Instale o Trident Protect

Trident

NetApp
January 15, 2026

Índice

Instale o Trident Protect

Requisitos do Trident Protect

Comece verificando se seu ambiente operacional, clusters de aplicativos, aplicativos e licenças estão prontos. Certifique-se de que seu ambiente atenda a esses requisitos para implantar e operar o Trident Protect.

Compatibilidade do cluster Kubernetes com o Trident Protect

O Trident Protect é compatível com uma ampla gama de ofertas de Kubernetes totalmente gerenciadas e autogerenciadas, incluindo:

- Serviço Amazon Elastic Kubernetes (EKS)
- Google Kubernetes Engine (GKE)
- Serviço Kubernetes do Microsoft Azure (AKS)
- Red Hat OpenShift
- Fazendeiro SUSE
- Portfólio VMware Tanzu
- Kubernetes upstream

- Os backups do Trident Protect são suportados apenas em nós de computação Linux. Os nós de computação do Windows não são suportados para operações de backup.
- Certifique-se de que o cluster no qual você instala o Trident Protect esteja configurado com um controlador de snapshots em execução e os CRDs relacionados. Para instalar um controlador de snapshots, consulte "estas instruções" .

Compatibilidade com o backend de armazenamento Trident Protect

O Trident Protect é compatível com os seguintes sistemas de armazenamento:

- Amazon FSx for NetApp ONTAP
- Cloud Volumes ONTAP
- matrizes de armazenamento ONTAP
- Google Cloud NetApp Volumes
- Azure NetApp Files

Certifique-se de que seu sistema de armazenamento atenda aos seguintes requisitos:

- Certifique-se de que o armazenamento NetApp conectado ao cluster esteja usando o Trident 24.02 ou mais recente (o Trident 24.10 é recomendado).
- Certifique-se de ter um backend de armazenamento NetApp ONTAP .
- Certifique-se de ter configurado um bucket de armazenamento de objetos para armazenar backups.
- Crie todos os namespaces de aplicativos que você planeja usar para aplicativos ou operações de gerenciamento de dados de aplicativos. O Trident Protect não cria esses namespaces para você; se você

especificar um namespace inexistente em um recurso personalizado, a operação falhará.

Requisitos para volumes de economia nas

O Trident Protect oferece suporte a operações de backup e restauração para volumes NAS Economy. Atualmente, não há suporte para snapshots, clones e replicação do SnapMirror para volumes nas-economy. Você precisa habilitar um diretório de snapshots para cada volume nas-economy que planeja usar com o Trident Protect.

> Algumas aplicações não são compatíveis com volumes que utilizam um diretório de instantâneos. Para essas aplicações, você precisa ocultar o diretório de snapshots executando o seguinte comando no sistema de armazenamento ONTAP :
>
> ```
> nfs modify -vserver <svm> -v3-hide-snapshot enabled
> ```

Você pode habilitar o diretório de snapshots executando o seguinte comando para cada volume nas-economy, substituindo `<volume-UUID>` com o UUID do volume que você deseja alterar:

```
tridentctl update volume <volume-UUID> --snapshot-dir=true --pool-level
=true -n trident
```

> Você pode habilitar diretórios de snapshots por padrão para novos volumes definindo a opção de configuração do backend Trident . `snapshotDir` para `true` . Os volumes existentes não serão afetados.

Protegendo dados com VMs KubeVirt

O Trident Protect 24.10 e versões posteriores, incluindo a 24.10.1, apresentam comportamentos diferentes ao proteger aplicativos executados em VMs do KubeVirt. Em ambas as versões, você pode ativar ou desativar o congelamento e descongelamento do sistema de arquivos durante as operações de proteção de dados.

> Durante as operações de restauração, qualquer `VirtualMachineSnapshots` Os dados criados para uma máquina virtual (VM) não são restaurados.

Trident Protect 24.10

O Trident Protect 24.10 não garante automaticamente um estado consistente para os sistemas de arquivos das VMs do KubeVirt durante as operações de proteção de dados. Se você deseja proteger os dados da sua máquina virtual KubeVirt usando o Trident Protect 24.10, precisa habilitar manualmente a funcionalidade de congelamento/descongelamento dos sistemas de arquivos antes da operação de proteção de dados. Isso garante que os sistemas de arquivos estejam em um estado consistente.

