



# **Gerenciar Trident Protect**

## **Trident**

NetApp  
January 14, 2026

# Índice

- Gerenciar Trident Protect ..... 1
  - Gerencie a autorização e o controle de acesso do Trident Protect. .... 1
    - Exemplo: Gerencie o acesso para dois grupos de usuários ..... 1
- Monitorar recursos do Trident Protect ..... 7
  - Passo 1: Instale as ferramentas de monitoramento ..... 8
  - Passo 2: Configure as ferramentas de monitoramento para trabalhar em conjunto ..... 10
  - Etapa 3: Configurar alertas e destinos de alerta ..... 11
- Gere um pacote de suporte Trident Protect ..... 12
- Aprimore o Trident Protect ..... 14

# Gerenciar Trident Protect

## Gerencie a autorização e o controle de acesso do Trident Protect.

O Trident Protect utiliza o modelo Kubernetes de controle de acesso baseado em funções (RBAC). Por padrão, o Trident Protect fornece um único namespace de sistema e sua respectiva conta de serviço padrão. Se a sua organização possui muitos usuários ou necessidades específicas de segurança, você pode usar os recursos RBAC do Trident Protect para obter um controle mais granular sobre o acesso a recursos e namespaces.

O administrador do cluster sempre tem acesso a recursos no namespace padrão `trident-protect` e também pode acessar recursos em todos os outros namespaces. Para controlar o acesso a recursos e aplicações, é necessário criar espaços de nomes adicionais e adicionar recursos e aplicações a esses espaços de nomes.

Observe que nenhum usuário pode criar CRS de gerenciamento de dados do aplicativo no namespace padrão `trident-protect`. Você precisa criar CRS de gerenciamento de dados de aplicativo em um namespace de aplicativo (como prática recomendada, criar CRS de gerenciamento de dados de aplicativo no mesmo namespace que seu aplicativo associado).



Somente os administradores devem ter acesso aos objetos de recursos personalizados privilegiados do Trident Protect, que incluem:

- **AppVault:** Requer dados de credenciais de bucket
- **Pacote de Suporte Automático:** Coleta métricas, registros e outros dados confidenciais do Trident Protect.
- **AutoSupportBundleSchedule:** Gerencia os horários de coleta de Registros

Como prática recomendada, use o RBAC para restringir o acesso a objetos privilegiados aos administradores.

Para obter mais informações sobre como o RBAC regula o acesso a recursos e namespaces, consulte o ["Documentação do Kubernetes RBAC"](#).

Para obter informações sobre contas de serviço, consulte o ["Documentação da conta de serviço do Kubernetes"](#).

### Exemplo: Gerencie o acesso para dois grupos de usuários

Por exemplo, uma organização tem um administrador de cluster, um grupo de usuários de engenharia e um grupo de usuários de marketing. O administrador do cluster concluiria as seguintes tarefas para criar um ambiente onde o grupo de engenharia e o grupo de marketing tenham acesso apenas aos recursos atribuídos aos respectivos namespaces.

#### Etapa 1: Crie um namespace para conter recursos para cada grupo

Criar um namespace permite separar recursos logicamente e controlar melhor quem tem acesso a esses recursos.

## Passos

1. Crie um namespace para o grupo de engenharia:

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

2. Crie um namespace para o grupo de marketing:

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

## Etapa 2: Crie novas contas de serviço para interagir com recursos em cada namespace

Cada novo namespace que você criar vem com uma conta de serviço padrão, mas você deve criar uma conta de serviço para cada grupo de usuários para que você possa dividir ainda mais Privileges entre grupos no futuro, se necessário.

## Passos

1. Crie uma conta de serviço para o grupo de engenharia:

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

2. Crie uma conta de serviço para o grupo de marketing:

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

## Passo 3: Crie um segredo para cada nova conta de serviço

Um segredo de conta de serviço é usado para autenticar com a conta de serviço e pode ser facilmente excluído e recriado se comprometido.

## Passos

1. Crie um segredo para a conta de serviço de engenharia:

```

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. Crie um segredo para a conta do serviço 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

```

#### Passo 4: Crie um objeto RoleBinding para vincular o objeto ClusterRole a cada nova conta de serviço

Um objeto ClusterRole padrão é criado quando você instala o Trident Protect. Você pode vincular essa ClusterRole à conta de serviço criando e aplicando um objeto RoleBinding.

