Restaurar aplicativos

Trident

NetApp
February 02, 2026

Índice

Restaurar aplicativos

Restaure aplicativos usando o Trident Protect.

Você pode usar o Trident Protect para restaurar seu aplicativo a partir de um snapshot ou backup. A restauração a partir de um snapshot existente será mais rápida ao restaurar o aplicativo no mesmo cluster.

- Quando você restaura um aplicativo, todos os ganchos de execução configurados para o aplicativo são restaurados com o aplicativo. Se um gancho de execução pós-restauração estiver presente, ele será executado automaticamente como parte da operação de restauração.
- A restauração de um backup para um namespace diferente ou para o namespace original é suportada para volumes qtree. No entanto, a restauração 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 "Utilize as configurações avançadas de restauração do Trident Protect.".

Restaure de um backup para um namespace diferente

Ao restaurar um backup para um namespace diferente usando um CR de BackupRestore, o Trident Protect restaura o aplicativo em um novo namespace e cria um CR de aplicativo para o aplicativo restaurado. Para proteger a aplicação restaurada, 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 no backup, exclua e recrie o namespace de destino ou restaure o backup para um novo namespace.
- Ao usar uma solicitação de restauração (CR) para restaurar em um novo namespace, você deve criar manualmente o namespace de destino antes de aplicar a CR. O Trident Protect cria namespaces automaticamente apenas quando se utiliza a CLI (linha de comando).

Antes de começar

Certifique-se de que a expiração do token de sessão da AWS seja suficiente para quaisquer operações de restauração S3 de longa duração. Se o token expirar durante a operação de restauração, a operação pode falhar.

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

Ao restaurar backups usando o Kopia como o movimentador de dados, você pode, opcionalmente, especificar anotações no CR ou usar a CLI para controlar o comportamento do armazenamento temporário usado pelo Kopia. Consulte o "Documentação da Kopia" Para obter mais informações sobre as opções que você pode configurar. Use o `tridentctl-protect create --help` Para obter mais informações sobre como especificar anotações com a CLI do Trident Protect, consulte o comando.

Use um CR

Passos

1. Crie o arquivo de recurso personalizado (CR) e nomeie-o `trident-protect-backup-restore-cr.yaml`.
2. No arquivo criado, configure os seguintes atributos:
   - **metadata.name**: (*required*) o nome deste recurso personalizado; escolha um nome único e sensível para o seu ambiente.
   - **Spec.appArchivePath**: O caminho dentro do AppVault onde o conteúdo do backup é armazenado. Você pode usar o seguinte comando para encontrar este caminho:

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

   - **Spec.appVaultRef**: (*required*) o nome do AppVault onde o conteúdo de backup é 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` por 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 você precisar selecionar apenas determinados recursos do aplicativo para restaurar, adicione filtragem que inclua ou exclua recursos marcados com rótulos específicos:

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

   - **ResourceFilter.resourceSelectionCriteria**: (Necessário para filtragem) Use `Include` ou `Exclude` inclua ou exclua um recurso definido em resourceMatchers. Adicione os seguintes parâmetros resourceMatchers para definir os recursos a serem incluídos ou excluídos:
     - **ResourceFilter.resourceMatchers**: Uma matriz de resourceMatcher objetos. Se você definir vários elementos nesse array, eles corresponderão como uma OPERAÇÃO OU, e os campos dentro de cada elemento (grupo, tipo, versão) corresponderão como uma OPERAÇÃO E.

- **ResourceMatchers[].group**: (*Optional*) Grupo do recurso a ser filtrado.
- **ResourceMatchers[].kind**: (*Opcional*) tipo do recurso a ser filtrado.
- **ResourceMatchers[].version**: (*Optional*) versão do recurso a ser filtrado.
- **ResourceMatchers[].names**: (*Optional*) nomes no campo Kubernetes metadata.name do recurso a ser filtrado.
- **ResourceMatchers[].namespaces**: (*Optional*) namespaces no campo Kubernetes metadata.name do recurso a ser filtrado.
- **ResourceMatchers[].labelSelectors**: (*Optional*) string de seleção de etiquetas no campo Kubernetes metadata.name 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. Depois de preencher o `trident-protect-backup-restore-cr.yaml` ficheiro com os valores corretos, aplique o CR:

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

Use a CLI

Passos

1. Restaure o backup para um namespace diferente, substituindo valores entre parênteses por informações do seu ambiente. O `namespace-mapping` argumento usa namespaces separados por dois pontos para mapear namespaces de origem para os namespaces de destino corretos no formato `source1:dest1,source2:dest2`. Por exemplo:

```
tridentctl-protect create backuprestore <my_restore_name> \
--backup <backup_namespace>/<backup_to_restore> \
--namespace-mapping <source_to_destination_namespace_mapping> \
-n <application_namespace>
```

Restaure de 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 a expiração do token de sessão da AWS seja suficiente para quaisquer operações de restauração S3 de longa duração. Se o token expirar durante a operação de restauração, a operação pode falhar.

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

> Ao restaurar backups usando o Kopia como o movimentador de dados, você pode, opcionalmente, especificar anotações no CR ou usar a CLI para controlar o comportamento do armazenamento temporário usado pelo Kopia. Consulte o "Documentação da Kopia" Para obter mais informações sobre as opções que você pode configurar. Use o `tridentctl-protect create --help` Para obter mais informações sobre como especificar anotações com a CLI do Trident Protect, consulte o comando.

Use um CR

Passos

1. Crie o arquivo de recurso personalizado (CR) e nomeie-o `trident-protect-backup-ipr-cr.yaml`.
2. No arquivo criado, configure os seguintes atributos:
   - **metadata.name**: (*required*) o nome deste recurso personalizado; escolha um nome único e sensível para o seu ambiente.
   - **Spec.appArchivePath**: O caminho dentro do AppVault onde o conteúdo do backup é armazenado. Você pode usar o seguinte comando para encontrar este caminho:

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

   - **Spec.appVaultRef**: (*required*) o nome do AppVault onde o conteúdo de backup é 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 você precisar selecionar apenas determinados recursos do aplicativo para restaurar, adicione filtragem que inclua ou exclua recursos marcados com rótulos específicos:

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

   - **ResourceFilter.resourceSelectionCriteria**: (Necessário para filtragem) Use `Include` ou `Exclude` inclua ou exclua um recurso definido em resourceMatchers. Adicione os seguintes parâmetros resourceMatchers para definir os recursos a serem incluídos ou excluídos:
     - **ResourceFilter.resourceMatchers**: Uma matriz de resourceMatcher objetos. Se você definir vários elementos nesse array, eles corresponderão como uma OPERAÇÃO OU, e os campos dentro de cada elemento (grupo, tipo, versão) corresponderão como uma OPERAÇÃO E.
       - **ResourceMatchers[].group**: (*Optional*) Grupo do recurso a ser filtrado.
       - **ResourceMatchers[].kind**: (*Opcional*) tipo do recurso a ser filtrado.

- **ResourceMatchers[].version**: (*Optional*) versão do recurso a ser filtrado.
- **ResourceMatchers[].names**: (*Optional*) nomes no campo Kubernetes metadata.name do recurso a ser filtrado.
- **ResourceMatchers[].namespaces**: (*Optional*) namespaces no campo Kubernetes metadata.name do recurso a ser filtrado.
- **ResourceMatchers[].labelSelectors**: (*Optional*) string de seleção de etiquetas no campo Kubernetes metadata.name 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. Depois de preencher o `trident-protect-backup-ipr-cr.yaml` ficheiro com os valores corretos, aplique o CR:

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

Use a CLI

Passos

1. Restaure o backup para o namespace original, substituindo valores entre parênteses por informações do seu ambiente. O `backup` argumento usa um namespace e um nome de backup no formato `<namespace>/<name>`. Por exemplo:

```
tridentctl-protect create backupinplacerestore <my_restore_name> \
--backup <namespace/backup_to_restore> \
-n <application_namespace>
```

Restaure de um backup para um cluster diferente

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

- Ao restaurar backups usando o Kopia como o movimentador de dados, você pode, opcionalmente, especificar anotações no CR ou usar a CLI para controlar o comportamento do armazenamento temporário usado pelo Kopia. Consulte o "Documentação da Kopia" Para obter mais informações sobre as opções que você pode configurar. Use o `tridentctl-protect create --help` Para obter mais informações sobre como especificar anotações com a CLI do Trident Protect, consulte o comando.
- Ao usar uma solicitação de restauração (CR) para restaurar em um novo namespace, você deve criar manualmente o namespace de destino antes de aplicar a CR. O Trident Protect cria namespaces automaticamente apenas quando se utiliza a CLI (linha de comando).

