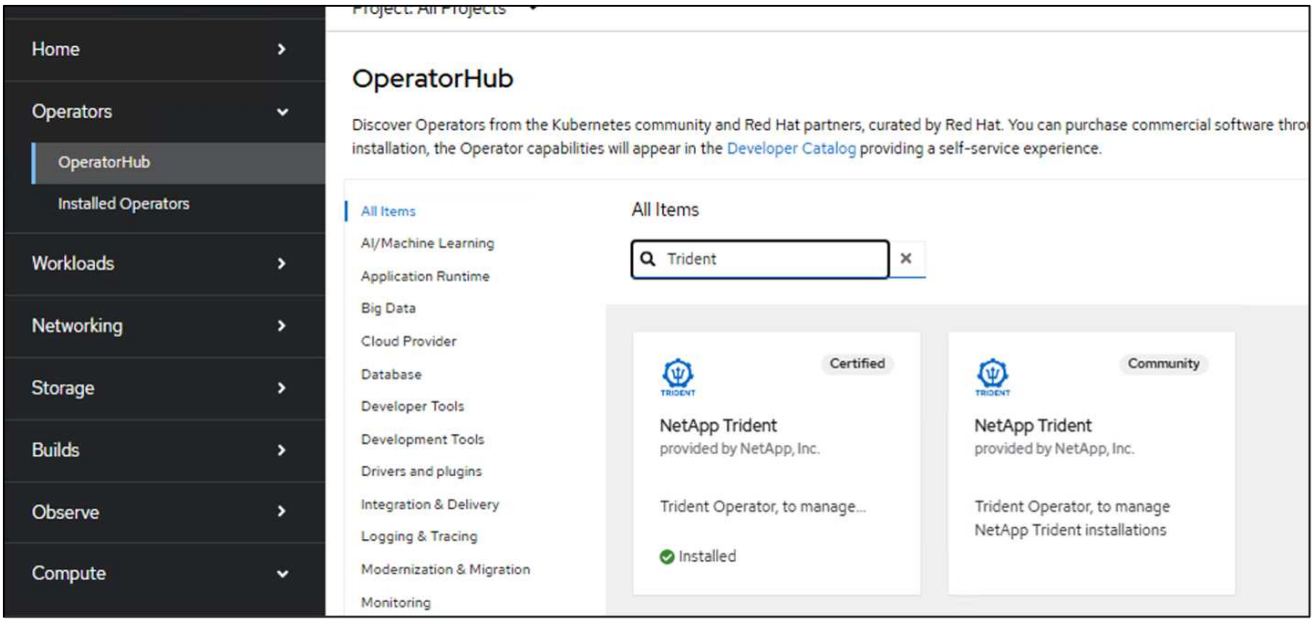
Instalando o Trident usando o Operador Certificado Red Hat

