Restaurar aplicativos Kubernetes

NetApp Backup and Recovery

NetApp
February 10, 2026

Índice

Restaurar aplicativos Kubernetes

Restaurar aplicações Kubernetes usando a interface web

O NetApp Backup and Recovery permite restaurar aplicativos que você protegeu com uma política de proteção. Para restaurar um aplicativo, ele precisa ter pelo menos um ponto de restauração disponível. Um ponto de restauração consiste no snapshot local ou no backup no repositório de objetos (ou ambos). Você pode restaurar um aplicativo usando o arquivo local, secundário ou do repositório de objetos.

Antes de começar

Se você estiver restaurando um aplicativo que foi copiado usando Trident Protect, certifique-se de que Trident Protect esteja instalado tanto no cluster de origem quanto no cluster de destino.

Função necessária do NetApp Console

Administrador da organização ou administrador do SnapCenter . "Saiba mais sobre as funções de acesso do NetApp Backup and Recovery" . "Saiba mais sobre as funções de acesso do NetApp Console para todos os serviços" .

Passos

1. No menu NetApp Backup e Recuperação, selecione **Restaurar**.
2. Escolha um aplicativo Kubernetes da lista e selecione **Visualizar e restaurar** para esse aplicativo.

   A lista de pontos de restauração é exibida.

3. Selecione o botão **Restaurar** para o ponto de recuperação que deseja usar.

Configurações gerais

1. Escolha o local de origem do qual restaurar.
2. Escolha o cluster de destino na lista **Cluster**.

   Restaurar um snapshot local criado pelo Trident Protect para um cluster diferente não é suportado no momento.

3. Escolha restaurar nos namespaces originais ou em novos namespaces.
4. Se você optar por restaurar para novos namespaces, insira o namespace ou namespaces de destino a serem usados.
5. Selecione **Avançar**.

Seleção de recursos

1. Escolha se deseja restaurar todos os recursos associados ao aplicativo ou usar um filtro para selecionar recursos específicos para restaurar:

Restaurar todos os recursos

1. Selecione **Restaurar todos os recursos**.
2. Selecione **Avançar**.

Restaurar recursos específicos

1. Selecione **Recursos seletivos**.
2. Escolha o comportamento do filtro de recursos. Se você escolher **Incluir**, os recursos selecionados serão restaurados. Se você escolher **Excluir**, os recursos selecionados não serão restaurados.
3. Selecione **Adicionar regras** para adicionar regras que definem filtros para selecionar recursos. Você precisa de pelo menos uma regra para filtrar recursos.

   Cada regra pode filtrar critérios como namespace do recurso, rótulos, grupo, versão e tipo.

4. Selecione **Salvar** para salvar cada regra.
5. Depois de adicionar todas as regras necessárias, selecione **Pesquisar** para ver os recursos disponíveis no arquivo de backup que correspondem aos seus critérios de filtro.

   Os recursos mostrados são os recursos que existem atualmente no cluster.

6. Quando estiver satisfeito com os resultados, selecione **Avançar**.

Configurações de destino

1. Expanda a seção **Configurações de destino** e escolha restaurar para a classe de armazenamento padrão, para uma classe de armazenamento diferente ou, se estiver restaurando para um cluster diferente, mapear as classes de armazenamento para o cluster de destino.
2. Se você optar por restaurar para uma classe de armazenamento diferente, selecione uma classe de armazenamento de destino que corresponda a cada classe de armazenamento de origem.
3. Opcionalmente, se estiver restaurando um backup ou snapshot criado com Trident Protect, visualize os detalhes do AppVault usado como o bucket de armazenamento para a operação de restauração. Se houver uma alteração no seu ambiente ou no status do AppVault, selecione **Sincronizar App Vault** para atualizar os detalhes.

   Se você precisar criar um AppVault em um cluster Kubernetes para facilitar a restauração de um backup ou snapshot criado usando Trident Protect, consulte "Use objetos do Trident Protect AppVault para gerenciar buckets".

