



Trident Protect verwalten

Trident

NetApp
July 01, 2026

Inhalt

- Trident Protect verwalten 1
 - Trident Protect-Autorisierung und Zugriffskontrolle verwalten 1
 - Beispiel: Zugriff für zwei Gruppen von Benutzern verwalten 1
 - Überwachen Sie die Trident Protect-Ressourcen 7
 - Schritt 1: Installieren Sie die Überwachungstools 8
 - Schritt 2: Konfigurieren Sie die Überwachungstools so, dass sie zusammenarbeiten 10
 - Schritt 3: Benachrichtigungen und Benachrichtigungsziele konfigurieren 11
- Generieren Sie ein Trident Protect Support-Bundle 12
 - Überwachen und rufen Sie das Support-Paket ab 14
- Trident Protect aktualisieren 14
 - Schritt 1: Wählen Sie eine Version 15
 - Schritt 2: Trident Protect aktualisieren 15

Trident Protect verwalten

Trident Protect-Autorisierung und Zugriffskontrolle verwalten

Trident Protect nutzt das Kubernetes-Modell der rollenbasierten Zugriffssteuerung (RBAC). Standardmäßig stellt Trident Protect einen einzelnen System-Namespaces und das zugehörige Standard-Dienstkonto bereit. Wenn Sie eine Organisation mit vielen Benutzern oder spezifischen Sicherheitsanforderungen haben, können Sie die RBAC-Funktionen von Trident Protect nutzen, um den Zugriff auf Ressourcen und Namespaces detaillierter zu steuern.

Der Clusteradministrator hat stets Zugriff auf Ressourcen im `trident-protect` Standard-Namespaces und kann auch auf Ressourcen in allen anderen Namespaces zugreifen. Um den Zugriff auf Ressourcen und Anwendungen zu steuern, müssen Sie zusätzliche Namespaces erstellen und Ressourcen und Anwendungen zu diesen Namespaces hinzufügen.

Beachten Sie, dass keine Benutzer Anwendungsdatenverwaltungs-CRs im Standard `trident-protect` -Namespaces erstellen können. Sie müssen Anwendungsdatenverwaltungs-CRs in einem Anwendungs-Namespaces erstellen (als Best Practice erstellen Sie Anwendungsdatenverwaltungs-CRs im selben Namespaces wie die zugehörige Anwendung).

Nur Administratoren sollten Zugriff auf privilegierte Trident Protect benutzerdefinierte Ressourcenobjekte haben, darunter:



- **AppVault**: Erfordert Bucket-Anmeldeinformationen
- **AutoSupportBundle**: Erfasst Metriken, Protokolle und andere sensible Trident Protect-Daten
- **AutoSupportBundleSchedule**: Verwaltet Protokollerfassungspläne

Als bewährte Methode verwenden Sie rollenbasierte Zugriffssteuerung (RBAC), um den Zugriff auf privilegierte Objekte auf Administratoren zu beschränken.

Weitere Informationen darüber, wie RBAC den Zugriff auf Ressourcen und Namensräume regelt, finden Sie unter "[Kubernetes RBAC-Dokumentation](#)".

Weitere Informationen zu Servicekonten finden Sie in der "[Dokumentation zum Kubernetes Servicekonto](#)".

Beispiel: Zugriff für zwei Gruppen von Benutzern verwalten

Eine Organisation verfügt beispielsweise über einen Cluster-Administrator, eine Gruppe von Engineering-Benutzern und eine Gruppe von Marketing-Benutzern. Der Cluster-Administrator würde die folgenden Aufgaben ausführen, um eine Umgebung zu schaffen, in der die Engineering-Gruppe und die Marketing-Gruppe jeweils nur auf die Ressourcen zugreifen können, die ihren jeweiligen Namensräumen zugewiesen sind.

Schritt 1: Erstellen Sie einen Namensraum, um Ressourcen für jede Gruppe zu enthalten

Durch das Erstellen eines Namensraums können Sie Ressourcen logisch trennen und besser steuern, wer Zugriff auf diese Ressourcen hat.

Schritte

1. Erstellen Sie einen Namensraum für die Engineering-Gruppe:

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

2. Erstellen Sie einen Namensraum für die Marketinggruppe:

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

Schritt 2: Erstellen Sie neue Dienstkonten, um mit Ressourcen in jedem Namespace zu interagieren

Jeder neu erstellte Namespace verfügt über ein Standarddienstkonto, aber Sie sollten für jede Benutzergruppe ein Dienstkonto erstellen, damit Sie die Berechtigungen in Zukunft bei Bedarf weiter zwischen den Gruppen aufteilen können.