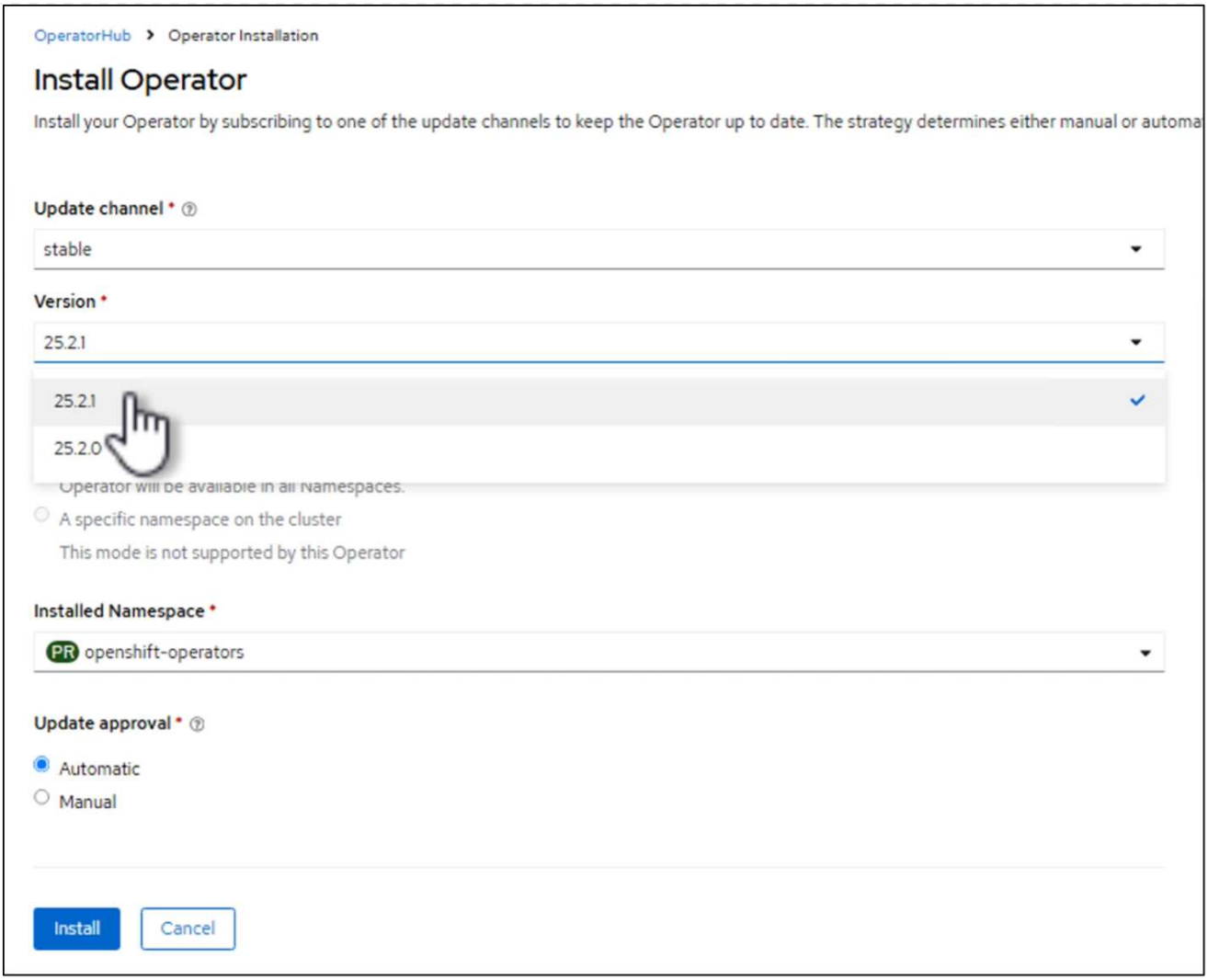
Nesta seção, são fornecidos detalhes sobre a instalação do Trident usando o Red Hat Certified Trident Operator"Consulte a documentação do Trident" para outras maneiras de instalar o Trident. Com o lançamento do Trident 25.02, os usuários do Trident no Red Hat OpenShift local e na nuvem e serviços gerenciados como o Red Hat OpenShift Service na AWS agora podem instalar o Trident usando o Trident Certified Operator no Operator Hub. Isso é significativo para a comunidade de usuários do OpenShift, já que o Trident estava disponível anteriormente apenas como um operador comunitário.

A vantagem do operador Red Hat Certified Trident é que a base para o operador e seus contêineres é totalmente suportada pela NetApp quando usada com o OpenShift (seja no local, na nuvem ou como um serviço gerenciado com o ROSA). Além disso, o NetApp Trident não tem custo para o cliente, então tudo o que você precisa fazer é instalá-lo usando o operador certificado que foi verificado para funcionar perfeitamente com o Red Hat OpenShift e empacotado para fácil gerenciamento do ciclo de vida.

Além disso, o operador Trident 25.02 (e versões futuras) oferece o benefício opcional de preparar os nós de trabalho para iSCSI. Isso é particularmente vantajoso se você planeja implantar suas cargas de trabalho em clusters ROSA e pretende usar o protocolo iSCSI com FSxN, especialmente para cargas de trabalho de VM do OpenShift Virtualization. O desafio dos preparativos de nós de trabalho para iSCSI em clusters ROSA usando FSxN foi atenuado com esse recurso ao instalar o Trident no cluster.

As etapas de instalação usando o operador são as mesmas, independentemente de você estar instalando em um cluster local ou no ROSA. Para instalar o Trident usando o Operador, clique no hub do Operador e selecione Certified NetApp Trident. Na página Instalar, a versão mais recente é selecionada por padrão. Clique em
Instalar.





Depois que o operador estiver instalado, clique em visualizar operador e crie uma instância do Trident Orchestrator. Se você quiser preparar os nós de trabalho para acesso ao armazenamento iSCSI, vá para a visualização yaml e modifique o parâmetro nodePrep adicionando iscsi.

Create TridentOrchestrator

Create by completing the form. Default values may be provided by the Operator authors.

