安全性
Astra Trident

NetApp
January 14, 2026

This PDF was generated from https://docs.netapp.com/zh-tw/trident-2402/trident-reco/security-reco.html on January 14, 2026. Always check docs.netapp.com for the latest.

目錄

安全性 1
安全性 1
在自己的命名空間中執行Astra Trident 1
使用CHAP驗證搭配ONTAP 使用支援SAN的功能 1
使用CHAP驗證NetApp HCI 搭配不景和SolidFire 不景的後端 1
使用Astra Trident搭配NVE和NAE 1
Linux統一化金鑰設定（LUKS） 2
啟用LUKS加密 2
用於匯入 LUKS Volume 的後端組態 4
旋轉LUKS複雜密碼 4
啟用Volume擴充 6
設定在線上 Kerberos 加密 7
使用內部部署的 ONTAP 磁碟區來設定在線上 Kerberos 加密 7
使用 Azure NetApp Files 磁碟區設定在線上 Kerberos 加密 11

安全性

安全性

請使用此處列出的建議、確保您的Astra Trident安裝安全無虞。

在自己的命名空間中執行**Astra Trident**

防止應用程式、應用程式管理員、使用者及管理應用程式存取Astra Trident物件定義或Pod、以確保可靠的儲存並封鎖潛在的惡意活動、這一點非常重要。

若要將其他應用程式和使用者與Astra Trident區隔開、請務必在自己的Kubernetes命名空間中安裝Astra Trident（`trident`）。將Astra Trident放在自己的命名空間中、可確保只有Kubernetes管理人員能夠存取Astra Trident Pod、以及儲存在命名式CRD物件中的成品（例如後端和CHAP機密）。
您應確保只允許系統管理員存取Astra Trident命名空間、進而存取 `tridentctl` 應用程式：

使用**CHAP**驗證搭配**ONTAP** 使用支援**SAN**的功能

Astra Trident支援ONTAP 以CHAP為基礎的驗證功能、適用於各種不實的SAN工作負載（使用 `ontap-san` 和 `ontap-san-economy` 驅動程式）。NetApp建議使用雙向CHAP搭配Astra Trident進行主機與儲存後端之間的驗證。

針對使用SAN儲存驅動程式的幕後作業、Astra Trident可設定雙向CHAP、並透過管理CHAP使用者名稱和機密ONTAP `tridentctl`。
請參閱 "" 瞭解Astra Trident如何在ONTAP 不景點上設定CHAP。

使用**CHAP**驗證**NetApp HCI** 搭配不景和**SolidFire** 不景的後端

NetApp建議部署雙向CHAP、以確保主機與NetApp HCI 支援功能及SolidFire 支援功能之間的驗證。Astra Trident使用每個租戶包含兩個CHAP密碼的秘密物件。安裝 Astra Trident 時、它會管理 CHAP 機密、並將其儲存在中 `tridentvolume` 對應PV的CR物件。建立 PV 時、 Astra Trident 會使用 CHAP 機密來啟動 iSCSI 工作階段、並透過 CHAP 與 NetApp HCI 和 SolidFire 系統通訊。

> Astra Trident 所建立的磁碟區不會與任何 Volume Access Group 相關聯。

使用**Astra Trident**搭配**NVE**和**NAE**

NetApp ONTAP 支援閒置資料加密、可在磁碟遭竊、退回或重新使用時、保護敏感資料。如需詳細資訊、請參閱 "設定NetApp Volume Encryption總覽"。

- 如果在後端啟用NAE、則Astra Trident中配置的任何磁碟區都將啟用NAE。
- 如果後端未啟用NAE、則在Astra Trident中配置的任何Volume都會啟用NVE、除非您將NVE加密旗標設為 `false` 在後端組態中。

在啟用NAE的後端Astra Trident中建立的磁碟區、必須加密NVE或NAE。

- 您可以將NVE加密旗標設為 `true` 在Trident後端組態中、覆寫NAE加密、並以每個磁碟區為基礎使用特定的加密金鑰。
- 將NVE加密旗標設定為 `false` 在啟用NAE的後端上、將會建立啟用NAE的Volume。您無法將NVE加密旗標設為、以停用NAE加密 `false`。