4. Opcionalmente, expanda a seção **Scripts de restauração** e habilite a opção **Pós-script** para escolher um modelo de gancho de execução que será executado após a conclusão da operação de restauração. Se necessário, insira quaisquer argumentos que o script precise e adicione seletores de rótulo para filtrar recursos com base nos rótulos dos recursos.
5. Selecione **Restaurar**.

Restaurar aplicativos Kubernetes usando um recurso personalizado

Você pode usar recursos personalizados para restaurar seus aplicativos a partir de um snapshot ou backup. A restauração a partir de um snapshot existente será mais rápida ao restaurar o aplicativo para o mesmo cluster.

- Ao restaurar um aplicativo, todos os ganchos de execução configurados para o aplicativo são restaurados juntamente com o aplicativo. Se houver um gancho de execução pós-restauração, ele é executado automaticamente como parte da operação de restauração.
- A restauração a partir de um backup para um namespace diferente ou para o namespace original é suportada para volumes qtree. No entanto, a restauração a partir de um snapshot para um namespace diferente ou para o namespace original não é suportada para volumes qtree.
- Você pode usar configurações avançadas para personalizar as operações de restauração. Para saber mais, consulte "Use configurações avançadas de restauração de recursos personalizados".

Restaurar um backup para um namespace diferente

Ao restaurar um backup para um namespace diferente usando uma BackupRestore CR, NetApp Backup and Recovery restaura o aplicativo em um novo namespace e cria uma CR de aplicativo para o aplicativo restaurado. Para proteger o aplicativo restaurado, crie backups ou snapshots sob demanda, ou estabeleça um cronograma de proteção.

- Restaurar um backup para um namespace diferente com recursos existentes não alterará nenhum recurso que compartilhe nomes com aqueles no backup. Para restaurar todos os recursos do backup, exclua e recrie o namespace de destino ou restaure o backup para um novo namespace.
- Ao usar uma CR para restaurar em um novo namespace, você deve criar manualmente o namespace de destino antes de aplicar a CR. NetApp Backup and Recovery cria namespaces automaticamente somente quando se usa a CLI.

Antes de começar

Certifique-se de que o tempo de expiração do token de sessão da AWS seja suficiente para quaisquer operações de restauração do s3 de longa duração. Se o token expirar durante a operação de restauração, a operação pode falhar.

- Consulte o "Documentação da API AWS" para mais informações sobre como verificar a expiração do token de sessão atual.
- Consulte "Documentação do AWS IAM" para obter mais informações sobre credenciais com recursos da AWS.

Ao restaurar backups usando Kopia como o data mover, você pode opcionalmente especificar anotações no CR para controlar o comportamento do armazenamento temporário usado pelo Kopia. Consulte a "Documentação Kopia" para mais informações sobre as opções que você pode configurar.

Passos

1. Crie o arquivo de recurso personalizado (CR) e nomeie-o `trident-protect-backup-restore-cr.yaml`.
2. No arquivo que você criou, configure os seguintes atributos:
   - **metadata.name**: (*Obrigatório*) O nome deste recurso personalizado; escolha um nome único e adequado ao seu ambiente.
   - **spec.appArchivePath**: O caminho dentro do AppVault onde o conteúdo do backup está armazenado. Você pode usar o seguinte comando para encontrar esse caminho:

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o jsonpath
     ='{.status.appArchivePath}'
     ```

   - **spec.appVaultRef**: (*Obrigatório*) O nome do AppVault onde o conteúdo do backup está armazenado.
   - **spec.namespaceMapping**: O mapeamento do namespace de origem da operação de restauração para o namespace de destino. Substitua `my-source-namespace` e `my-destination-namespace` pelas informações do seu ambiente.