Você pode configurar o Trident Protect 24.10 para gerenciar o congelamento e o descongelamento do sistema de arquivos da VM durante as operações de proteção de dados."configurando a virtualização" e então usando o seguinte comando:

```
kubectl set env deployment/trident-protect-controller-manager
NEPTUNE_VM_FREEZE=true -n trident-protect
```

Trident Protect 24.10.1 e versões mais recentes

A partir da versão 24.10.1 do Trident Protect, o sistema congela e descongela automaticamente os sistemas de arquivos do KubeVirt durante as operações de proteção de dados. Opcionalmente, você pode desativar esse comportamento automático usando o seguinte comando:

```
kubectl set env deployment/trident-protect-controller-manager
NEPTUNE_VM_FREEZE=false -n trident-protect
```

Requisitos para replicação do SnapMirror

A replicação NetApp SnapMirror está disponível para uso com o Trident Protect para as seguintes soluções ONTAP :

- Clusters NetApp FAS, AFF e ASA locais
- NetApp ONTAP Select
- NetApp Cloud Volumes ONTAP
- Amazon FSx for NetApp ONTAP

Requisitos do cluster ONTAP para replicação do SnapMirror

Certifique-se de que seu cluster ONTAP atenda aos seguintes requisitos caso planeje usar a replicação SnapMirror :

- * NetApp Trident*: O NetApp Trident deve estar presente nos clusters Kubernetes de origem e destino que utilizam o ONTAP como backend. O Trident Protect oferece suporte à replicação com a tecnologia NetApp SnapMirror usando classes de armazenamento com suporte dos seguintes drivers:
  - `ontap-nas`: NFS
  - `ontap-san`: iSCSI
  - `ontap-san`: FC
  - `ontap-san`: NVMe/TCP (requer versão mínima do ONTAP 9.15.1)
- **Licenças**: As licenças assíncronas do ONTAP SnapMirror que utilizam o pacote Data Protection devem estar habilitadas nos clusters ONTAP de origem e destino. Consulte "Visão geral do licenciamento do SnapMirror no ONTAP" para mais informações.

  A partir do ONTAP 9.10.1, todas as licenças são fornecidas como um arquivo de licença NetApp (NLF), que é um único arquivo que habilita vários recursos. Consulte"Licenças incluídas no ONTAP One" para mais informações.

  Somente a proteção assíncrona SnapMirror é suportada.

Considerações sobre peering para replicação do SnapMirror

Certifique-se de que seu ambiente atenda aos seguintes requisitos caso planeje usar o peering de backend de armazenamento:

- **Cluster e SVM**: Os backends de armazenamento ONTAP devem estar interligados. Consulte "Visão geral do peering de clusters e SVMs" para mais informações.

  Certifique-se de que os nomes SVM usados na relação de replicação entre dois clusters ONTAP sejam únicos.

- * NetApp Trident e SVM*: Os SVMs remotos emparelhados devem estar disponíveis para o NetApp Trident no cluster de destino.
- **Backends gerenciados**: Você precisa adicionar e gerenciar backends de armazenamento ONTAP no Trident Protect para criar uma relação de replicação.

Configuração do Trident / ONTAP para replicação do SnapMirror

O Trident Protect exige que você configure pelo menos um backend de armazenamento que suporte replicação para os clusters de origem e destino. Se os clusters de origem e destino forem os mesmos, o aplicativo de destino deverá usar um backend de armazenamento diferente do aplicativo de origem para obter a melhor resiliência.

Requisitos do cluster Kubernetes para replicação do SnapMirror

Certifique-se de que seus clusters Kubernetes atendam aos seguintes requisitos:

- **Acessibilidade do AppVault**: Tanto o cluster de origem quanto o de destino devem ter acesso à rede para ler e gravar no AppVault para replicação de objetos de aplicativos.
- **Conectividade de rede**: Configure regras de firewall, permissões de bucket e listas de permissão de IP para habilitar a comunicação entre os dois clusters e o AppVault através de WANs.

  Muitos ambientes empresariais implementam políticas de firewall rigorosas em conexões WAN. Verifique esses requisitos de rede com sua equipe de infraestrutura antes de configurar a replicação.

Instale e configure o Trident Protect.

Se o seu ambiente atender aos requisitos do Trident Protect, você pode seguir estas etapas para instalar o Trident Protect em seu cluster. Você pode obter o Trident Protect da NetApp ou instalá-lo a partir do seu próprio registro privado. A instalação a partir de um registro privado é útil caso seu cluster não tenha acesso à Internet.

Instale o Trident Protect

Instale o Trident Protect da NetApp.

