



安全性 Trident

NetApp
January 14, 2026

目录

- 安全性..... 1
 - 安全性..... 1
 - 在自己的命名空间中运行Trident 1
 - 对 ONTAP SAN 后端使用 CHAP 身份验证 1
 - 对 NetApp HCI 和 SolidFire 后端使用 CHAP 身份验证 1
 - 将Trident与NVE和NAE结合使用 1
 - Linux统一密钥设置(LUKS)..... 2
 - 启用LUKS加密..... 2
 - 用于导入LUKS卷的后端配置 4
 - 用于导入UKS卷的PVC配置 4
 - 轮换LUKS密码短语 5
 - 启用卷扩展..... 7
 - Kerberos传输中加密..... 7
 - 使用内部ONTAP卷配置传输中的Kerberos加密 8
 - 为Azure NetApp Files卷配置传输中的Kerberos加密 12

安全性

安全性

请按照此处列出的建议确保Trident安装安全。

在自己的命名空间中运行Trident

请务必防止应用程序、应用程序管理员、用户和管理应用程序访问Trident对象定义或Pod、以确保可靠的存储并阻止潜在的恶意活动。

要将其他应用程序和用户与Trident分离，请始终将Trident安装在其自身的Kubernetes命名空间中(`trident`)。将Trident置于自己的命名空间中可确保只有Kubernetes管理人员才能访问Trident Pod和存储在命名空间CRD对象中的项目(如适用、例如后端和CHAP密码)。您应确保仅允许管理员访问Trident命名空间、从而访问`tridentctl`应用程序。

对 ONTAP SAN 后端使用 CHAP 身份验证

Trident支持对ONTAP SAN工作负载进行基于CHAP的身份验证(使用`ontap-san`和`ontap-san-economy`驱动程序)。NetApp建议对Trident使用双向CHAP在主机和存储后端之间进行身份验证。

对于使用SAN存储驱动程序的ONTAP后端，Trident可以通过设置双向CHAP并管理CHAP用户名和机密`tridentctl`。要了解Trident如何在ONTAP后端配置CHAP、请参见["准备使用ONTAP SAN驱动程序配置后端"](#)。

对 NetApp HCI 和 SolidFire 后端使用 CHAP 身份验证

NetApp 建议部署双向 CHAP，以确保主机与 NetApp HCI 和 SolidFire 后端之间的身份验证。Trident使用一个机密对象、其中每个租户包含两个CHAP密码。安装Trident后、它会管理CHAP密钥、并将其存储在相应PV的CR对象中`tridentvolume`。创建PV时、Trident会使用CHAP密码启动iSCSI会话并通过CHAP与NetApp HCI和SolidFire系统进行通信。



Trident创建的卷不与任何卷访问组关联。

将Trident与NVE和NAE结合使用

NetApp ONTAP 提供空闲数据加密、可在磁盘被盗、退回或重新利用时保护敏感数据。有关详细信息，请参见["配置 NetApp 卷加密概述"](#)。

- 如果在后端启用了NAE、则在Trident中配置的任何卷都将启用NAE。
 - 您可以将NVE加密标志设置为`""`、以创建启用了NAE的卷。
- 如果未在后端启用NAE、则在Trident中配置的任何卷都将启用NVE、除非在后端配置中NVE加密标志设置为`false` (默认值)。

在启用了NAE的后端的Trident中创建的卷必须已进行NVE或NAE加密。



- 您可以在Trident后端配置中将NVE加密标志设置为 `true`、以覆盖NAE加密并按卷使用特定的加密密钥。
- 如果在启用了NAE的后端将NVE加密标志设置为 `false`、则会创建启用了NAE的卷。您不能通过将NVE加密标志设置为来禁用NAE加密 `false`。

- 您可以通过将NVE加密标志显式设置为来在Trident中手动创建NVE卷 `true`。

有关后端配置选项的详细信息、请参见：

- ["ONTAP SAN配置选项"](#)
- ["ONTAP NAS配置选项"](#)