     ```
     apiVersion: protect.trident.netapp.io/v1
     kind: BackupRestore
     metadata:
       name: my-cr-name
       namespace: my-destination-namespace
     spec:
       appArchivePath: my-backup-path
       appVaultRef: appvault-name
       namespaceMapping: [{"source": "my-source-namespace", "destination":
     "my-destination-namespace"}]
     ```

3. (*Opcional*) Se precisar selecionar apenas determinados recursos do aplicativo para restaurar, adicione filtros que incluam ou excluam recursos marcados com rótulos específicos:

   > Trident Protect seleciona alguns recursos automaticamente devido à sua relação com os recursos que você seleciona. Por exemplo, se você selecionar um recurso de reivindicação de volume persistente e ele tiver um pod associado, Trident Protect também restaurará o pod associado.

   - **resourceFilter.resourceSelectionCriteria**: (obrigatório para filtragem) Use `Include` ou `Exclude` para incluir ou excluir um recurso definido em resourceMatchers. Adicione os seguintes parâmetros resourceMatchers para definir os recursos a serem incluídos ou excluídos:
     - **resourceFilter.resourceMatchers**: Uma matriz de objetos resourceMatcher. Se você definir vários elementos nesta matriz, eles correspondem como uma operação OR, e os campos dentro de cada elemento (group, kind, version) correspondem como uma operação AND.
       - **resourceMatchers[].group**: (*Opcional*) Grupo do recurso a ser filtrado.
       - **resourceMatchers[].kind**: (*Opcional*) Tipo do recurso a ser filtrado.
       - **resourceMatchers[].version**: (Opcional) Versão do recurso a ser filtrado.

- **resourceMatchers[].names**: (*Opcional*) Nomes no campo metadata.name do Kubernetes do recurso a ser filtrado.
- **resourceMatchers[].namespaces**: (*Opcional*) Namespaces no campo metadata.name do Kubernetes do recurso a ser filtrado.
- **resourceMatchers[].labelSelectors**: (*Opcional*) String seletora de rótulo no campo metadata.name do Kubernetes do recurso, conforme definido no "Documentação do Kubernetes". Por exemplo: `"trident.netapp.io/os=linux"`.

  Por exemplo:

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. Após preencher o `trident-protect-backup-restore-cr.yaml` file com os valores corretos, aplique a CR:

```
kubectl apply -f trident-protect-backup-restore-cr.yaml
```

Restaurar um backup para o namespace original

Você pode restaurar um backup para o namespace original a qualquer momento.

Antes de começar

Certifique-se de que o tempo de expiração do token de sessão da AWS seja suficiente para quaisquer operações de restauração do s3 de longa duração. Se o token expirar durante a operação de restauração, a operação pode falhar.

- Consulte o "Documentação da API AWS" para mais informações sobre como verificar a expiração do token de sessão atual.
- Consulte "Documentação do AWS IAM" para obter mais informações sobre credenciais com recursos da AWS.

Ao restaurar backups usando Kopia como o data mover, você pode opcionalmente especificar anotações no CR para controlar o comportamento do armazenamento temporário usado pelo Kopia. Consulte a "Documentação Kopia" para mais informações sobre as opções que você pode configurar.

Passos

1. Crie o arquivo de recurso personalizado (CR) e nomeie-o `trident-protect-backup-ipr-cr.yaml`.
2. No arquivo que você criou, configure os seguintes atributos:
   - **metadata.name**: (*Obrigatório*) O nome deste recurso personalizado; escolha um nome único e adequado ao seu ambiente.
   - **spec.appArchivePath**: O caminho dentro do AppVault onde o conteúdo do backup está armazenado. Você pode usar o seguinte comando para encontrar esse caminho:

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o jsonpath
     ='{.status.appArchivePath}'
     ```

   - **spec.appVaultRef**: (*Obrigatório*) O nome do AppVault onde o conteúdo do backup está armazenado.

     Por exemplo:

```
apiVersion: protect.trident.netapp.io/v1
kind: BackupInplaceRestore
metadata:
  name: my-cr-name
  namespace: my-app-namespace
spec:
  appArchivePath: my-backup-path
  appVaultRef: appvault-name
```

3. (*Opcional*) Se precisar selecionar apenas determinados recursos do aplicativo para restaurar, adicione filtros que incluam ou excluam recursos marcados com rótulos específicos:

   Trident Protect seleciona alguns recursos automaticamente devido à sua relação com os recursos que você seleciona. Por exemplo, se você selecionar um recurso de reivindicação de volume persistente e ele tiver um pod associado, Trident Protect também restaurará o pod associado.