Configure via: ○ Form view ◉ YAML view

```yaml
kind: TridentOrchestrator
apiVersion: trident.netapp.io/v1
metadata:
  name: trident
spec:
  IPv6: false
  debug: true
  nodePrep:
  - iscsi
  imagePullSecrets: []
  imageRegistry: ''
  namespace: trident
  silenceAutosupport: false
```

Agora você deve ter todos os pods Trident em execução no seu cluster.

```
[root@localhost ~]# oc get pods -n trident
NAME                                  READY   STATUS    RESTARTS   AGE
trident-controller-84cb9bff89-lkx6k   6/6     Running   0          16h
trident-node-linux-d88b9              2/2     Running   0          16h
trident-node-linux-ld4b8              2/2     Running   0          16h
trident-node-linux-mj5r8              2/2     Running   0          16h
trident-node-linux-mkmmp              2/2     Running   0          16h
trident-node-linux-qhgr7              2/2     Running   0          16h
trident-node-linux-vt9tp              2/2     Running   0          16h
[root@localhost ~]#
```

Para verificar se as ferramentas iSCSI foram habilitadas nos nós de trabalho do OpenShift Cluster, faça login nos nós de trabalho e verifique se você vê o iscsid, o multipathd ativo e as entradas no arquivo multipath.conf, conforme mostrado.

```
sh-5.1# systemctl status iscsid
● iscsid.service - Open-iSCSI
     Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-04-25 00:23:49 UTC; 3 days ago
TriggeredBy: ● iscsid.socket
       Docs: man:iscsid(8)
             man:iscsiuio(8)
             man:iscsiadm(8)
   Main PID: 74787 (iscsid)
     Status: "Ready to process requests"
      Tasks: 1 (limit: 410912)
     Memory: 1.8M
        CPU: 6ms
     CGroup: /system.slice/iscsid.service
             └─74787 /usr/sbin/iscsid -f

Apr 25 00:23:49 ocp11-worker1 systemd[1]: Starting Open-iSCSI...
Apr 25 00:23:49 ocp11-worker1 systemd[1]: Started Open-iSCSI.
sh-5.1#
```

```
sh-5.1# systemctl status multipathd
● multipathd.service - Device-Mapper Multipath Device Controller
     Loaded: loaded (/usr/lib/systemd/system/multipathd.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-04-25 00:23:50 UTC; 3 days ago
TriggeredBy: ● multipathd.socket
    Process: 74905 ExecStartPre=/sbin/modprobe -a scsi_dh_alua scsi_dh_emc scsi_dh_rdac dm-multipath (code=exited, status=0/SUCCESS)
    Process: 74906 ExecStartPre=/sbin/multipath -A (code=exited, status=0/SUCCESS)
   Main PID: 74907 (multipathd)
     Status: "up"
      Tasks: 7
     Memory: 18.3M
        CPU: 23.008s
     CGroup: /system.slice/multipathd.service
             └─74907 /sbin/multipathd -d -s

Apr 25 00:23:50 ocp11-worker1 systemd[1]: Starting Device-Mapper Multipath Device Controller...
Apr 25 00:23:50 ocp11-worker1 multipathd[74907]: --------start up--------
Apr 25 00:23:50 ocp11-worker1 multipathd[74907]: read /etc/multipath.conf
Apr 25 00:23:50 ocp11-worker1 multipathd[74907]: path checkers start up
Apr 25 00:23:50 ocp11-worker1 systemd[1]: Started Device-Mapper Multipath Device Controller.
sh-5.1#
```

```
sh-5.1# cat /etc/multipath.conf
defaults {
    find_multipaths no
}
blacklist {
    device {
        product .*
        vendor  .*
    }
}
blacklist_exceptions {
    device {
        product LUN
        vendor  NETAPP
    }
}
sh-5.1#
```

Demonstração em vídeo

O vídeo a seguir mostra uma demonstração da instalação do Trident usando o Red Hat Certified Trident Operator

Instalando o Trident 25.02.1 usando o Trident Operator certificado no OpenShift

Configuração do Trident para cluster OpenShift local

Classe de armazenamento e backend Trident para NAS

```
cat tbc-nas.yaml
apiVersion: v1
kind: Secret
metadata:
  name: tbc-nas-secret
type: Opaque
stringData:
  username: <cluster admin username>
  password: <cluster admin password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-nas
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: <cluster management lif>
  backendName: tbc-nas
  svm: zoneb
  storagePrefix: testzoneb
  defaults:
    nameTemplate: "{{ .config.StoragePrefix }}_{{ .volume.Namespace
}}_{{ .volume.RequestName }}"
  credentials:
    name: tbc-nas-secret
```

```
cat sc-nas.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-nas
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
allowVolumeExpansion: true
```