- 您可以在Astra Trident中手動建立NVE磁碟區、方法是將NVE加密旗標明確設定為 `true`。

如需後端組態選項的詳細資訊、請參閱：

- "支援SAN組態選項ONTAP"
- "ASNAS組態選項ONTAP"

Linux統一化金鑰設定（LUKS）

您可以在ONTAP Astra Trident上啟用Linux Unified Key Setup（LUKS）來加密支援的SAN和ONTAP 支援的SAN經濟版磁碟區。Astra Trident支援使用密碼的輪替和磁碟區擴充、適用於使用LUKS加密的磁碟區。

在Astra Trident中、LUKS加密的磁碟區會依照所建議的方式使用AES-XTS-plain64 cypher和模式 "NIST"。

開始之前

- 工作者節點必須安裝密碼設定2.1或更高版本（但低於3.0）。如需詳細資訊、請造訪 "Gitlab：密碼設定"。
- 基於效能考量、我們建議工作節點支援進階加密標準新增指令（AES-NI）。若要驗證AES-NI支援、請執行下列命令：

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

如果沒有歸還任何內容、您的處理器就不支援AES-NI。如需AES-NI的詳細資訊、請造訪： "Intel：進階加密標準指令（AES-NI）"。

啟用LUKS加密

您可以使用Linux Unified Key Setup（LUKS）來啟用每個Volume、主機端的加密功能、以利ONTAP 執行SAN和ONTAP 支援SAN經濟效益的磁碟區。

步驟

1. 在後端組態中定義LUKS加密屬性。如需ONTAP 有關支援不支援SAN的後端組態選項的詳細資訊、請參閱 "支援SAN組態選項ONTAP"。

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

2. 使用 `parameters.selector` 使用LUKS加密定義儲存資源池。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. 建立包含LUKS通關密碼的秘密。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加密磁碟區無法利用ONTAP 重複資料刪除技術與壓縮技術。

用於匯入 **LUKS Volume** 的後端組態

若要匯入 LUKS Volume 、您必須設定 `luksEncryption` 至(`true` 在後端。。 `luksEncryption` 選項會告訴 Astra Trident 、磁碟區是否符合 LUKS 標準 (`true`）或不符合 LUKS 規範 (`false`）如以下範例所示。

```
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'
```

旋轉**LUKS**複雜密碼

您可以旋轉LUKS複雜密碼並確認輪調。

> 請勿忘記密碼、除非您已驗證它不再被任何磁碟區、快照或機密所引用。如果參考的通關密碼遺失、您可能無法掛載磁碟區、而且資料將保持加密且無法存取。

關於這項工作

如果在指定新的LUKS通關密碼之後建立裝載磁碟區的Pod、則會發生LUKS通關密碼循環。建立新的Pod時、Astra Trident會比較磁碟區上的LUKS通關密碼與機密中的作用中通關密碼。

- 如果磁碟區上的通關密碼與機密中的作用中通關密碼不相符、就會發生輪調。
- 如果磁碟區上的通關密碼與機密中的作用中通關密碼相符 `previous-luks-passphrase` 參數被忽略。

步驟

1. 新增 `node-publish-secret-name` 和 `node-publish-secret-namespace` StorageClass參數。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. 識別磁碟區或快照上的現有密碼。

Volume

```
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. 更新磁碟區的LUKS機密、以指定新的和先前的密碼。確保 `previous-luke-passphrase-name` 和 `previous-luks-passphrase` 請與先前的通關密碼相符。

```
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. 建立新的Pod以掛載Volume。這是啟動旋轉所需的。
5. 確認複雜密碼已旋轉。