   - **resourceFilter.resourceSelectionCriteria**: (obrigatório para filtragem) Use `Include` ou `Exclude` para incluir ou excluir um recurso definido em resourceMatchers. Adicione os seguintes parâmetros resourceMatchers para definir os recursos a serem incluídos ou excluídos:
     - **resourceFilter.resourceMatchers**: Uma matriz de objetos resourceMatcher. Se você definir vários elementos nesta matriz, eles correspondem como uma operação OR, e os campos dentro de cada elemento (group, kind, version) correspondem como uma operação AND.
       - **resourceMatchers[].group**: (*Opcional*) Grupo do recurso a ser filtrado.

- **resourceMatchers[].kind**: (*Opcional*) Tipo do recurso a ser filtrado.
- **resourceMatchers[].version**: (Opcional) Versão do recurso a ser filtrado.
- **resourceMatchers[].names**: (*Opcional*) Nomes no campo metadata.name do Kubernetes do recurso a ser filtrado.
- **resourceMatchers[].namespaces**: (*Opcional*) Namespaces no campo metadata.name do Kubernetes do recurso a ser filtrado.
- **resourceMatchers[].labelSelectors**: (*Opcional*) String seletora de rótulo no campo metadata.name do Kubernetes do recurso, conforme definido no "Documentação do Kubernetes". Por exemplo: `"trident.netapp.io/os=linux"`.

  Por exemplo:

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. Após preencher o arquivo `trident-protect-backup-ipr-cr.yaml` com os valores corretos, aplique a CR:

```
kubectl apply -f trident-protect-backup-ipr-cr.yaml
```

Restaurar um backup em um cluster diferente

Você pode restaurar um backup em um cluster diferente se houver um problema com o cluster original.

- Ao restaurar backups usando Kopia como o data mover, você pode opcionalmente especificar anotações no CR para controlar o comportamento do armazenamento temporário usado pelo Kopia. Consulte a "Documentação Kopia" para mais informações sobre as opções que você pode configurar.
- Ao usar uma CR para restaurar em um novo namespace, você deve criar manualmente o namespace de destino antes de aplicar a CR.

Antes de começar

Certifique-se de que os seguintes pré-requisitos sejam atendidos:

- O cluster de destino tem Trident Protect instalado.
- O cluster de destino tem acesso ao caminho do bucket do mesmo AppVault que o cluster de origem, onde o backup está armazenado.
- Certifique-se de que o tempo de expiração do token de sessão da AWS seja suficiente para quaisquer operações de restauração de longa duração. Se o token expirar durante a operação de restauração, a operação pode falhar.
  - Consulte o "Documentação da API AWS" para mais informações sobre como verificar a expiração do token de sessão atual.
  - Consulte "Documentação da AWS" para obter mais informações sobre credenciais com recursos da AWS.

Passos

1. Verifique a disponibilidade do AppVault CR no cluster de destino usando o plugin Trident Protect CLI:

```
tridentctl-protect get appvault --context <destination_cluster_name>
```

   Certifique-se de que o namespace destinado à restauração do aplicativo exista no cluster de destino.

2. Visualize o conteúdo do backup disponível AppVault do cluster de destino:

```
tridentctl-protect get appvaultcontent <appvault_name> \
--show-resources backup \
--show-paths \
--context <destination_cluster_name>
```

   Executar este comando exibe os backups disponíveis no AppVault, incluindo seus clusters de origem, nomes de aplicativos correspondentes, carimbos de data/hora e caminhos de arquivamento.