Antes de começar

Certifique-se de que os seguintes pré-requisitos são cumpridos:

- O cluster de destino tem o Trident Protect instalado.
- O cluster de destino tem acesso ao caminho do bucket do mesmo AppVault que o cluster de origem, onde o backup é armazenado.
- Ao executar o AppVault CR, certifique-se de que seu ambiente local possa se conectar ao bucket de armazenamento de objetos definido nele. `tridentctl-protect get appvaultcontent` comando. Caso as restrições de rede impeçam o acesso, execute a CLI do Trident Protect a partir de um pod no cluster de destino.
- Certifique-se de que a 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 a "Documentação da API da AWS" para obter mais informações sobre como verificar a expiração do token de sessão atual.
  - Consulte o "Documentação do 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>
```

   Verifique se o namespace destinado à restauração do aplicativo existe no cluster de destino.

2. Veja o conteúdo de backup do AppVault disponível no cluster de destino:

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

Executar esse comando exibe os backups disponíveis no AppVault, incluindo os 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 para o cluster de destino usando o nome do AppVault e o caminho do arquivo:

Use um CR

1. Crie o arquivo de recurso personalizado (CR) e nomeie-o `trident-protect-backup-restore-cr.yaml`.
2. No arquivo criado, configure os seguintes atributos:
   - **metadata.name**: (*required*) o nome deste recurso personalizado; escolha um nome único e sensível para o seu ambiente.
   - **Spec.appVaultRef**: (*required*) o nome do AppVault onde o conteúdo de backup é armazenado.
   - **Spec.appArchivePath**: O caminho dentro do AppVault onde o conteúdo do backup é armazenado. Você pode usar o seguinte comando para encontrar este caminho:

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

     Se o BackupRestore CR não estiver disponível, você poderá 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` por 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"}]
```

3. Depois de preencher o `trident-protect-backup-restore-cr.yaml` ficheiro com os valores corretos, aplique o CR:

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

Use a CLI

1. Use o comando a seguir para restaurar o aplicativo, substituindo valores entre parênteses por informações do ambiente. O argumento namespace-mapping usa namespaces separados por dois pontos para mapear namespaces de origem para os namespaces de destino corretos no formato

source1:dest1,source2:dest2. Por exemplo:

```
tridentctl-protect create backuprestore <restore_name> \
--namespace-mapping <source_to_destination_namespace_mapping> \
--appvault <appvault_name> \
--path <backup_path> \
--context <destination_cluster_name> \
-n <application_namespace>
```

Restauração de 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 CR de SnapshotRestore, o Trident Protect restaura o aplicativo em um novo namespace e cria um CR de aplicativo para o aplicativo restaurado. Para proteger a aplicação restaurada, crie backups ou snapshots sob demanda, ou estabeleça um cronograma de proteção.

- O SnapshotRestore oferece suporte ao `spec.storageClassMapping` atributo, mas somente quando as classes de armazenamento de origem e destino usam o mesmo backend de armazenamento. Se você tentar restaurar para um `StorageClass` que usa um backend de armazenamento diferente, a operação de restauração falhará.
- Ao usar uma solicitação de restauração (CR) para restaurar em um novo namespace, você deve criar manualmente o namespace de destino antes de aplicar a CR. O Trident Protect cria namespaces automaticamente apenas quando se utiliza a CLI (linha de comando).

Antes de começar

Certifique-se de que a expiração do token de sessão da AWS seja suficiente para quaisquer operações de restauração S3 de longa duração. Se o token expirar durante a operação de restauração, a operação pode falhar.