Classe de armazenamento e backend Trident para iSCSI

```yaml
# cat tbc-iscsi.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-iscsi-secret
type: Opaque
stringData:
  username: <cluster admin username>
  password: <cluster admin password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: ontap-iscsi
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <management LIF>
  backendName: ontap-iscsi
  svm: <SVM name>
  credentials:
    name: backend-tbc-ontap-iscsi-secret
```

```yaml
# cat sc-iscsi.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-iscsi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  fsType: ext4
  snapshots: "true"
allowVolumeExpansion: true
```

Classe de armazenamento e backend Trident para NVMe/TCP

```yaml
# cat tbc-nvme.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-nvme-secret
type: Opaque
stringData:
  username: <cluster admin password>
  password: <cluster admin password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nvme
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <cluster management LIF>
  backendName: backend-tbc-ontap-nvme
  svm: <SVM name>
  credentials:
    name: backend-tbc-ontap-nvme-secret
```

```yaml
# cat sc-nvme.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-nvme
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  fsType: ext4
  snapshots: "true"
allowVolumeExpansion: true
```

Classe de armazenamento e backend Trident para FC

```
# cat tbc-fc.yaml
apiVersion: v1
kind: Secret
metadata:
  name: tbc-fc-secret
type: Opaque
stringData:
  username: <cluster admin password>
  password: <cluster admin password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-fc
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <cluster mgmt lif>
  backendName: tbc-fc
  svm: openshift-fc
  sanType: fcp
  storagePrefix: demofc
  defaults:
    nameTemplate: "{{ .config.StoragePrefix }}_{{ .volume.Namespace
}}_{{ .volume.RequestName }}"
  credentials:
    name: tbc-fc-secret
```

```
# cat sc-fc.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-fc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  fsType: ext4
  snapshots: "true"
allowVolumeExpansion: true
```

Configuração Trident para cluster ROSA usando armazenamento FSxN

Classe de armazenamento e backend Trident para FSxN NAS

```
#cat tbc-fsx-nas.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-fsx-ontap-nas-secret
  namespace: trident
type: Opaque
stringData:
  username: <cluster admin lif>
  password: <cluster admin passwd>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-fsx-ontap-nas
  namespace: trident
spec:
  version: 1
  backendName: fsx-ontap
  storageDriverName: ontap-nas
  managementLIF: <Management DNS name>
  dataLIF: <NFS DNS name>
  svm: <SVM NAME>
  credentials:
    name: backend-fsx-ontap-nas-secret
```

```
# cat sc-fsx-nas.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: trident-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "ext4"
allowVolumeExpansion: True
reclaimPolicy: Retain
```

Classe de armazenamento e backend Trident para FSxN iSCSI

```
# cat tbc-fsx-iscsi.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-fsx-iscsi-secret
type: Opaque
stringData:
  username: <cluster admin username>
  password: <cluster admin password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: fsx-iscsi
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <management LIF>
  backendName: fsx-iscsi
  svm: <SVM name>
  credentials:
    name: backend-tbc-ontap-iscsi-secret
```

```
# cat sc-fsx-iscsi.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-fsx-iscsi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  fsType: ext4
  snapshots: "true"
allowVolumeExpansion: true
```

Criando uma classe de instantâneo de volume Trident

Classe de instantâneo de volume Trident

```
# cat snapshot-class.yaml
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: trident-snapshotclass
driver: csi.trident.netapp.io
deletionPolicy: Retain
```

Depois de ter os arquivos yaml necessários para a configuração do backend, a configuração da classe de armazenamento e as configurações de snapshot, você pode criar os objetos de backend, classe de armazenamento e classe de snapshot do Trident usando o seguinte comando

```
oc create -f <backend-filename.yaml> -n trident
oc create -f < storageclass-filename.yaml>
oc create -f <snapshotclass-filename.yaml>
```

Definindo padrões com Trident Storage e Snapshot Class

Definindo padrões com Trident Storage e Snapshot Class

Agora você pode tornar a classe de armazenamento trident necessária e a classe de instantâneo de volume como padrão no OpenShift Cluster. Conforme mencionado anteriormente, é necessário definir a classe de armazenamento padrão e a classe de instantâneo de volume para permitir que o OpenShift Virtualization disponibilize a fonte de imagem dourada para criar VMs a partir de modelos padrão.