Linux统一密钥设置(LUKS)

您可以启用Linux统一密钥设置(Unified Key Setup、LKS)来对Trident上的ONTAP SAN和ONTAP SAN经济型卷进行加密。Trident支持对经过LUN加密的卷进行密码短语轮换和卷扩展。

在Trident中，根据的建议，经过LUN加密的卷使用AES-XTS-PLAN 64 cypher和模式"[NIST](#)"。

开始之前

- 工作节点必须安装加密设置2.1或更高版本(但低于3.0)。有关详细信息，请访问["Gitlab：密码设置"](#)。
- 出于性能原因、NetApp建议工作节点支持高级加密标准新指令(AES-NI)。要验证AES-NI支持、请运行以下命令：

```
grep "aes" /proc/cpuinfo
```

如果未返回任何内容、则您的处理器不支持AES-NI。有关AES-NI的详细信息，请访问：["Intel：高级加密标准说明\(AES-NI\)"](#)。

启用LUKS加密

您可以对ONTAP SAN和ONTAP SAN经济卷使用Linux统一密钥设置(Unified Key Setup、LUKS)启用每个卷的主机端加密。

步骤

1. 在后端配置中定义LUKS加密属性。有关ONTAP SAN后端配置选项的详细信息，请参阅["ONTAP SAN配置选项"](#)。

```

{
  "storage": [
    {
      "labels": {
        "luks": "true"
      },
      "zone": "us_east_1a",
      "defaults": {
        "luksEncryption": "true"
      }
    },
    {
      "labels": {
        "luks": "false"
      },
      "zone": "us_east_1a",
      "defaults": {
        "luksEncryption": "false"
      }
    }
  ]
}

```

2. `parameters.selector`用于定义使用了一个使用了此加密的存储池。例如：

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: luks
provisioner: csi.trident.netapp.io
parameters:
  selector: "luks=true"
  csi.storage.k8s.io/node-stage-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}

```

3. 创建一个包含LUKS密码短语的密钥。例如：

```
kubectl -n trident create -f luks-pvc1.yaml
apiVersion: v1
kind: Secret
metadata:
  name: luks-pvc1
stringData:
  luks-passphrase-name: A
  luks-passphrase: secretA
```

限制

LUKS加密的卷无法利用ONTAP 重复数据删除和数据压缩功能。

用于导入LUKS卷的后端配置

要导入LUN卷、必须在后端将设置 `luksEncryption`` 为 ``true`。该 `luksEncryption`` 选项会告诉Trident卷是符合LUN (``true`(兼容)还是不符合LUN (`false`(兼容)，如以下示例所示。

```
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: trident_svm
username: admin
password: password
defaults:
  luksEncryption: 'true'
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'
```

用于导入UKS卷的PVC配置

要动态导入LUKS卷、请将标注设置 `trident.netapp.io/luksEncryption`` 为 ``true`、并在PVC中包含启用了LUKS的存储类、如本示例所示。

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: luks-pvc
  namespace: trident
  annotations:
    trident.netapp.io/luksEncryption: "true"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: luks-sc

```

轮换LUKS密码短语

您可以轮换LUKS密码短语并确认轮换。



请勿忘记密码短语、除非您确认任何卷、快照或密钥不再引用它。如果引用的密码短语丢失、您可能无法挂载此卷、并且数据将保持加密状态且无法访问。

关于此任务

如果在指定新的LUKS密码短语后创建了挂载卷的POD、则会发生LUKS密码短语轮换。创建新Pod时、Trident会将卷上的LUN密码短语与密钥中的活动密码短语进行比较。

- 如果卷上的密码短语与密钥中的活动密码短语不匹配、则会发生轮换。
- 如果卷上的密码短语与密钥中的活动密码短语匹配、`previous-luks-passphrase`则会忽略参数。