Passos

1. Adicione o repositório Trident Helm:

```
helm repo add netapp-trident-protect
https://netapp.github.io/trident-protect-helm-chart
```

2. Use o Helm para instalar o Trident Protect. Substituir `<name-of-cluster>` com um nome de cluster, que será atribuído ao cluster e usado para identificar os backups e snapshots do cluster:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --version 100.2506.0 --create
-namespace --namespace trident-protect
```

Instale o Trident Protect a partir de um registro privado.

Você pode instalar o Trident Protect a partir de um registro de imagens privado caso seu cluster Kubernetes não tenha acesso à Internet. Nestes exemplos, substitua os valores entre colchetes por informações do seu ambiente:

Passos

1. Baixe as seguintes imagens para sua máquina local, atualize as tags e, em seguida, envie-as para seu registro privado:

```
netapp/controller:25.06.0
netapp/restic:25.06.0
netapp/kopia:25.06.0
netapp/trident-autosupport:25.06.0
netapp/exechook:25.06.0
netapp/resourcebackup:25.06.0
netapp/resourcerestore:25.06.0
netapp/resourcedelete:25.06.0
bitnami/kubectl:1.30.2
kubebuilder/kube-rbac-proxy:v0.16.0
```

   Por exemplo:

```
docker pull netapp/controller:25.06.0
```

```
docker tag netapp/controller:25.06.0 <private-registry-
url>/controller:25.06.0
```

```
docker push <private-registry-url>/controller:25.06.0
```

2. Crie o namespace do sistema Trident Protect:

```
kubectl create ns trident-protect
```

3. Faça login no registro:

```
helm registry login <private-registry-url> -u <account-id> -p <api-token>
```

4. Crie um segredo de pull para usar na autenticação de registro privado:

```
kubectl create secret docker-registry regcred --docker-username=<registry-username> --docker-password=<api-token> -n trident-protect --docker-server=<private-registry-url>
```

5. Adicione o repositório Trident Helm:

```
helm repo add netapp-trident-protect https://netapp.github.io/trident-protect-helm-chart
```

6. Crie um arquivo chamado `protectValues.yaml` . Certifique-se de que contenha as seguintes configurações do Trident Protect:

```
---
image:
  registry: <private-registry-url>
imagePullSecrets:
  - name: regcred
controller:
  image:
    registry: <private-registry-url>
rbacProxy:
  image:
    registry: <private-registry-url>
crCleanup:
  imagePullSecrets:
    - name: regcred
webhooksCleanup:
  imagePullSecrets:
    - name: regcred
```

7. Use o Helm para instalar o Trident Protect. Substituir `<name_of_cluster>` com um nome de cluster, que será atribuído ao cluster e usado para identificar os backups e snapshots do cluster:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name_of_cluster> --version 100.2506.0 --create
-namespace --namespace trident-protect -f protectValues.yaml
```

Instale o plugin Trident Protect CLI

Você pode usar o plugin de linha de comando Trident Protect, que é uma extensão do Trident. `tridentctl` Utilitário para criar e interagir com recursos personalizados (CRs) do Trident Protect.

Instale o plugin Trident Protect CLI

Antes de usar o utilitário de linha de comando, você precisa instalá-lo na máquina que utiliza para acessar o cluster. Siga estes passos, dependendo se a sua máquina utiliza uma CPU x64 ou ARM .

Baixe o plugin para CPUs Linux AMD64

Passos

1. Baixe o plugin Trident Protect CLI:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-linux-amd64
```

Baixe o plugin para CPUs Linux ARM64

Passos

1. Baixe o plugin Trident Protect CLI:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-linux-arm64
```

Baixe o plugin para CPUs Mac AMD64

Passos

1. Baixe o plugin Trident Protect CLI:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-macos-amd64
```

Baixe o plugin para CPUs Mac ARM64

Passos

1. Baixe o plugin Trident Protect CLI:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-macos-arm64
```

1. Habilite as permissões de execução para o binário do plugin:

```
chmod +x tridentctl-protect
```

2. Copie o arquivo binário do plugin para um local definido na sua variável PATH. Por exemplo, `/usr/bin` ou `/usr/local/bin` (Você pode precisar de privilégios elevados):

```
cp ./tridentctl-protect /usr/local/bin/
```

3. Opcionalmente, você pode copiar o arquivo binário do plugin para um local em seu diretório pessoal. Nesse caso, recomenda-se garantir que o local faça parte da sua variável PATH:

```
cp ./tridentctl-protect ~/bin/
```

Copiar o plugin para um local na sua variável PATH permite que você o utilize digitando o comando. `tridentctl-protect` ou `tridentctl protect` de qualquer lugar.