   Exemplo de saída:

```
+------------+----------+--------+----------------
+-------------------------+-------------+
|   CLUSTER   |    APP    |  TYPE  |      NAME       |        TIMESTAMP
|    PATH     |
+------------+----------+--------+----------------
+-------------------------+-------------+
| production1 | wordpress | backup | wordpress-bkup-1| 2024-10-30
08:37:40 (UTC)| backuppath1 |
| production1 | wordpress | backup | wordpress-bkup-2| 2024-10-30
08:37:40 (UTC)| backuppath2 |
+------------+----------+--------+----------------
+-------------------------+-------------+
```

3. Restaure o aplicativo no cluster de destino usando o nome AppVault e o caminho do arquivo:
4. Crie o arquivo de recurso personalizado (CR) e nomeie-o `trident-protect-backup-restore-cr.yaml`.
5. No arquivo que você criou, configure os seguintes atributos:
   - **metadata.name**: (*Obrigatório*) O nome deste recurso personalizado; escolha um nome único e adequado ao seu ambiente.
   - **spec.appVaultRef**: (*Obrigatório*) O nome do AppVault onde o conteúdo do backup está armazenado.
   - **spec.appArchivePath**: O caminho dentro do AppVault onde o conteúdo do backup está armazenado. Você pode usar o seguinte comando para encontrar esse caminho:

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o jsonpath
     ='{.status.appArchivePath}'
     ```

     Se o BackupRestore CR não estiver disponível, você pode usar o comando mencionado na etapa 2 para visualizar o conteúdo do backup.

   - **spec.namespaceMapping**: O mapeamento do namespace de origem da operação de restauração para o namespace de destino. Substitua `my-source-namespace` e `my-destination-namespace` pelas informações do seu ambiente.

     Por exemplo:

```
apiVersion: protect.trident.netapp.io/v1
kind: BackupRestore
metadata:
  name: my-cr-name
  namespace: my-destination-namespace
spec:
  appVaultRef: appvault-name
  appArchivePath: my-backup-path
  namespaceMapping: [{"source": "my-source-namespace", "destination":
"my-destination-namespace"}]
```

6. Após preencher o `trident-protect-backup-restore-cr.yaml` file com os valores corretos, aplique a CR:

```
kubectl apply -f trident-protect-backup-restore-cr.yaml
```

Restaurar um snapshot para um namespace diferente

Você pode restaurar dados de um snapshot usando um arquivo de recurso personalizado (CR) para um namespace diferente ou para o namespace de origem original. Ao restaurar um snapshot para um namespace diferente usando um SnapshotRestore CR, NetApp Backup and Recovery restaura o aplicativo em um novo namespace e cria um CR de aplicativo para o aplicativo restaurado. Para proteger o aplicativo restaurado, crie backups ou snapshots sob demanda, ou estabeleça um agendamento de proteção.

- SnapshotRestore é compatível com o atributo `spec.storageClassMapping`, mas somente quando as classes de armazenamento de origem e destino usam o mesmo backend de armazenamento. Se você tentar restaurar para uma `StorageClass` que usa um backend de armazenamento diferente, a operação de restauração falhará.
- Ao usar uma CR para restaurar em um novo namespace, você deve criar manualmente o namespace de destino antes de aplicar a CR.

Antes de começar

Certifique-se de que o tempo de expiração do token de sessão da AWS seja suficiente para quaisquer operações de restauração do s3 de longa duração. Se o token expirar durante a operação de restauração, a operação pode falhar.

- Consulte o "Documentação da API AWS" para mais informações sobre como verificar a expiração do token de sessão atual.
- Consulte "Documentação do AWS IAM" para obter mais informações sobre credenciais com recursos da AWS.