步骤

1. 添加 `node-publish-secret-name` 和 `node-publish-secret-namespace` StorageClass参数。例如：

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-san
provisioner: csi.trident.netapp.io
parameters:
  trident.netapp.io/backendType: "ontap-san"
  csi.storage.k8s.io/node-stage-secret-name: luks
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
  csi.storage.k8s.io/node-publish-secret-name: luks
  csi.storage.k8s.io/node-publish-secret-namespace: ${pvc.namespace}

```

2. 确定卷或快照上的现有密码短语。

卷

```
tridentctl -d get volume luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>

...luksPassphraseNames:["A"]
```

Snapshot

```
tridentctl -d get snapshot luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>/<snapshotID>

...luksPassphraseNames:["A"]
```

3. 更新卷的LUKS密钥以指定新密码短语和上一密码短语。确保 `previous-luke-passphrase-name` 与 `previous-luks-passphrase` 与先前的密码短语匹配。

```
apiVersion: v1
kind: Secret
metadata:
  name: luks-pvc1
stringData:
  luks-passphrase-name: B
  luks-passphrase: secretB
  previous-luks-passphrase-name: A
  previous-luks-passphrase: secretA
```

4. 创建一个新的装载卷的POD。这是启动轮换所必需的。

5. 验证密码短语是否已轮换。

卷

```
tridentctl -d get volume luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>

...luksPassphraseNames:["B"]
```

Snapshot

```
tridentctl -d get snapshot luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>/<snapshotID>

...luksPassphraseNames: ["B"]
```

结果

仅在卷和快照上返回新密码短语时、才会轮换密码短语。



如果返回两个密码，例如 `luksPassphraseNames: ["B", "A"]`，则旋转不完整。您可以触发新POD以尝试完成轮换。

启用卷扩展

您可以在LUKS加密的卷上启用卷扩展。

步骤

1. 启用 `CSINodeExpandSecret` 特征门(beta 1.25+)。有关详细信息、请参见 ["Kubernetes 1.25: 使用机密进行节点驱动型CSI卷扩展"](#)。
2. 添加 `node-expand-secret-name` 和 `node-expand-secret-namespace` StorageClass参数。例如：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: luks
provisioner: csi.trident.netapp.io
parameters:
  selector: "luks=true"
  csi.storage.k8s.io/node-stage-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
  csi.storage.k8s.io/node-expand-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-expand-secret-namespace: ${pvc.namespace}
allowVolumeExpansion: true
```

结果

启动联机存储扩展时、kubelet会将相应的凭据传递给驱动程序。

Kerberos传输中加密

通过使用Kerberos传输中加密、您可以对受管集群和存储后端之间的流量启用加密、从而提高数据访问安全性。

Trident支持将ONTAP用作存储后端的Kerberos加密：

- 内部部署**Kubernetes**—Trident支持通过从ONTAP OpenShift和上游Kubernetes集群到内部部署ONTAP卷的NFS3和NFSv4连接进行Kerberos加密。

您可以创建、删除、调整大小、创建快照、克隆、只读克隆、并导入使用NFS加密的卷。

使用内部ONTAP卷配置传输中的Kerberos加密

您可以对受管集群与内部ONTAP存储后端之间的存储流量启用Kerberos加密。



只有使用存储驱动程序、才支持对使用内部ONTAP存储后端的NFS流量进行Kerberos加密
ontap-nas。

开始之前

- 确保您可以访问该 `tridentctl` 实用程序。
- 确保您对ONTAP存储后端具有管理员访问权限。
- 确保您知道要从ONTAP存储后端共享的一个或多个卷的名称。
- 确保已准备好ONTAP Storage VM以支持NFS卷的Kerberos加密。有关说明、请参见 ["在dataLIF上启用Kerberos"](#)。
- 确保已正确配置使用Kerberos加密的任何NFSv4卷。请参阅的“NetApp NFSv4域配置”一节(第13页) "[《NetApp NFSv4增强功能和最佳实践指南》](#)"。