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

Use um CR

Passos

1. Crie o arquivo de recurso personalizado (CR) e nomeie-o `trident-protect-snapshot-restore-cr.yaml`.
2. No arquivo criado, configure os seguintes atributos:
   - **metadata.name**: (*required*) o nome deste recurso personalizado; escolha um nome único e sensível para o seu ambiente.
   - **Spec.appVaultRef**: (*required*) o nome do AppVault onde o conteúdo do instantâneo é armazenado.
   - **Spec.appArchivePath**: O caminho dentro do AppVault onde o conteúdo do snapshot é armazenado. Você pode usar o seguinte comando para encontrar este 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` por 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 você precisar selecionar apenas determinados recursos do aplicativo para restaurar, adicione filtragem que inclua ou exclua recursos marcados com rótulos específicos:

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

   - **ResourceFilter.resourceSelectionCriteria**: (Necessário para filtragem) Use `Include` ou `Exclude` inclua ou exclua um recurso definido em resourceMatchers. Adicione os seguintes parâmetros resourceMatchers para definir os recursos a serem incluídos ou excluídos:
     - **ResourceFilter.resourceMatchers**: Uma matriz de resourceMatcher objetos. Se você definir vários elementos nesse array, eles corresponderão como uma OPERAÇÃO OU, e os campos

dentro de cada elemento (grupo, tipo, versão) corresponderão como uma OPERAÇÃO E.

- **ResourceMatchers[].group**: (*Optional*) Grupo do recurso a ser filtrado.
- **ResourceMatchers[].kind**: (*Opcional*) tipo do recurso a ser filtrado.
- **ResourceMatchers[].version**: (*Optional*) versão do recurso a ser filtrado.
- **ResourceMatchers[].names**: (*Optional*) nomes no campo Kubernetes metadata.name do recurso a ser filtrado.
- **ResourceMatchers[].namespaces**: (*Optional*) namespaces no campo Kubernetes metadata.name do recurso a ser filtrado.
- **ResourceMatchers[].labelSelectors**: (*Optional*) string de seleção de etiquetas no campo Kubernetes metadata.name 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. Depois de preencher o `trident-protect-snapshot-restore-cr.yaml` ficheiro com os valores corretos, aplique o CR:

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

Use a CLI

Passos

1. Restaure o snapshot para um namespace diferente, substituindo valores entre parênteses por informações do seu ambiente.

   - O `snapshot` argumento usa um namespace e um nome instantâneo no formato `<namespace>/<name>`.
   - O `namespace-mapping` argumento usa namespaces separados por dois pontos para mapear

namespaces de origem para os namespaces de destino corretos no formato `source1:dest1,source2:dest2`.

Por exemplo:

```
tridentctl-protect create snapshotrestore <my_restore_name> \
--snapshot <namespace/snapshot_to_restore> \
--namespace-mapping <source_to_destination_namespace_mapping> \
-n <application_namespace>
```

Restauração de 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 a expiração do token de sessão da AWS seja suficiente para quaisquer operações de restauração S3 de longa duração. Se o token expirar durante a operação de restauração, a operação pode falhar.

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

Use um CR

Passos

1. Crie o arquivo de recurso personalizado (CR) e nomeie-o `trident-protect-snapshot-ipr-cr.yaml`.
2. No arquivo criado, configure os seguintes atributos:
   - **metadata.name**: (*required*) o nome deste recurso personalizado; escolha um nome único e sensível para o seu ambiente.
   - **Spec.appVaultRef**: (*required*) o nome do AppVault onde o conteúdo do instantâneo é armazenado.
   - **Spec.appArchivePath**: O caminho dentro do AppVault onde o conteúdo do snapshot é armazenado. Você pode usar o seguinte comando para encontrar este 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 você precisar selecionar apenas determinados recursos do aplicativo para restaurar, adicione filtragem que inclua ou exclua recursos marcados com rótulos específicos:

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

   - **ResourceFilter.resourceSelectionCriteria**: (Necessário para filtragem) Use `Include` ou `Exclude` inclua ou exclua um recurso definido em resourceMatchers. Adicione os seguintes parâmetros resourceMatchers para definir os recursos a serem incluídos ou excluídos:
     - **ResourceFilter.resourceMatchers**: Uma matriz de resourceMatcher objetos. Se você definir vários elementos nesse array, eles corresponderão como uma OPERAÇÃO OU, e os campos dentro de cada elemento (grupo, tipo, versão) corresponderão como uma OPERAÇÃO E.
       - **ResourceMatchers[].group**: (*Optional*) Grupo do recurso a ser filtrado.
       - **ResourceMatchers[].kind**: (*Opcional*) tipo do recurso a ser filtrado.
       - **ResourceMatchers[].version**: (*Optional*) versão do recurso a ser filtrado.