Passos

1. Crie o arquivo de recurso personalizado (CR) e nomeie-o `trident-protect-snapshot-restore-cr.yaml`.
2. No arquivo que você criou, configure os seguintes atributos:

- **metadata.name**: (*Obrigatório*) O nome deste recurso personalizado; escolha um nome único e adequado ao seu ambiente.
- **spec.appVaultRef**: (*Obrigatório*) O nome do AppVault onde o conteúdo do snapshot está armazenado.
- **spec.appArchivePath**: O caminho dentro do AppVault onde o conteúdo do snapshot está armazenado. Você pode usar o seguinte comando para encontrar esse caminho:

```
kubectl get snapshots <SNAPHOT_NAME> -n my-app-namespace -o jsonpath
='{.status.appArchivePath}'
```

- **spec.namespaceMapping**: O mapeamento do namespace de origem da operação de restauração para o namespace de destino. Substitua `my-source-namespace` e `my-destination-namespace` pelas informações do seu ambiente.

```
apiVersion: protect.trident.netapp.io/v1
kind: SnapshotRestore
metadata:
  name: my-cr-name
  namespace: my-app-namespace
spec:
  appVaultRef: appvault-name
  appArchivePath: my-snapshot-path
  namespaceMapping: [{"source": "my-source-namespace", "destination":
"my-destination-namespace"}]
```

3. (*Opcional*) Se precisar selecionar apenas determinados recursos do aplicativo para restaurar, adicione filtros que incluam ou excluam recursos marcados com rótulos específicos:

> Trident Protect seleciona alguns recursos automaticamente devido à sua relação com os recursos que você seleciona. Por exemplo, se você selecionar um recurso de reivindicação de volume persistente e ele tiver um pod associado, Trident Protect também restaurará o pod associado.

- **resourceFilter.resourceSelectionCriteria**: (obrigatório para filtragem) Use `Include` ou `Exclude` para incluir ou excluir um recurso definido em resourceMatchers. Adicione os seguintes parâmetros resourceMatchers para definir os recursos a serem incluídos ou excluídos:
  - **resourceFilter.resourceMatchers**: Uma matriz de objetos resourceMatcher. Se você definir vários elementos nesta matriz, eles correspondem como uma operação OR, e os campos dentro de cada elemento (group, kind, version) correspondem como uma operação AND.
    - **resourceMatchers[].group**: (*Opcional*) Grupo do recurso a ser filtrado.
    - **resourceMatchers[].kind**: (*Opcional*) Tipo do recurso a ser filtrado.
    - **resourceMatchers[].version**: (Opcional) Versão do recurso a ser filtrado.
    - **resourceMatchers[].names**: (*Opcional*) Nomes no campo metadata.name do Kubernetes do recurso a ser filtrado.

- **resourceMatchers[].namespaces**: (*Opcional*) Namespaces no campo metadata.name do Kubernetes do recurso a ser filtrado.
- **resourceMatchers[].labelSelectors**: (*Opcional*) String seletora de rótulo no campo metadata.name do Kubernetes do recurso, conforme definido no "Documentação do Kubernetes". Por exemplo: `"trident.netapp.io/os=linux"`.

  Por exemplo:

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. Após preencher o arquivo `trident-protect-snapshot-restore-cr.yaml` com os valores corretos, aplique a CR:

```
kubectl apply -f trident-protect-snapshot-restore-cr.yaml
```

Restaurar um snapshot para o namespace original

Você pode restaurar um snapshot para o namespace original a qualquer momento.

Antes de começar

Certifique-se de que o tempo de expiração do token de sessão da AWS seja suficiente para quaisquer operações de restauração do s3 de longa duração. Se o token expirar durante a operação de restauração, a operação pode falhar.

- Consulte o "Documentação da API AWS" para mais informações sobre como verificar a expiração do token de sessão atual.
- Consulte "Documentação do AWS IAM" para obter mais informações sobre credenciais com recursos da AWS.

Passos

1. Crie o arquivo de recurso personalizado (CR) e nomeie-o `trident-protect-snapshot-ipr-`

`cr.yaml`.