##### Passos

1. Vincule o ClusterRole à conta de serviço de engenharia:

```

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. Vincule o ClusterRole à conta do serviço 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
```

## Passo 5: Testar permissões

Teste se as permissões estão corretas.

### Passos

1. Confirme se os usuários de engenharia podem acessar os recursos de engenharia:

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

2. Confirme que os usuários de engenharia não podem acessar recursos de marketing:

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

## Etapa 6: Conceder acesso a objetos AppVault

Para executar tarefas de gerenciamento de dados, como backups e snapshots, o administrador do cluster precisa conceder acesso a objetos AppVault a usuários individuais.

### Passos

1. Crie e aplique um arquivo YAML de combinação secreta e AppVault que concede a um usuário acesso a um AppVault. Por exemplo, o CR a seguir concede acesso a um AppVault ao usuário 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. Crie e aplique um CR de função para permitir que os administradores de cluster concedam acesso a recursos específicos em um namespace. Por exemplo:

```

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. Criar e aplicar um RoleBinding CR para vincular as permissões ao usuário eng-user. Por exemplo:

```

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. Verifique se as permissões estão corretas.

a. Tente recuperar informações de objeto AppVault para todos os namespaces:

```

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

```

Você deve ver saída semelhante ao seguinte:



```
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. Teste para ver se o usuário pode obter as informações do AppVault que ele agora tem permissão para acessar:

```
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
```

Você deve ver saída semelhante ao seguinte:

```
yes
```

## Resultado

Os usuários aos quais você concedeu permissões AppVault devem poder usar objetos AppVault autorizados para operações de gerenciamento de dados de aplicativos e não devem poder acessar recursos fora dos namespaces atribuídos ou criar novos recursos aos quais eles não têm acesso.

## Monitorar recursos do Trident Protect

Você pode usar as ferramentas de código aberto kube-state-metrics, Prometheus e Alertmanager para monitorar a integridade dos recursos protegidos pelo Trident Protect.

O serviço kube-state-metrics gera métricas a partir da comunicação da API do Kubernetes. Ao utilizá-lo com o Trident Protect, você obtém informações úteis sobre o estado dos recursos em seu ambiente.

O Prometheus é um conjunto de ferramentas que pode ingerir os dados gerados pelo kube-state-metrics e apresentá-los como informações facilmente legíveis sobre esses objetos. Em conjunto, o kube-state-metrics e o Prometheus oferecem uma maneira de monitorar a integridade e o status dos recursos que você está gerenciando com o Trident Protect.

Alertmanager é um serviço que ingere os alertas enviados por ferramentas como Prometheus e os encaminha para destinos que você configura.



As configurações e orientações incluídas nessas etapas são apenas exemplos; você precisa personalizá-las para corresponder ao seu ambiente. Consulte a seguinte documentação oficial para obter instruções e suporte específicos:

- ["documentação de métricas de estado do kube"](#)
- ["Documentação do Prometheus"](#)
- ["Documentação do Alertmanager"](#)

## Passo 1: Instale as ferramentas de monitoramento

Para habilitar o monitoramento de recursos no Trident Protect, você precisa instalar e configurar o kube-state-metrics, o Prometheus e o Alertmanager.

### Instalar métricas de estado do kube

Você pode instalar métricas de estado do kube usando o Helm.

#### Passos

1. Adicione o gráfico Helm de métricas de estado kube. Por exemplo:

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

2. Crie um arquivo de configuração para o gráfico Helm (por exemplo, `metrics-config.yaml`). Você pode personalizar o seguinte exemplo de configuração para corresponder ao seu ambiente:

## Metrics-config.yaml: Configuração do gráfico Helm do kube-State-metrics

```
---
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 as métricas de estado do kube implantando o gráfico Helm. Por exemplo:

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

4. Configure o kube-state-metrics para gerar métricas para os recursos personalizados usados pelo Trident Protect, seguindo as instruções em ["documentação de recursos personalizados de métricas de estado do kube"](#).

## Instale Prometheus

Você pode instalar o Prometheus seguindo as instruções no ["Documentação do Prometheus"](#).

## Instale o Alertmanager

Você pode instalar o Alertmanager seguindo as instruções no ["Documentação do Alertmanager"](#).

## Passo 2: Configure as ferramentas de monitoramento para trabalhar em conjunto

Depois de instalar as ferramentas de monitoramento, você precisa configurá-las para trabalhar em conjunto.