Você pode definir a classe de armazenamento Trident e a classe de snapshot como padrão editando a anotação no console ou aplicando patches na linha de comando com o seguinte.

```
storageclass.kubernetes.io/is-default-class:true
or
kubectl patch storageclass standard -p '{"metadata": {"annotations":{
"storageclass.kubernetes.io/is-default-class":"true"}}}'

storageclass.kubevirt.io/is-default-virt-class: true
or
kubectl patch storageclass standard -p '{"metadata": {"annotations":{
"storageclass.kubevirt.io/is-default-virt-class": "true"}}}'
```

Depois que isso estiver definido, você pode excluir quaisquer objetos dv e VolumeSnapShot pré-existentes usando o seguinte comando:

```
oc delete dv,VolumeSnapshot -n openshift-virtualization-os-images
--selector=cdi.kubevirt.io/dataImportCron
```

Informações sobre direitos autorais

Copyright © 2025 NetApp, Inc. Todos os direitos reservados. Impresso nos EUA. Nenhuma parte deste documento protegida por direitos autorais pode ser reproduzida de qualquer forma ou por qualquer meio — gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, gravação em fita ou storage em um sistema de recuperação eletrônica — sem permissão prévia, por escrito, do proprietário dos direitos autorais.

O software derivado do material da NetApp protegido por direitos autorais está sujeito à seguinte licença e isenção de responsabilidade:

ESTE SOFTWARE É FORNECIDO PELA NETAPP "NO PRESENTE ESTADO" E SEM QUAISQUER GARANTIAS EXPRESSAS OU IMPLÍCITAS, INCLUINDO, SEM LIMITAÇÕES, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO, CONFORME A ISENÇÃO DE RESPONSABILIDADE DESTE DOCUMENTO. EM HIPÓTESE ALGUMA A NETAPP SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, INCIDENTAL, ESPECIAL, EXEMPLAR OU CONSEQUENCIAL (INCLUINDO, SEM LIMITAÇÕES, AQUISIÇÃO DE PRODUTOS OU SERVIÇOS SOBRESSALENTES; PERDA DE USO, DADOS OU LUCROS; OU INTERRUPÇÃO DOS NEGÓCIOS), INDEPENDENTEMENTE DA CAUSA E DO PRINCÍPIO DE RESPONSABILIDADE, SEJA EM CONTRATO, POR RESPONSABILIDADE OBJETIVA OU PREJUÍZO (INCLUINDO NEGLIGÊNCIA OU DE OUTRO MODO), RESULTANTE DO USO DESTE SOFTWARE, MESMO SE ADVERTIDA DA RESPONSABILIDADE DE TAL DANO.

A NetApp reserva-se o direito de alterar quaisquer produtos descritos neste documento, a qualquer momento e sem aviso. A NetApp não assume nenhuma responsabilidade nem obrigação decorrentes do uso dos produtos descritos neste documento, exceto conforme expressamente acordado por escrito pela NetApp. O uso ou a compra deste produto não representam uma licença sob quaisquer direitos de patente, direitos de marca comercial ou quaisquer outros direitos de propriedade intelectual da NetApp.

O produto descrito neste manual pode estar protegido por uma ou mais patentes dos EUA, patentes estrangeiras ou pedidos pendentes.

LEGENDA DE DIREITOS LIMITADOS: o uso, a duplicação ou a divulgação pelo governo estão sujeitos a restrições conforme estabelecido no subparágrafo (b)(3) dos Direitos em Dados Técnicos - Itens Não Comerciais no DFARS 252.227-7013 (fevereiro de 2014) e no FAR 52.227- 19 (dezembro de 2007).

Os dados aqui contidos pertencem a um produto comercial e/ou serviço comercial (conforme definido no FAR 2.101) e são de propriedade da NetApp, Inc. Todos os dados técnicos e software de computador da NetApp fornecidos sob este Contrato são de natureza comercial e desenvolvidos exclusivamente com despesas privadas. O Governo dos EUA tem uma licença mundial limitada, irrevogável, não exclusiva, intransferível e não sublicenciável para usar os Dados que estão relacionados apenas com o suporte e para cumprir os contratos governamentais desse país que determinam o fornecimento de tais Dados. Salvo disposição em contrário no presente documento, não é permitido usar, divulgar, reproduzir, modificar, executar ou exibir os dados sem a aprovação prévia por escrito da NetApp, Inc. Os direitos de licença pertencentes ao governo dos Estados Unidos para o Departamento de Defesa estão limitados aos direitos identificados na cláusula 252.227-7015(b) (fevereiro de 2014) do DFARS.

Informações sobre marcas comerciais

NETAPP, o logotipo NETAPP e as marcas listadas em http://www.netapp.com/TM são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respectivos proprietários.