2. No arquivo que você criou, configure os seguintes atributos:
   - **metadata.name**: (*Obrigatório*) O nome deste recurso personalizado; escolha um nome único e adequado ao seu ambiente.
   - **spec.appVaultRef**: (*Obrigatório*) O nome do AppVault onde o conteúdo do snapshot está armazenado.
   - **spec.appArchivePath**: O caminho dentro do AppVault onde o conteúdo do snapshot está armazenado. Você pode usar o seguinte comando para encontrar esse caminho:

```
kubectl get snapshots <SNAPSHOT_NAME> -n my-app-namespace -o
jsonpath='{.status.appArchivePath}'
```

```
apiVersion: protect.trident.netapp.io/v1
kind: SnapshotInplaceRestore
metadata:
  name: my-cr-name
  namespace: my-app-namespace
spec:
  appVaultRef: appvault-name
  appArchivePath: my-snapshot-path
```

3. (*Opcional*) Se precisar selecionar apenas determinados recursos do aplicativo para restaurar, adicione filtros que incluam ou excluam recursos marcados com rótulos específicos:

> Trident Protect seleciona alguns recursos automaticamente devido à sua relação com os recursos que você seleciona. Por exemplo, se você selecionar um recurso de reivindicação de volume persistente e ele tiver um pod associado, Trident Protect também restaurará o pod associado.

   - **resourceFilter.resourceSelectionCriteria**: (obrigatório para filtragem) Use `Include` ou `Exclude` para incluir ou excluir um recurso definido em resourceMatchers. Adicione os seguintes parâmetros resourceMatchers para definir os recursos a serem incluídos ou excluídos:
     - **resourceFilter.resourceMatchers**: Uma matriz de objetos resourceMatcher. Se você definir vários elementos nesta matriz, eles correspondem como uma operação OR, e os campos dentro de cada elemento (group, kind, version) correspondem como uma operação AND.
       - **resourceMatchers[].group**: (*Opcional*) Grupo do recurso a ser filtrado.
       - **resourceMatchers[].kind**: (*Opcional*) Tipo do recurso a ser filtrado.
       - **resourceMatchers[].version**: (Opcional) Versão do recurso a ser filtrado.
       - **resourceMatchers[].names**: (*Opcional*) Nomes no campo metadata.name do Kubernetes do recurso a ser filtrado.
       - **resourceMatchers[].namespaces**: (*Opcional*) Namespaces no campo metadata.name do Kubernetes do recurso a ser filtrado.
       - **resourceMatchers[].labelSelectors**: (*Opcional*) String seletora de rótulo no campo metadata.name do Kubernetes do recurso, conforme definido no "Documentação do

Kubernetes". Por exemplo: `"trident.netapp.io/os=linux"`.

Por exemplo:

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. Após preencher o arquivo `trident-protect-snapshot-ipr-cr.yaml` com os valores corretos, aplique a CR:

```
kubectl apply -f trident-protect-snapshot-ipr-cr.yaml
```

Use configurações avançadas de restauração de recursos personalizados

Você pode personalizar as operações de restauração usando configurações avançadas, como anotações, configurações de namespace e opções de armazenamento para atender às suas necessidades específicas.

Anotações e rótulos de namespace durante operações de restauração e failover

Durante as operações de restauração e failover, os rótulos e anotações no namespace de destino são ajustados para corresponder aos rótulos e anotações no namespace de origem. Rótulos ou anotações do namespace de origem que não existem no namespace de destino são adicionados, e quaisquer rótulos ou anotações já existentes são sobrescritos para corresponder ao valor do namespace de origem. Rótulos ou anotações que existem apenas no namespace de destino permanecem inalterados.

Se você usa Red Hat OpenShift, é importante observar o papel crucial das anotações de namespace em ambientes OpenShift. As anotações de namespace garantem que os pods restaurados sigam as permissões e configurações de segurança apropriadas definidas pelas restrições de contexto de segurança (SCCs) do OpenShift e possam acessar volumes sem problemas de permissão. Para mais informações, consulte o "OpenShift security context constraints documentação".