Volume

```
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"]
```

結果

只有在磁碟區和快照上傳回新的通關密碼時、才會旋轉通關密碼。

> 例如、如果傳回兩個複雜密碼 `luksPassphraseNames: ["B", "A"]`、旋轉不完整。您可以觸發新的Pod以嘗試完成旋轉。

啟用Volume擴充

您可以在LUKS加密的Volume上啟用Volume擴充。

步驟

1. 啟用 `CSINodeExpandSecret` 功能閘道（beta 1.25 +）。請參閱 "Kubernetes 1.25：使用Secrets進行節點導向的SCSI Volume擴充" 以取得詳細資料。
2. 新增 `node-expand-secret-name` 和 `node-expand-secret-namespace` StorageClass參數。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
```

結果

當您啟動線上儲存擴充時、kubelet會將適當的認證資料傳遞給驅動程式。

設定在線上 Kerberos 加密

使用 Astra Control Provisioner 、您可以為託管叢集與儲存後端之間的流量啟用加密、藉此改善資料存取安全性。

Astra Control Provisioner 支援透過 NFSv3 和 NFSv4 連線進行 Kerberos 加密，從 Red Hat OpenShift 和上游 Kubernetes 叢集到內部部署 ONTAP 磁碟區。

您可以建立、刪除、調整大小、快照、複製、 唯讀複製及匯入使用 NFS 加密的磁碟區。

使用內部部署的 ONTAP 磁碟區來設定在線上 Kerberos 加密

您可以在託管叢集與內部部署 ONTAP 儲存後端之間的儲存流量上啟用 Kerberos 加密。

> 只有使用儲存驅動程式、才支援內部部署 ONTAP 儲存後端的 NFS 流量 Kerberos 加密 `ontap-nas` 。

開始之前

- 請確定您已 "已啟用 Astra Control Provisioner" 在託管叢集上。
- 請確定您可以存取 `tridentctl` 公用程式。
- 確保您具有 ONTAP 儲存後端的管理員存取權。
- 確保您知道將從 ONTAP 儲存後端共用的磁碟區名稱。
- 請確定您已準備好 ONTAP 儲存 VM 、以支援 NFS 磁碟區的 Kerberos 加密。請參閱 "在資料LIF上啟用Kerberos" 以取得指示。
- 請確定您使用 Kerberos 加密的任何 NFSv4 磁碟區都已正確設定。請參閱的 NetApp NFSv4 網域組態一節（第 13 頁） "NetApp NFSv4 增強與最佳實務指南"。

新增或修改 **ONTAP** 匯出原則

您需要將規則新增至現有的 ONTAP 匯出原則、或建立新的匯出原則、以支援 ONTAP 儲存 VM 根磁碟區的 Kerberos 加密、以及與上游 Kubernetes 叢集共用的任何 ONTAP 磁碟區。您新增的匯出原則規則或您建立的新匯出原則需要支援下列存取通訊協定和存取權限：

存取傳輸協定

使用 NFS 、 NFSv3 和 NFSv4 存取通訊協定來設定匯出原則。

存取詳細資料

您可以根據對磁碟區的需求、設定 Kerberos 加密的三個不同版本之一：

- * Kerberos 5* -（驗證與加密）
- * Kerberos 5i* -（身分識別保護的驗證與加密）
- * Kerberos 5p* -（身分識別與隱私保護的驗證與加密）

使用適當的存取權限來設定 ONTAP 匯出原則規則。例如、如果叢集將使用 Kerberos 5i 和 Kerberos 5p 加密混

合安裝 NFS 磁碟區、請使用下列存取設定：

類型	唯讀存取	讀取 / 寫入存取權	超級使用者存取權
UNIX	已啟用	已啟用	已啟用
Kerberos 5i	已啟用	已啟用	已啟用
Kerberos 5p	已啟用	已啟用	已啟用

請參閱下列文件、瞭解如何建立 ONTAP 匯出原則和匯出原則規則：

- "建立匯出原則"
- "新增規則至匯出原則"

建立儲存後端

您可以建立內含 Kerberos 加密功能的 Astra Control Provisioner 儲存後端組態。

關於這項工作

當您建立設定 Kerberos 加密的儲存後端組態檔時、可以使用參數指定 Kerberos 加密的三個不同版本之一 `spec.nfsMountOptions` ：

- `spec.nfsMountOptions: sec=krb5` （驗證與加密）
- `spec.nfsMountOptions: sec=krb5i` （身分識別保護的驗證與加密）
- `spec.nfsMountOptions: sec=krb5p` （身分識別與隱私保護的驗證與加密）

只指定一個 Kerberos 層級。如果您在參數清單中指定多個 Kerberos 加密層級、則只會使用第一個選項。

步驟

1. 在託管叢集上、使用下列範例建立儲存後端組態檔案。以您環境的資訊取代括弧 <> 中的值：

```yaml
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. 使用您在上一個步驟中建立的組態檔來建立後端：

