



# **Administrar Trident Protect**

## **Trident**

NetApp  
January 14, 2026

# Tabla de contenidos

- Administrar Trident Protect ..... 1
  - Administrar la autorización y el control de acceso de Trident Protect ..... 1
    - Ejemplo: Administrar el acceso para dos grupos de usuarios ..... 1
- Supervisar los recursos de Trident Protect ..... 7
  - Paso 1: Instale las herramientas de monitoreo ..... 8
  - Paso 2: Configure las herramientas de monitoreo para que funcionen juntas ..... 10
  - Paso 3: Configure las alertas y los destinos de alertas ..... 11
- Generar un paquete de soporte de Trident Protect ..... 12
- Actualizar Trident Protect ..... 14

# Administrar Trident Protect

## Administrar la autorización y el control de acceso de Trident Protect

Trident Protect utiliza el modelo Kubernetes de control de acceso basado en roles (RBAC). De forma predeterminada, Trident Protect proporciona un único espacio de nombres de sistema y su cuenta de servicio predeterminada asociada. Si tiene una organización con muchos usuarios o necesidades de seguridad específicas, puede utilizar las funciones RBAC de Trident Protect para obtener un control más granular sobre el acceso a los recursos y espacios de nombres.

El administrador de clúster siempre tiene acceso a los recursos del espacio de nombres predeterminado `trident-protect` y también puede acceder a los recursos en el resto de espacios de nombres. Para controlar el acceso a recursos y aplicaciones, es necesario crear espacios de nombres adicionales y agregar recursos y aplicaciones a esos espacios de nombres.

Tenga en cuenta que ningún usuario puede crear CRS de gestión de datos de aplicaciones en el espacio de nombres predeterminado `trident-protect`. Debe crear CRS de gestión de datos de aplicaciones en un espacio de nombres de aplicaciones (como práctica recomendada, crear CRS de gestión de datos de aplicaciones en el mismo espacio de nombres que la aplicación asociada).



Solo los administradores deben tener acceso a los objetos de recursos personalizados privilegiados de Trident Protect, que incluyen:

- **AppVault:** Requiere datos de credenciales de bucket
- **AutoSupportBundle:** Recopila métricas, registros y otros datos confidenciales de Trident Protect
- **BundleSchedule:** Gestiona los horarios de recopilación de registros

Como práctica recomendada, use RBAC para restringir el acceso a los objetos con privilegios a los administradores.

Para obtener más información sobre cómo el RBAC regula el acceso a los recursos y espacios de nombres, consulte la ["Documentación de RBAC de Kubernetes"](#).

Para obtener información sobre las cuentas de servicio, consulte la ["Documentación de la cuenta de servicio de Kubernetes"](#).

### Ejemplo: Administrar el acceso para dos grupos de usuarios

Por ejemplo, una organización tiene un administrador de clústeres, un grupo de usuarios de ingeniería y un grupo de usuarios de marketing. El administrador del clúster debe realizar las siguientes tareas para crear un entorno en el que el grupo de ingeniería y el grupo de marketing tengan acceso solo a los recursos asignados a sus respectivos espacios de nombres.

#### Paso 1: Crear un espacio de nombres para contener recursos para cada grupo

La creación de un espacio de nombres permite separar los recursos de forma lógica y controlar mejor quién tiene acceso a dichos recursos.

## Pasos

1. Cree un espacio de nombres para el grupo de ingeniería:

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

2. Cree un espacio de nombres para el grupo de marketing:

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

## Paso 2: Crear nuevas cuentas de servicio para interactuar con los recursos de cada espacio de nombres

Cada nuevo espacio de nombres que cree viene con una cuenta de servicio predeterminada, pero debe crear una cuenta de servicio para cada grupo de usuarios para que pueda dividir aún más Privileges entre grupos en el futuro si es necesario.

## Pasos

1. Cree una cuenta de servicio para el grupo de ingeniería:

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

2. Cree una cuenta de servicio para el grupo de marketing:

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

## Paso 3: Crear un secreto para cada nueva cuenta de servicio

Un secreto de cuenta de servicio se utiliza para autenticarse con la cuenta de servicio, y se puede eliminar y volver a crear fácilmente si está comprometido.

## Pasos

1. Cree un secreto para la cuenta de servicio de ingeniería:

```

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. Cree un secreto para la cuenta de servicio de 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

```

#### Paso 4: Cree un objeto RoleBinding para enlazar el objeto ClusterRole a cada nueva cuenta de servicio

Se crea un objeto ClusterRole predeterminado cuando instala Trident Protect. Puede vincular este ClusterRole a la cuenta de servicio creando y aplicando un objeto RoleBinding.