Schritte

1. Erstellen Sie ein Dienstkonto für die Engineering-Gruppe:

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

2. Erstellen Sie ein Servicekonto für die Marketinggruppe:

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

Schritt 3: Erstellen Sie ein Geheimnis für jedes neue Dienstkonto

Ein Dienstkontogeheimnis wird zur Authentifizierung mit dem Dienstkonto verwendet und kann im Falle einer Kompromittierung leicht gelöscht und neu erstellt werden.

Schritte

1. Erstellen Sie ein Secret für das Engineering-Service-Konto:

```
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. Erstellen Sie ein Geheimnis für das Marketing-Service-Konto:

```
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
```

Schritt 4: Erstellen Sie ein RoleBinding-Objekt, um das ClusterRole-Objekt an jedes neue Dienstkonto zu binden

Ein Standard-ClusterRole-Objekt wird erstellt, wenn Sie Trident Protect installieren. Sie können dieses ClusterRole an das Dienstkonto binden, indem Sie ein RoleBinding-Objekt erstellen und anwenden.

Schritte

1. Binden Sie die ClusterRole an das Engineering-Servicekonto:

```
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. Binden Sie die ClusterRole an das Marketing-Servicekonto:

```
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
```

Schritt 5: Berechtigungen testen

Testen Sie, ob die Berechtigungen korrekt sind.

Schritte

1. Bestätigen Sie, dass technische Benutzer auf technische Ressourcen zugreifen können:

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

2. Bestätigen Sie, dass technische Benutzer keinen Zugriff auf Marketingressourcen haben:

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

Schritt 6: Zugriff auf AppVault-Objekte gewähren

Um Datenverwaltungsaufgaben wie Backups und Snapshots durchzuführen, muss der Clusteradministrator einzelnen Benutzern Zugriff auf AppVault-Objekte gewähren.

Schritte

1. Erstellen und wenden Sie eine AppVault- und Secret-Kombinations-YAML-Datei an, die einem Benutzer Zugriff auf eine AppVault gewährt. Beispielsweise gewährt die folgende CR einem Benutzer Zugriff auf eine AppVault `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

```

- Erstellen und wenden Sie eine Role-CR an, um Clusteradministratoren die Möglichkeit zu geben, Zugriff auf bestimmte Ressourcen in einem Namespace zu gewähren. Beispiel:

```
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. Erstellen und wenden Sie eine RoleBinding CR an, um die Berechtigungen an den Benutzer eng-user zu binden. Beispiel:

```
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. Überprüfen Sie, ob die Berechtigungen korrekt sind.
- a. Versuch, AppVault-Objektinformationen für alle Namensräume abzurufen:

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

Sie sollten eine Ausgabe ähnlich der folgenden sehen:

```
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. Testen Sie, ob der Benutzer die AppVault-Informationen abrufen kann, auf die er nun Zugriffsberechtigung hat:

```
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
```

Sie sollten eine Ausgabe ähnlich der folgenden sehen:

```
yes
```

Ergebnis

Die Benutzer, denen Sie AppVault-Berechtigungen erteilt haben, sollten in der Lage sein, autorisierte AppVault-Objekte für Anwendungsdatenverwaltungsoperationen zu verwenden und sollten nicht in der Lage sein, auf Ressourcen außerhalb der zugewiesenen Namensräume zuzugreifen oder neue Ressourcen zu erstellen, auf die sie keinen Zugriff haben.

Überwachen Sie die Trident Protect-Ressourcen

Sie können die Open-Source-Tools kube-state-metrics, Prometheus und Alertmanager verwenden, um die Integrität der durch Trident Protect geschützten Ressourcen zu überwachen.

Der Dienst kube-state-metrics generiert Metriken aus der Kubernetes-API-Kommunikation. Die Verwendung mit Trident Protect liefert nützliche Informationen über den Zustand der Ressourcen in Ihrer Umgebung.