Você pode impedir que anotações específicas no namespace de destino sejam sobrescritas definindo a variável de ambiente do Kubernetes `RESTORE_SKIP_NAMESPACE_ANNOTATIONS` antes de executar a operação de restauração ou failover. Por exemplo:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect \
  --set-string
restoreSkipNamespaceAnnotations="{<annotation_key_to_skip_1>,<annotation_k
ey_to_skip_2>}" \
  --reuse-values
```

Ao executar uma operação de restauração ou failover, quaisquer anotações e rótulos de namespace especificados em `restoreSkipNamespaceAnnotations` e `restoreSkipNamespaceLabels` são excluídos da operação de restauração ou failover. Certifique-se de que essas configurações sejam definidas durante a instalação inicial do Helm. Para saber mais, consulte "Configurar configurações adicionais do helm chart do Trident Protect".

Se você instalou o aplicativo de origem usando Helm com a `--create-namespace` flag, um tratamento especial é dado à chave de rótulo `name`. Durante o processo de restauração ou failover, Trident Protect copia esse rótulo para o namespace de destino, mas atualiza o valor para o valor do namespace de destino se o valor da origem corresponder ao namespace de origem. Se esse valor não corresponder ao namespace de origem, ele é copiado para o namespace de destino sem alterações.

Exemplo

O exemplo a seguir apresenta um namespace de origem e um de destino, cada um com anotações e rótulos diferentes. Você pode ver o estado do namespace de destino antes e depois da operação, e como as anotações e os rótulos são combinados ou sobrescritos no namespace de destino.

Antes da operação de restauração ou failover

A tabela a seguir ilustra o estado dos namespaces de origem e destino do exemplo antes da operação de restauração ou failover:

Espaço de nomes	Anotações	Etiquetas
Namespace ns-1 (fonte)	• annotation.one/key: "valoratualizado" • anotação.dois/chave: "true"	• ambiente=produção • compliance=hipaa • name=ns-1

Espaço de nomes	Anotações	Etiquetas
Espaço de nomes ns-2 (destino)	• annotation.one/key: "true" • anotação.three/chave: "falso"	• role=database

Após a operação de restauração

A tabela a seguir ilustra o estado do namespace de destino de exemplo após a operação de restauração ou failover. Algumas chaves foram adicionadas, outras foram sobrescritas e o `name` rótulo foi atualizado para corresponder ao namespace de destino:

Espaço de nomes	Anotações	Etiquetas
Espaço de nomes ns-2 (destino)	• annotation.one/key: "valoratualizado" • anotação.dois/chave: "true" • anotação.three/chave: "falso"	• name=ns-2 • compliance=hipaa • ambiente=produção • role=database

Campos suportados

Esta seção descreve os campos adicionais disponíveis para operações de restauração.

Mapeamento de classe de armazenamento

O `spec.storageClassMapping` atributo define um mapeamento de uma classe de armazenamento presente na aplicação de origem para uma nova classe de armazenamento no cluster de destino. Você pode usar isso ao migrar aplicações entre clusters com classes de armazenamento diferentes ou ao alterar o backend de armazenamento para operações de BackupRestore.

Exemplo:

```
storageClassMapping:
  - destination: "destinationStorageClass1"
    source: "sourceStorageClass1"
  - destination: "destinationStorageClass2"
    source: "sourceStorageClass2"
```

Anotações suportadas

Esta seção lista as anotações suportadas para configurar diversos comportamentos no sistema. Se uma anotação não for definida explicitamente pelo usuário, o sistema usará o valor padrão.

Anotação	Tipo	Descrição	Valor padrão
protect.trident.netapp.io/data-mover-timeout-sec	string	O tempo máximo (em segundos) permitido para a operação de movimentação de dados ficar parada.	"300"
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	string	O limite máximo de tamanho (em megabytes) para o cache de conteúdo do Kopia.	"1000"
protect.trident.netapp.io/pvc-bind-timeout-sec	string	Tempo máximo (em segundos) de espera para que qualquer PersistentVolumeClaims (PVC) recém-criado atinja a `Bound` fase antes que a operação falhe. Aplica-se a todos os tipos de CR de restauração (BackupRestore, BackupInplaceRestore, SnapshotRestore, SnapshotInplaceRestore). Use um valor maior se o seu backend de armazenamento ou cluster exigir mais tempo com frequência.	"1200" (20 minutos)

Informações sobre direitos autorais

Informações sobre marcas comerciais