### Passos

1. Integre o kube-State-metrics com Prometheus. Edite o arquivo de configuração Prometheus (`prometheus.yaml`) e adicione as informações do serviço kube-State-metrics. Por exemplo:

#### **prometheus.yaml: integração do serviço kube-state-metrics com o 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. Configure Prometheus para rotear alertas para Alertmanager. Edite o arquivo de configuração Prometheus (`prometheus.yaml`) e adicione a seguinte seção:

## prometheus.yaml: Enviar alertas para o Alertmanager

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

### Resultado

Prometheus agora pode coletar métricas de kube-State-metrics e enviar alertas para Alertmanager. Agora você está pronto para configurar quais condições acionam um alerta e onde os alertas devem ser enviados.

## Etapa 3: Configurar alertas e destinos de alerta

Depois de configurar as ferramentas para trabalhar em conjunto, você precisa configurar que tipo de informação aciona alertas e para onde os alertas devem ser enviados.

### Exemplo de alerta: Falha de backup

O exemplo a seguir define um alerta crítico que é acionado quando o status do recurso personalizado de backup é definido como `Error` por 5 segundos ou mais. Você pode personalizar este exemplo para corresponder ao seu ambiente e incluir esse snippet YAML em seu `prometheus.yaml` arquivo de configuração:

### rules.yaml: Defina um alerta do Prometheus para backups com falha

```
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 o Alertmanager para enviar alertas para outros canais

Você pode configurar o Alertmanager para enviar notificações para outros canais, como e-mail, PagerDuty, Microsoft Teams ou outros serviços de notificação especificando a respectiva configuração no `alertmanager.yaml` arquivo.

O exemplo a seguir configura o Alertmanager para enviar notificações para um canal do Slack. Para personalizar este exemplo para o ambiente, substitua o valor da `api_url` chave pelo URL do webhook do Slack usado no ambiente:

### alertmanager.yaml: Enviar alertas para um canal do 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
```

## Gere um pacote de suporte Trident Protect

O Trident Protect permite que os administradores gerem pacotes que incluem informações úteis para o Suporte da NetApp , como logs, métricas e informações de topologia sobre os clusters e aplicativos gerenciados. Se você estiver conectado à Internet, poderá enviar pacotes de suporte para o site de suporte da NetApp (NSS) usando um arquivo de recurso personalizado (CR).

## Crie um pacote de suporte usando um CR

### Passos

1. Crie o arquivo de recurso personalizado (CR) e nomeie-o (por exemplo, `trident-protect-support-bundle.yaml`).
2. Configure os seguintes atributos:
  - **metadata.name:** (*required*) o nome deste recurso personalizado; escolha um nome único e sensível para o seu ambiente.
  - **Spec.triggerType:** (*required*) determina se o pacote de suporte é gerado imediatamente ou programado. A geração de pacotes programados acontece às 12AM UTC. Valores possíveis:
    - Programado
    - Manual
  - **Spec.uploadEnabled:** (*Optional*) controla se o pacote de suporte deve ser carregado para o site de suporte da NetApp depois que ele é gerado. Se não for especificado, o padrão é `false`. Valores possíveis:
    - verdadeiro
    - falso (padrão)
  - **Spec.dataWindowStart:** (*Optional*) Uma cadeia de caracteres de data no formato RFC 3339 que especifica a data e a hora em que a janela de dados incluídos no pacote de suporte deve começar. Se não for especificado, o padrão é 24 horas atrás. A data da janela mais antiga que você pode especificar é de 7 dias atrás.

Exemplo 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. Depois de preencher o `astra-support-bundle.yaml` ficheiro com os valores corretos, aplique o CR:

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

## Crie um pacote de suporte usando a CLI

### Passos

1. Crie o pacote de suporte, substituindo valores entre parênteses por informações do seu ambiente. O `trigger-type` determina se o pacote é criado imediatamente ou se o tempo de criação é ditado

pelo agendamento e pode ser Manual ou Scheduled. A predefinição é Manual.

Por exemplo:

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

## Aprimore o Trident Protect

Você pode atualizar o Trident Protect para a versão mais recente para aproveitar os novos recursos ou correções de bugs.

Para atualizar o Trident Protect, siga os passos abaixo.

### Passos

1. Atualize o repositório Helm do Trident:

```
helm repo update
```

2. Atualize os CRDs do Trident Protect:

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

3. Aprimore o Trident Protect:

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



## **Informações sobre direitos autorais**

Copyright © 2026 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 DESTES DOCUMENTOS. 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 SOBRESSALIENTES; 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 DESTES DOCUMENTOS, 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.