Prometheus ist ein Toolkit, das die von kube-state-metrics generierten Daten erfassen und als leicht lesbare Informationen über diese Objekte darstellen kann. Zusammen bieten kube-state-metrics und Prometheus eine Möglichkeit, den Zustand und Status der Ressourcen zu überwachen, die Sie mit Trident Protect verwalten.

Alertmanager ist ein Dienst, der die von Tools wie Prometheus gesendeten Warnmeldungen aufnimmt und sie an Ziele weiterleitet, die Sie konfigurieren.

Die in diesen Schritten enthaltenen Konfigurationen und Anleitungen sind lediglich Beispiele; Sie müssen sie an Ihre Umgebung anpassen. Spezifische Anweisungen und Unterstützung finden Sie in der folgenden offiziellen Dokumentation:



- ["kube-state-metrics Dokumentation"](#)
- ["Prometheus-Dokumentation"](#)
- ["Alertmanager-Dokumentation"](#)

Schritt 1: Installieren Sie die Überwachungstools

Um die Ressourcenüberwachung in Trident Protect zu aktivieren, müssen Sie kube-state-metrics, Prometheus und Alertmanager installieren und konfigurieren.

Installieren Sie kube-state-metrics

Sie können kube-state-metrics mit Helm installieren.

Schritte

1. Fügen Sie das kube-state-metrics Helm-Chart hinzu. Beispiel:

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

2. Wenden Sie die Prometheus ServiceMonitor CRD auf den Cluster an:

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

3. Erstellen Sie eine Konfigurationsdatei für das Helm-Chart (zum Beispiel `metrics-config.yaml`). Sie können die folgende Beispielkonfiguration an Ihre Umgebung anpassen:

metrics-config.yaml: kube-state-metrics Helm-Chart-Konfiguration

```
---
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. Installieren Sie kube-state-metrics, indem Sie das Helm-Chart bereitstellen. Beispiel:

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

5. Konfigurieren Sie kube-state-metrics, um Metriken für die von Trident Protect verwendeten benutzerdefinierten Ressourcen zu generieren, indem Sie den Anweisungen in der "[kube-state-metrics Custom Resource Dokumentation](#)" folgen.

Prometheus installieren

Sie können Prometheus installieren, indem Sie den Anweisungen in der "[Prometheus-Dokumentation](#)" folgen.

Installieren Sie Alertmanager

Sie können Alertmanager installieren, indem Sie den Anweisungen in der "[Alertmanager-Dokumentation](#)" folgen.

Schritt 2: Konfigurieren Sie die Überwachungstools so, dass sie zusammenarbeiten

Nach der Installation der Überwachungstools müssen Sie diese so konfigurieren, dass sie zusammenarbeiten.

Schritte

1. Integrieren Sie kube-state-metrics in Prometheus. Bearbeiten Sie die Prometheus-Konfigurationsdatei (`prometheus.yaml`) und fügen Sie die Informationen zum kube-state-metrics-Service hinzu. Beispiel:

prometheus.yaml: Integration des kube-state-metrics service mit 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. Konfigurieren Sie Prometheus so, dass Warnmeldungen an Alertmanager weitergeleitet werden. Bearbeiten Sie die Prometheus Konfigurationsdatei (`prometheus.yaml`) und fügen Sie den folgenden Abschnitt hinzu:

prometheus.yaml: Senden Sie Warnmeldungen an Alertmanager

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

Ergebnis

Prometheus kann nun Metriken von kube-state-metrics erfassen und Warnmeldungen an Alertmanager senden. Sie sind jetzt bereit zu konfigurieren, welche Bedingungen eine Warnmeldung auslösen und wohin die Warnmeldungen gesendet werden sollen.

Schritt 3: Benachrichtigungen und Benachrichtigungsziele konfigurieren

Nachdem Sie die Tools so konfiguriert haben, dass sie zusammenarbeiten, müssen Sie konfigurieren, welche Art von Informationen Warnmeldungen auslösen und wohin die Warnmeldungen gesendet werden sollen.

Warnungsbeispiel: Backup-Fehler

Das folgende Beispiel definiert eine kritische Warnung, die ausgelöst wird, wenn der Status der benutzerdefinierten Backup-Ressource auf `Error` für 5 Sekunden oder länger gesetzt ist. Sie können dieses Beispiel an Ihre Umgebung anpassen und diesen YAML-Ausschnitt in Ihre `prometheus.yaml` Konfigurationsdatei einfügen:

rules.yaml: Definiere eine Prometheus-Alarmierung für fehlgeschlagene Backups