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

如果後端建立失敗、表示後端組態有問題。您可以執行下列命令來檢視記錄、以判斷原因：

```
tridentctl logs
```

識別並修正組態檔的問題之後、您可以再次執行create命令。

建立儲存類別

您可以建立儲存類別、以使用 Kerberos 加密來配置磁碟區。

關於這項工作

當您建立儲存類別物件時、可以使用下列參數、指定 Kerberos 加密的三個不同版本之一 `mountOptions` ：

- `mountOptions: sec=krb5` （驗證與加密）
- `mountOptions: sec=krb5i` （身分識別保護的驗證與加密）
- `mountOptions: sec=krb5p` （身分識別與隱私保護的驗證與加密）

只指定一個 Kerberos 層級。如果您在參數清單中指定多個 Kerberos 加密層級、則只會使用第一個選項。如果您在儲存後端組態中指定的加密層級與您在儲存類別物件中指定的層級不同、則儲存類別物件會優先。

步驟

1. 使用以下範例建立 StorageClass Kubernetes 物件：

```
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. 建立儲存類別：

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

3. 確定已建立儲存類別：

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

您應該會看到類似下列的輸出：

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

配置 Volume

建立儲存後端和儲存類別之後、您現在可以配置 Volume 。請參閱以下的說明 "資源配置"。

使用 **Azure NetApp Files** 磁碟區設定在線上 **Kerberos** 加密

您可以在託管叢集與單一 Azure NetApp Files 儲存後端或 Azure NetApp Files 儲存後端的虛擬集區之間的儲存流量上啟用 Kerberos 加密。

開始之前

- 確保您已在託管的 Red Hat OpenShift 叢集上啟用 Astra Control Provisioner 。請參閱 "啟用 Astra Control Provisioner" 以取得指示。
- 請確定您可以存取 `tridentctl` 公用程式。
- 請注意中的要求並遵循中的指示、以確保您已準備好 Azure NetApp Files 儲存後端進行 Kerberos 加密 "本文檔 Azure NetApp Files"。
- 請確定您使用 Kerberos 加密的任何 NFSv4 磁碟區都已正確設定。請參閱的 NetApp NFSv4 網域組態一節（第 13 頁） "NetApp NFSv4 增強與最佳實務指南"。

建立儲存後端

您可以建立包含 Kerberos 加密功能的 Azure NetApp Files 儲存後端組態。

關於這項工作

當您建立儲存後端組態檔案來設定 Kerberos 加密時、您可以加以定義、以便將其套用至下列兩種可能的層級之一：

- 使用欄位的 * 儲存後端層級 * `spec.kerberos`
- 使用欄位的 * 虛擬集區層級 * `spec.storage.kerberos`

當您在虛擬集區層級定義組態時、會使用儲存類別中的標籤來選取集區。

在任一層級、您都可以指定 Kerberos 加密的三個不同版本之一：

- `kerberos: sec=krb5` （驗證與加密）
- `kerberos: sec=krb5i` （身分識別保護的驗證與加密）
- `kerberos: sec=krb5p` （身分識別與隱私保護的驗證與加密）

步驟

1. 在託管叢集上、根據您需要定義儲存後端（儲存後端層級或虛擬集區層級）的位置、使用下列其中一個範例建立儲存後端組態檔案。以您環境的資訊取代括弧 <> 中的值：

儲存後端層級範例

```yaml
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
```

虛擬集區層級範例

```yaml
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. 使用您在上一個步驟中建立的組態檔來建立後端：

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

如果後端建立失敗、表示後端組態有問題。您可以執行下列命令來檢視記錄、以判斷原因：

```
tridentctl logs
```

識別並修正組態檔的問題之後、您可以再次執行create命令。

建立儲存類別

您可以建立儲存類別、以使用 Kerberos 加密來配置磁碟區。

步驟

1. 使用以下範例建立 StorageClass Kubernetes 物件：

```
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. 建立儲存類別：

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

3. 確定已建立儲存類別：

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

您應該會看到類似下列的輸出：

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

配置 **Volume**

建立儲存後端和儲存類別之後、您現在可以配置 Volume 。請參閱以下的說明 "資源配置"。

版權資訊

Copyright © 2026 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。 美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 http://www.netapp.com/TM 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。