- **ResourceMatchers[].names**: (*Optional*) nomes no campo Kubernetes metadata.name do recurso a ser filtrado.
- **ResourceMatchers[].namespaces**: (*Optional*) namespaces no campo Kubernetes metadata.name do recurso a ser filtrado.
- **ResourceMatchers[].labelSelectors**: (*Optional*) string de seleção de etiquetas no campo Kubernetes metadata.name 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. Depois de preencher o `trident-protect-snapshot-ipr-cr.yaml` ficheiro com os valores corretos, aplique o CR:

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

Use a CLI

Passos

1. Restaure o snapshot para o namespace original, substituindo valores entre parênteses por informações do seu ambiente. Por exemplo:

```
tridentctl-protect create snapshotinplacerestore <my_restore_name> \
--snapshot <namespace/snapshot_to_restore> \
-n <application_namespace>
```

Verifique o status de uma operação de restauração

Você pode usar a linha de comando para verificar o status de uma operação de restauração que está em andamento, concluiu ou falhou.

Passos

1. Use o seguinte comando para recuperar o status da operação de restauração, substituindo valores em brackes por informações do seu ambiente:

```
kubectl get backuprestore -n <namespace_name> <my_restore_cr_name> -o
jsonpath='{.status}'
```

Utilize as configurações avançadas de restauração do Trident Protect.

Você pode personalizar operações de restauração usando configurações avançadas, como anotações, configurações de namespace e opções de armazenamento para atender aos seus requisitos específicos.

Anotações e rótulos de namespace durante operações de restauração e failover

Durante as operações de restauração e failover, rótulos e anotações no namespace de destino são feitos 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 que já existem 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 o Red Hat OpenShift, é importante observar o papel crítico das anotações de namespace em ambientes OpenShift. As anotações de namespace garantem que os pods restaurados cumpram 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"Documentação de restrições de contexto de segurança OpenShift" .

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

Se você instalou o aplicativo de origem usando o Helm com o `--create-namespace` bandeira, tratamento especial é dado ao `name` Legenda da etiqueta. Durante o processo de restauração ou failover, o 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 será copiado para o namespace de destino sem alterações.

Exemplo

O exemplo a seguir apresenta um namespace de origem e 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 rótulos são combinados ou substituídos no namespace de destino.

Antes da operação de restauração ou failover

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

Namespace	Anotações	Etiquetas
Namespace ns-1 (fonte)	• annotation.one/key: "updatedvalue" • annotation.two/key: "true"	• ambiente de produção • conformidade hipaa • nome: ns-1
Namespace ns-2 (destino)	• annotation.one/key: "true" (verdadeiro) • annotation.three/key: "false"	• banco de dados

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, algumas foram sobrescritas e o `name` rótulo foi atualizado para corresponder ao namespace de destino:

Namespace	Anotações	Etiquetas
Namespace ns-2 (destino)	• annotation.one/key: "updatedvalue" • annotation.two/key: "true" • annotation.three/key: "false"	• nome: ns-2 • conformidade hipaa • ambiente de produção • banco de dados

Campos suportados

Esta seção descreve campos adicionais disponíveis para operações de restauração.

Mapeamento de classes de armazenamento

O `spec.storageClassMapping` atributo define um mapeamento de uma classe de armazenamento presente no aplicativo de origem para uma nova classe de armazenamento no cluster de destino. Você pode usar isso ao migrar aplicativos entre clusters com diferentes classes de armazenamento 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	cadeia de carateres	O tempo máximo (em segundos) permitido para a operação do movimentador de dados ser interrompida.	"300"
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	cadeia de carateres	O limite máximo de tamanho (em megabytes) para o cache de conteúdo do Kopia.	"1000"
protect.trident.netapp.io/pvc-bind-timeout-sec	cadeia de carateres	Tempo máximo (em segundos) de espera para que quaisquer PersistentVolumeClaims (PVCs) recém-criados cheguem ao `Bound` fase anterior à falha das operações. Aplica-se a todos os tipos de restauração de CR (BackupRestore, BackupInplaceRestore, SnapshotRestore, SnapshotInplaceRestore). Use um valor mais alto se o seu sistema de armazenamento ou cluster frequentemente exigir mais tempo.	"1200" (20 minutos)

Informações sobre direitos autorais

Informações sobre marcas comerciais