```
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."
```

Konfigurieren Sie Alertmanager, um Benachrichtigungen an andere Kanäle zu senden

Sie können Alertmanager so konfigurieren, dass Benachrichtigungen an andere Kanäle wie E-Mail, PagerDuty, Microsoft Teams oder andere Benachrichtigungsdienste gesendet werden, indem Sie die jeweilige Konfiguration in der `alertmanager.yaml` Datei angeben.

Das folgende Beispiel konfiguriert Alertmanager so, dass Benachrichtigungen an einen Slack-Kanal gesendet werden. Um dieses Beispiel an Ihre Umgebung anzupassen, ersetzen Sie den Wert des `api_url` Schlüssels

durch die in Ihrer Umgebung verwendete Slack-Webhook-URL:

alertmanager.yaml: Senden Sie Warnmeldungen an einen Slack-Kanal

```
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
```

Generieren Sie ein Trident Protect Support-Bundle

Trident Protect ermöglicht Administratoren, Bundles zu erstellen, die Informationen enthalten, die für den NetApp Support nützlich sind, einschließlich Protokollen, Metriken und Topologieinformationen über die verwalteten Cluster und Apps. Wenn Sie mit dem Internet verbunden sind, können Sie Support-Bundles mithilfe einer benutzerdefinierten Ressourcendatei (CR) auf die NetApp Support Site (NSS) hochladen.

Erstellen Sie ein Support-Bundle mithilfe eines CR

Schritte

1. Erstellen Sie die benutzerdefinierte Ressourcendatei (CR) und benennen Sie sie (zum Beispiel `trident-protect-support-bundle.yaml`).
2. Konfigurieren Sie die folgenden Attribute:
 - **metadata.name:** (*Erforderlich*) Der Name dieser benutzerdefinierten Ressource; wählen Sie einen eindeutigen und sinnvollen Namen für Ihre Umgebung.
 - **spec.triggerType:** (*Erforderlich*) Legt fest, ob das Support-Bundle sofort generiert oder geplant wird. Die geplante Bundle-Generierung erfolgt um 12:00 Uhr UTC. Mögliche Werte:
 - Geplant
 - Handbuch
 - **spec.uploadEnabled:** (*Optional*) Steuert, ob das Support-Bundle nach seiner Generierung auf die NetApp Support-Website hochgeladen werden soll. Wenn nicht angegeben, ist der Standardwert `false`. Mögliche Werte:
 - `true`
 - `false` (Standardeinstellung)
 - **spec.dataWindowStart:** (*Optional*) Eine Datumszeichenkette im RFC-3339-Format, die das Datum und die Uhrzeit angibt, zu der das Fenster der im Support-Bundle enthaltenen Daten beginnen soll. Wenn nicht angegeben, wird standardmäßig 24 Stunden zurück angenommen. Das früheste Fensterdatum, das Sie angeben können, liegt 7 Tage zurück.

Beispiel 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. Nachdem Sie die `trident-protect-support-bundle.yaml` Datei mit den korrekten Werten gefüllt haben, wenden Sie die CR an:

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

Erstellen Sie ein Support-Bundle mithilfe der CLI

Schritte

1. Erstellen Sie das Support-Bundle, indem Sie die Werte in Klammern durch Informationen aus Ihrer

Umgebung ersetzen. Das `trigger-type` bestimmt, ob das Bundle sofort erstellt wird oder ob die Erstellungszeit durch den Zeitplan vorgegeben ist, und kann `Manual` oder `Scheduled` sein. Die Standardeinstellung ist `Manual`.

Beispiel:

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

Überwachen und rufen Sie das Support-Paket ab

Nachdem Sie mit einer der beiden Methoden ein Support-Bundle erstellt haben, können Sie den Generierungsfortschritt überwachen und es auf Ihr lokales System abrufen.

Schritte

1. Warten Sie, bis der `status.generationState` den `Completed` Status erreicht. Sie können den Generierungsfortschritt mit dem folgenden Befehl überwachen:

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

2. Rufen Sie das Support-Bundle auf Ihr lokales System ab. Rufen Sie den Kopierbefehl aus dem abgeschlossenen AutoSupport-Bundle ab:

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

Suchen Sie den `kubectl cp` Befehl in der Ausgabe und führen Sie ihn aus, wobei Sie das Argument Ziel durch Ihr bevorzugtes lokales Verzeichnis ersetzen.

Trident Protect aktualisieren

Sie können Trident Protect auf die neueste Version aktualisieren, um neue Funktionen oder Fehlerbehebungen zu nutzen.

- Beim Upgrade von Version 24.10 kann es vorkommen, dass während des Upgrades laufende Snapshots fehlschlagen. Dieses Fehlschlagen verhindert nicht, dass zukünftige Snapshots, egal ob manuell oder geplant, erstellt werden können. Wenn ein Snapshot während des Upgrades fehlschlägt, können Sie manuell einen neuen Snapshot erstellen, um sicherzustellen, dass Ihre Anwendung geschützt ist.



Um mögliche Fehler zu vermeiden, können Sie vor dem Upgrade alle Snapshot-Zeitpläne deaktivieren und sie danach wieder aktivieren. Dies führt jedoch dazu, dass während des Upgrade-Zeitraums geplante Snapshots fehlen.

- Bei Installationen in privaten Registries stellen Sie sicher, dass das erforderliche Helm-Chart und die Images für die Zielversion in Ihrer privaten Registry verfügbar sind, und überprüfen Sie, ob Ihre benutzerdefinierten Helm-Werte mit der neuen Chart-Version kompatibel sind. Weitere Informationen finden sich unter ["Installieren Sie Trident Protect aus einem privaten Registry."](#)

Schritt 1: Wählen Sie eine Version

Trident Protect-Versionen folgen einer datumsbasierten YY.MM Namenskonvention, wobei „YY“ die letzten beiden Ziffern des Jahres und „MM“ der Monat sind. Dot-Releases folgen einer YY.MM.X Konvention, wobei „X“ das Patch-Level ist. Sie wählen die Version aus, auf die Sie aktualisieren möchten, basierend auf der Version, von der Sie aktualisieren.

- Sie können ein direktes Upgrade auf jede Zielversion durchführen, die innerhalb eines Vier-Versions-Fensters Ihrer installierten Version liegt. Beispielsweise können Sie direkt von 24.10 (oder jeder 24.10 dot release) auf 25.10 aktualisieren.
- Wenn Sie von einer Version außerhalb des Vier-Versions-Fensters aktualisieren, ist ein mehrstufiges Upgrade erforderlich. Die Upgrade-Anweisungen für die ["frühere Version"](#) Version, von der Sie aktualisieren, sind zu verwenden, um auf die neueste Version innerhalb des Vier-Versions-Fensters zu aktualisieren. Beispielsweise, wenn Version 24.10 verwendet wird und ein Upgrade auf 26.02 erfolgen soll:
 - a. Erstes Upgrade von 24.10 auf 25.02.
 - b. Führen Sie dann ein Upgrade von 25.02 auf 26.02 durch.

Schritt 2: Trident Protect aktualisieren

Um Trident Protect zu aktualisieren, führen Sie die folgenden Schritte aus.

Schritte

1. Aktualisieren Sie das Trident Helm-Repository:

```
helm repo update
```

2. Aktualisieren Sie die Trident Protect CRDs:



Dieser Schritt ist erforderlich, wenn Sie von einer Version vor 25.06 aktualisieren, da die CRDs jetzt im Trident Protect Helm chart enthalten sind.

- a. Führen Sie diesen Befehl aus, um die Verwaltung von CRDs von `trident-protect-crds` zu `trident-protect` zu verschieben:

```
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. Führen Sie diesen Befehl aus, um das Helm-Secret für das `trident-protect-crds` Chart zu löschen:



Deinstallieren Sie das `trident-protect-crds` Chart nicht mit Helm, da dies Ihre CRDs und alle zugehörigen Daten entfernen könnte.

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

3. Trident Protect aktualisieren:

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



Sie können den Protokollierungsgrad während des Upgrades konfigurieren, indem Sie `--set logLevel=debug` zum Upgrade-Befehl hinzufügen. Der Standard-Protokollierungsgrad ist `warn`. Debug-Protokollierung wird für die Fehlerbehebung empfohlen, da sie NetApp Support hilft, Probleme zu diagnostizieren, ohne dass Änderungen am Protokollierungsgrad oder eine Problemreproduktion erforderlich sind.

Copyright-Informationen

Copyright © 2026 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFT SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGEND EINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.