##### Pasos

1. Enlazar ClusterRole a la cuenta de servicio de ingeniería:

```

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. Enlazar ClusterRole a la cuenta de servicio de 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
```

## Paso 5: Probar permisos

Compruebe que los permisos son correctos.

### Pasos

1. Confirme que los usuarios de ingeniería pueden acceder a los recursos de ingeniería:

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

2. Confirme que los usuarios de ingeniería no pueden acceder a los recursos de marketing:

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

## Paso 6: Otorgar acceso a los objetos de AppVault

Para realizar tareas de gestión de datos, como backups e instantáneas, el administrador de clúster debe conceder acceso a los objetos de AppVault a usuarios individuales.

### Pasos

1. Cree y aplique un archivo YAML de combinación secreta y AppVault que otorgue acceso a un usuario a un AppVault. Por ejemplo, el siguiente CR otorga acceso a un AppVault al usuario 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. Cree y aplique un CR de rol para permitir que los administradores del cluster concedan acceso a recursos específicos en un espacio de nombres. Por ejemplo:

```

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. Cree y aplique un CR de RoleBinding para enlazar los permisos al usuario eng-user. Por ejemplo:

```

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. Compruebe que los permisos son correctos.

a. Se ha intentado recuperar la información del objeto AppVault para todos los espacios de nombres:

```

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

```

Debería ver una salida similar a la siguiente:



```
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. Prueba para ver si el usuario puede obtener la información de AppVault a la que ahora tiene permiso para acceder:

```
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
```

Debería ver una salida similar a la siguiente:

```
yes
```

## Resultado

Los usuarios a los que ha otorgado permisos de AppVault deben poder usar objetos de AppVault autorizados para operaciones de gestión de datos de aplicaciones y no deben poder acceder a ningún recurso fuera de los espacios de nombres asignados ni crear nuevos recursos a los que no tengan acceso.

## Supervisar los recursos de Trident Protect

Puede utilizar las herramientas de código abierto kube-state-metrics, Prometheus y Alertmanager para supervisar el estado de los recursos protegidos por Trident Protect.

El servicio kube-state-metrics genera métricas a partir de la comunicación con la API de Kubernetes. Usarlo con Trident Protect expone información útil sobre el estado de los recursos en su entorno.

Prometheus es un conjunto de herramientas que puede ingerir los datos generados por kube-state-metrics y presentarlos como información fácilmente legible sobre estos objetos. Juntos, kube-state-metrics y Prometheus le brindan una manera de monitorear la salud y el estado de los recursos que administra con Trident Protect.

Alertmanager es un servicio que ingiere las alertas enviadas por herramientas como Prometheus y las redirige a los destinos que configure.



Las configuraciones y directrices incluidas en estos pasos son solo ejemplos; debe personalizarlas para que se adapten a su entorno. Consulte la siguiente documentación oficial para obtener instrucciones específicas y soporte:

- ["documentación sobre métricas del estado de kube"](#)
- ["Documentación de Prometheus"](#)
- ["Documentación de Alertmanager"](#)

## Paso 1: Instale las herramientas de monitoreo

Para habilitar la supervisión de recursos en Trident Protect, debe instalar y configurar kube-state-metrics, Prometheus y Alertmanager.

### Instale kube-state-metrics

Puede instalar kube-state-metrics usando Helm.

#### Pasos

1. Agregue el gráfico Helm de métricas de estado-kube. Por ejemplo:

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

2. Cree un archivo de configuración para el diagrama Helm (por ejemplo, `metrics-config.yaml`). Puede personalizar la siguiente configuración de ejemplo para que coincida con su entorno:

## Metrics-config.yaml: Configuración del gráfico Helm de métricas de estado de kube