添加或修改ONTAP导出策略

您需要向现有ONTAP导出策略添加规则、或者创建新的导出策略、以便对ONTAP Storage VM根卷以及与上游Kubornetes集群共享的任何ONTAP卷支持Kerberos加密。您添加的导出策略规则或创建的新导出策略需要支持以下访问协议和访问权限：

访问协议

使用NFS、NFSv3和NFSv4访问协议配置导出策略。

访问详细信息

您可以根据卷的需求配置以下三种不同版本的Kerberos加密之一：

- **Kerberos 5**-(身份验证和加密)
- **Kerberos 5i**-(身份验证和加密与身份保护)
- **Kerberos 5p**-(身份验证和加密、具有身份和隐私保护功能)

使用适当的访问权限配置ONTAP导出策略规则。例如、如果集群要挂载混合使用Kerberos 5i和Kerberos 5p加密的NFS卷、请使用以下访问设置：

键入	只读访问	读/写访问	超级用户访问
UNIX	已启用	已启用	已启用
Kerberos 5i	已启用	已启用	已启用

键入	只读访问	读/写访问	超级用户访问
Kerberos 5p	已启用	已启用	已启用

有关如何创建ONTAP导出策略和导出策略规则、请参见以下文档：

- ["创建导出策略"](#)
- ["向导出策略添加规则"](#)

创建存储后端

您可以创建包含Kerberos加密功能的Trident存储后端配置。

关于此任务

在创建用于配置Kerberos加密的存储后端配置文件时、您可以使用参数指定以下三种不同版本的Kerberos加密之一 `spec.nfsMountOptions`：

- `spec.nfsMountOptions: sec=krb5` (身份验证和加密)
- `spec.nfsMountOptions: sec=krb5i` (身份验证和加密以及身份保护)
- `spec.nfsMountOptions: sec=krb5p` (身份验证和加密以及身份和隐私保护)

请仅指定一个Kerberos级别。如果在参数列表中指定多个Kerberos加密级别、则仅会使用第一个选项。

步骤

1. 在受管集群上、使用以下示例创建存储后端配置文件。将括号<>中的值替换为您环境中的信息：

```

apiVersion: v1
kind: Secret
metadata:
  name: backend-ontap-nas-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  storageDriverName: "ontap-nas"
  managementLIF: <STORAGE_VM_MGMT_LIF_IP_ADDRESS>
  dataLIF: <PROTOCOL_LIF_FQDN_OR_IP_ADDRESS>
  svm: <STORAGE_VM_NAME>
  username: <STORAGE_VM_USERNAME_CREDENTIAL>
  password: <STORAGE_VM_PASSWORD_CREDENTIAL>
  nasType: nfs
  nfsMountOptions: ["sec=krb5i"] #can be krb5, krb5i, or krb5p
  qtreesPerFlexvol:
  credentials:
    name: backend-ontap-nas-secret

```

2. 使用您在上一步中创建的配置文件创建后端：

```
tridentctl create backend -f <backend-configuration-file>
```

如果后端创建失败，则后端配置出现问题。您可以运行以下命令来查看日志以确定发生原因：

```
tridentctl logs
```

确定并更正配置文件中的问题后，您可以再次运行 create 命令。

创建存储类。

您可以创建存储类来配置采用Kerberos加密的卷。

关于此任务

创建存储类对象时、可以使用参数指定以下三种不同版本的Kerberos加密之一 mountOptions：

- mountOptions: sec=krb5 (身份验证和加密)
- mountOptions: sec=krb5i (身份验证和加密以及身份保护)
- mountOptions: sec=krb5p (身份验证和加密以及身份和隐私保护)

请仅指定一个Kerberos级别。如果在参数列表中指定多个Kerberos加密级别、则仅会使用第一个选项。如果您在存储后端配置中指定的加密级别与您在存储类对象中指定的加密级别不同、则存储类对象优先。