Veja a ajuda do plugin Trident CLI

Você pode usar os recursos de ajuda integrados do plugin para obter ajuda detalhada sobre as funcionalidades do plugin:

Passos

1. Utilize a função de ajuda para visualizar as instruções de utilização:

```
tridentctl-protect help
```

Ativar o preenchimento automático de comandos

Após instalar o plugin Trident Protect CLI, você pode ativar o recurso de autocompletar para determinados comandos.

Ative o recurso de autocompletar para o shell Bash.

Passos

1. Baixe o script de conclusão:

```
curl -L -O https://github.com/NetApp/tridentctl-
protect/releases/download/25.06.0/tridentctl-completion.bash
```

2. Crie um novo diretório em seu diretório pessoal para armazenar o script:

```
mkdir -p ~/.bash/completions
```

3. Mova o script baixado para o `~/.bash/completions` diretório:

```
mv tridentctl-completion.bash ~/.bash/completions/
```

4. Adicione a seguinte linha ao `~/.bashrc` Arquivo no seu diretório pessoal:

```
source ~/.bash/completions/tridentctl-completion.bash
```

Ative o recurso de autocompletar para o shell Z.

Passos

1. Baixe o script de conclusão:

```
curl -L -O https://github.com/NetApp/tridentctl-
protect/releases/download/25.06.0/tridentctl-completion.zsh
```

2. Crie um novo diretório em seu diretório pessoal para armazenar o script:

```
mkdir -p ~/.zsh/completions
```

3. Mova o script baixado para o `~/.zsh/completions` diretório:

```
mv tridentctl-completion.zsh ~/.zsh/completions/
```

4. Adicione a seguinte linha ao `~/.zprofile` Arquivo no seu diretório pessoal:

```
source ~/.zsh/completions/tridentctl-completion.zsh
```

Resultado

Na sua próxima sessão de login no shell, você poderá usar o recurso de autocompletar comandos com o plugin tridentctl-protect.

Personalize a instalação do Trident Protect

Você pode personalizar a configuração padrão do Trident Protect para atender aos requisitos específicos do seu ambiente.

Especifique os limites de recursos do contêiner Trident Protect.

Você pode usar um arquivo de configuração para especificar limites de recursos para os contêineres do Trident Protect após a instalação do Trident Protect. A definição de limites de recursos permite controlar a quantidade de recursos do cluster que são consumidos pelas operações do Trident Protect.

Passos

1. Crie um arquivo chamado `resourceLimits.yaml` .
2. Preencha o arquivo com as opções de limite de recursos para os contêineres do Trident Protect de acordo com as necessidades do seu ambiente.

   O seguinte arquivo de configuração de exemplo mostra as configurações disponíveis e contém os valores padrão para cada limite de recurso:

```
---
jobResources:
  defaults:
    limits:
      cpu: 8000m
      memory: 10000Mi
      ephemeralStorage: ""
    requests:
      cpu: 100m
      memory: 100Mi
      ephemeralStorage: ""
  resticVolumeBackup:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  resticVolumeRestore:
    limits:
      cpu: ""
      memory: ""
```

```
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  kopiaVolumeBackup:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  kopiaVolumeRestore:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
```

3. Aplique os valores de `resourceLimits.yaml` arquivo:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f resourceLimits.yaml --reuse-values
```

Personalizar restrições de contexto de segurança

Você pode usar um arquivo de configuração para modificar as restrições de contexto de segurança (SCCs) do OpenShift para contêineres Trident Protect após instalar o Trident Protect. Essas restrições definem as limitações de segurança para os pods em um cluster Red Hat OpenShift.

Passos

1. Crie um arquivo chamado `sccconfig.yaml` .
2. Adicione a opção SCC ao arquivo e modifique os parâmetros de acordo com as necessidades do seu ambiente.

   O exemplo a seguir mostra os valores padrão dos parâmetros para a opção SCC:

```
scc:
  create: true
  name: trident-protect-job
  priority: 1
```

Esta tabela descreve os parâmetros para a opção SCC:

Parâmetro	Descrição	Padrão
criar	Determina se um recurso SCC pode ser criado. Um recurso SCC será criado somente se `scc.create` está definido para `true` e o processo de instalação do Helm identifica um ambiente OpenShift. Se não estiver operando no OpenShift, ou se `scc.create` está definido para `false` Nenhum recurso SCC será criado.	verdadeiro
nome	Especifica o nome do SCC.	tridente-proteger-trabalho
prioridade	Define a prioridade do SCC. Os carcinomas de células escamosas (CCEs) com valores de prioridade mais altos são avaliados antes daqueles com valores mais baixos.	1

3. Aplique os valores de `sccconfig.yaml` arquivo:

```
helm upgrade trident-protect netapp-trident-protect/trident-protect -f
sccconfig.yaml --reuse-values
```

Isso substituirá os valores padrão pelos valores especificados em `sccconfig.yaml` arquivo.

Configure as definições adicionais do gráfico de navegação do Trident Protect.

Você pode personalizar as configurações do AutoSupport e a filtragem de namespace para atender aos seus requisitos específicos. A tabela a seguir descreve os parâmetros de configuração disponíveis:

Parâmetro	Tipo	Descrição
autoSupport.proxy	corda	Configura um URL de proxy para conexões do NetApp AutoSupport . Use isso para rotear uploads de pacotes de suporte por meio de um servidor proxy. Exemplo: `http://my.proxy.url` .

Parâmetro	Tipo	Descrição
autoSupport.inseguro	booleano	Ignora a verificação TLS para conexões de proxy do AutoSupport quando configurado para `true` . Use somente para conexões proxy inseguras. (padrão: `false`)
autoSupport.habilitado	booleano	Ativa ou desativa os uploads diários do pacote Trident Protect AutoSupport . Quando definido para `false` Os uploads diários agendados estão desativados, mas você ainda pode gerar pacotes de suporte manualmente. (padrão: `true`)
restaurarSkipNamespaceAnnotations	corda	Lista separada por vírgulas de anotações de namespace a serem excluídas das operações de backup e restauração. Permite filtrar namespaces com base em anotações.
restaurarIgnorarEtiquetasDeEspaçoDeNomes	corda	Lista separada por vírgulas de rótulos de namespace a serem excluídos das operações de backup e restauração. Permite filtrar namespaces com base em rótulos.

Você pode configurar essas opções usando um arquivo de configuração YAML ou sinalizadores de linha de comando:

Usar arquivo YAML

Passos

1. Crie um arquivo de configuração e dê um nome a ele. `values.yaml` .
2. No arquivo que você criou, adicione as opções de configuração que deseja personalizar.

```
autoSupport:
  enabled: false
  proxy: http://my.proxy.url
  insecure: true
restoreSkipNamespaceAnnotations: "annotation1,annotation2"
restoreSkipNamespaceLabels: "label1,label2"
```

3. Depois de preencher o `values.yaml` Arquivo com os valores corretos, aplique o arquivo de configuração:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f values.yaml --reuse-values
```

Usar sinalizador CLI

Passos

1. Use o seguinte comando com o `--set` Sinalizador para especificar parâmetros individuais:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect \
  --set autoSupport.enabled=false \
  --set autoSupport.proxy=http://my.proxy.url \
  --set restoreSkipNamespaceAnnotations="annotation1,annotation2" \
  --set restoreSkipNamespaceLabels="label1,label2" \
  --reuse-values
```

Restrinja os pods do Trident Protect a nós específicos.

Você pode usar a restrição de seleção de nós `nodeSelector` do Kubernetes para controlar quais de seus nós são elegíveis para executar pods do Trident Protect, com base nos rótulos dos nós. Por padrão, o Trident Protect é restrito a nós que executam Linux. Você pode personalizar ainda mais essas restrições de acordo com suas necessidades.

Passos

1. Crie um arquivo chamado `nodeSelectorConfig.yaml` .
2. Adicione a opção nodeSelector ao arquivo e modifique-o para adicionar ou alterar rótulos de nós, restringindo a seleção de acordo com as necessidades do seu ambiente. Por exemplo, o arquivo a seguir

contém a restrição padrão do sistema operacional, mas também tem como alvo uma região e um nome de aplicativo específicos:

```
nodeSelector:
  kubernetes.io/os: linux
  region: us-west
  app.kubernetes.io/name: mysql
```

3. Aplique os valores de `nodeSelectorConfig.yaml` arquivo:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f nodeSelectorConfig.yaml --reuse-values
```

Isso substitui as restrições padrão pelas que você especificou em `nodeSelectorConfig.yaml` arquivo.

Informações sobre direitos autorais

Informações sobre marcas comerciais