```
---
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
```

3. Instale kube-state-metrics mediante el despliegue del gráfico Helm. Por ejemplo:

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

4. Configure kube-state-metrics para generar métricas para los recursos personalizados utilizados por Trident Protect siguiendo las instrucciones en ["documentación de recursos personalizados de kube-state-metrics"](#)

## Instale Prometheus

Puede instalar Prometheus siguiendo las instrucciones del ["Documentación de Prometheus"](#) .

## Instale Alertmanager

Puede instalar Alertmanager siguiendo las instrucciones del ["Documentación de Alertmanager"](#) .

## Paso 2: Configure las herramientas de monitoreo para que funcionen juntas

Después de instalar las herramientas de supervisión, debe configurarlas para que funcionen juntas.

### Pasos

1. Integre métricas de estado-kube con Prometheus. Edite el archivo de configuración Prometheus (prometheus.yaml) y agregue la información del servicio kube-state-metrics. Por ejemplo:

#### prometheus.yaml: integración del servicio 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. Configurar Prometheus para enrutar alertas a Alertmanager. Edite el archivo de configuración de Prometheus (prometheus.yaml) y agregue la siguiente sección:

## prometheus.yaml: Enviar alertas a Alertmanager

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

### Resultado

Ahora Prometheus puede recopilar métricas de kube-state-metrics y puede enviar alertas a Alertmanager. Ahora está listo para configurar qué condiciones desencadenan una alerta y dónde se deben enviar las alertas.

## Paso 3: Configure las alertas y los destinos de alertas

Después de configurar las herramientas para que funcionen juntas, debe configurar qué tipo de información activa alertas y dónde se deben enviar las alertas.

### Ejemplo de alerta: Fallo de backup

En el siguiente ejemplo se define una alerta crucial que se activa cuando el estado del recurso personalizado de backup se establece en `Error` 5 segundos o más. Puede personalizar este ejemplo para que coincida con su entorno e incluir este fragmento de YAML en su `prometheus.yaml` archivo de configuración:

### rules.yaml: Define una alerta de Prometheus para copias de seguridad fallidas

```
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."
```

## Configure Alertmanager para que envíe alertas a otros canales

Puede configurar Alertmanager para que envíe notificaciones a otros canales, como correo electrónico, PagerDuty, Microsoft Teams u otros servicios de notificación especificando la configuración respectiva en `alertmanager.yaml` el archivo.

El siguiente ejemplo configura Alertmanager para enviar notificaciones a un canal de Slack. Para personalizar este ejemplo a su entorno, reemplace el valor de `api_url` la clave por la URL del webhook de Slack utilizada en su entorno:

### alertmanager.yaml: envía alertas a un canal de 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
```

## Generar un paquete de soporte de Trident Protect

Trident Protect permite a los administradores generar paquetes que incluyen información útil para el soporte de NetApp , incluidos registros, métricas e información de topología sobre los clústeres y las aplicaciones bajo administración. Si está conectado a Internet, puede cargar paquetes de soporte al sitio de soporte de NetApp (NSS) utilizando un archivo de recursos personalizados (CR).

## Cree un paquete de soporte mediante un CR

### Pasos

1. Cree el archivo de recursos personalizados (CR) y asígnele un nombre (por ejemplo, `trident-protect-support-bundle.yaml`).
2. Configure los siguientes atributos:
  - **metadata.name:** (*required*) El nombre de este recurso personalizado; elija un nombre único y sensible para su entorno.
  - **Spec.triggerType:** (*required*) Determina si el paquete de soporte se genera inmediatamente o se programa. La generación de paquetes programada se tiene lugar a LAS 12am UTC. Los posibles valores son los siguientes:
    - Programado
    - Manual
  - **SPEC.uploadEnabled:** (*Opcional*) Controla si el paquete de soporte debe cargarse en el sitio de soporte de NetApp después de que se genere. Si no se especifica, el valor por defecto es `false`. Los posibles valores son los siguientes:
    - verdadero
    - false (predeterminado)
  - **Spec.dataWindowStart:** (*Optional*) Una cadena de fecha en formato RFC 3339 que especifica la fecha y la hora en que debe comenzar la ventana de datos incluidos en el paquete de soporte. Si no se especifica, el valor predeterminado es hace 24 horas. La fecha de ventana más antigua que puede especificar es hace 7 días.

Ejemplo 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. Después de rellenar `astra-support-bundle.yaml` el archivo con los valores correctos, aplique el CR:

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

## Cree un bundle de soporte mediante la CLI

### Pasos

1. Cree el paquete de soporte, reemplazando valores entre paréntesis con la información del entorno.

`trigger-type` Determina si el grupo se crea inmediatamente o si la hora de creación está determinada por la programación, y puede ser `Manual` o `Scheduled`. El valor predeterminado es `Manual`.

Por ejemplo:

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

## Actualizar Trident Protect

Puede actualizar Trident Protect a la última versión para aprovechar nuevas funciones o correcciones de errores.

Para actualizar Trident Protect, realice los siguientes pasos.

### Pasos

1. Actualice el repositorio de Trident Helm:

```
helm repo update
```

2. Actualice los CRD Trident Protect:

```
helm upgrade trident-protect-crds netapp-trident-protect/trident-protect-crds --version 100.2502.0 --namespace trident-protect
```

3. Mejora Trident Protect:

```
helm upgrade trident-protect netapp-trident-protect/trident-protect --version 100.2502.0 --namespace trident-protect
```



## Información de copyright

Copyright © 2026 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPCIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

## Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.