步骤

1. 使用以下示例创建StorageClass Kubernetes对象：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas-sc
provisioner: csi.trident.netapp.io
mountOptions:
  - sec=krb5i #can be krb5, krb5i, or krb5p
parameters:
  backendType: ontap-nas
  storagePools: ontapnas_pool
  trident.netapp.io/nasType: nfs
allowVolumeExpansion: true
```

2. 创建存储类：

```
kubectl create -f sample-input/storage-class-ontap-nas-sc.yaml
```

3. 确保已创建存储类：

```
kubectl get sc ontap-nas-sc
```

您应看到类似于以下内容的输出：

NAME	PROVISIONER	AGE
ontap-nas-sc	csi.trident.netapp.io	15h

配置卷

创建存储后端和存储类后、您现在可以配置卷。有关说明，请参阅 ["配置卷"](#)。

为Azure NetApp Files卷配置传输中的Kerberos加密

您可以对受管集群与单个Azure NetApp Files存储后端或Azure NetApp Files存储后端虚拟池之间的存储流量启用Kerberos加密。

开始之前

- 确保已在受管Red Hat OpenShift集群上启用Trident。
- 确保您可以访问该 `tridentctl` 实用程序。
- 请注意要求并按照中的说明，确保已为Kerberos加密准备好Azure NetApp Files存储后端 "[Azure NetApp Files 文档](#)"。
- 确保已正确配置使用Kerberos加密的任何NFSv4卷。请参阅的“NetApp NFSv4域配置”一节(第13页) "[《NetApp NFSv4增强功能和最佳实践指南》](#)"。

创建存储后端

您可以创建包含Kerberos加密功能的Azure NetApp Files存储后端配置。

关于此任务

在创建配置Kerberos加密的存储后端配置文件时、您可以对其进行定义、使其应用于以下两个可能的级别之一：

- 使用字段的*存储后端级别* `spec.kerberos`
- 使用字段的*虚拟池级别* `spec.storage.kerberos`

在虚拟池级别定义配置时、系统会使用存储类中的标签来选择该池。

在任一级别、您都可以指定以下三种不同版本的Kerberos加密之一：

- `kerberos: sec=krb5` (身份验证和加密)
- `kerberos: sec=krb5i` (身份验证和加密以及身份保护)
- `kerberos: sec=krb5p` (身份验证和加密以及身份和隐私保护)

步骤

1. 在受管集群上、根据需要定义存储后端的位置(存储后端级别或虚拟池级别)、使用以下示例之一创建存储后端配置文件。将括号<>中的值替换为您环境中的信息：

存储后端级别示例

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-secret
```

虚拟池级别示例

```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  storage:
    - labels:
        type: encryption
        kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-secret

```

2. 使用您在上一步中创建的配置文件创建后端：

```
tridentctl create backend -f <backend-configuration-file>
```

如果后端创建失败，则后端配置出现问题。您可以运行以下命令来查看日志以确定发生原因：

```
tridentctl logs
```

确定并更正配置文件中的问题后，您可以再次运行 create 命令。

创建存储类。

您可以创建存储类来配置采用Kerberos加密的卷。

步骤

1. 使用以下示例创建StorageClass Kubernetes对象：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-nfs
provisioner: csi.trident.netapp.io
parameters:
  backendType: azure-netapp-files
  trident.netapp.io/nasType: nfs
  selector: type=encryption
```

2. 创建存储类：

```
kubectl create -f sample-input/storage-class-sc-nfs.yaml
```

3. 确保已创建存储类：

```
kubectl get sc -sc-nfs
```

您应看到类似于以下内容的输出：

NAME	PROVISIONER	AGE
sc-nfs	csi.trident.netapp.io	15h

配置卷

创建存储后端和存储类后、您现在可以配置卷。有关说明，请参阅 ["配置卷"](#)。

版权信息

版权所有 © 2026 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。