Restaurar aplicativos

Trident

NetApp
July 01, 2026

Índice

Restaurar aplicativos . . . 1
- Restaure aplicativos usando Trident Protect . . . 1
  - Restaurar a partir de um backup para um namespace diferente . . . 1
  - Restaurar de um backup para o namespace original . . . 5
  - Restaurar de um backup para um cluster diferente . . . 8
  - Restaurar a partir de um Snapshot para um namespace diferente . . . 11
  - Restaurar de um Snapshot para o namespace original . . . 14
  - Verifique o status de uma operação de restauração . . . 17
- Use as configurações avançadas de restauração do Trident Protect . . . 17
  - Anotações e rótulos de namespace durante operações de restauração e failover . . . 17
  - Campos suportados . . . 19
  - Anotações suportadas . . . 19

Restaurar aplicativos

Restaure aplicativos usando Trident Protect

Você pode usar Trident Protect para restaurar seu aplicativo a partir de um snapshot ou backup. Restaurar a partir de um snapshot existente será mais rápido 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 as configurações avançadas de restauração do Trident Protect".

Restaurar a partir de um backup para um namespace diferente

Ao restaurar um backup para um namespace diferente usando um BackupRestore CR, Trident Protect 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.

- 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 para um novo namespace, você deve criar manualmente o namespace de destino antes de aplicar a CR. Trident Protect cria namespaces automaticamente apenas 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 do AWS API" 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 ou usando a CLI para controlar o comportamento do armazenamento temporário usado pelo Kopia. Consulte o "Documentação Kopia" para mais informações sobre as opções que você pode configurar. Use o comando `tridentctl-protect create --help` para mais informações sobre como especificar anotações com a Trident Protect CLI.

Use um CR

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.destinationApplicationName**: (*Opcional*) O nome para o aplicativo restaurado. Se fornecido, o aplicativo restaurado usa este nome. Se não for fornecido, o aplicativo restaurado usa o nome do aplicativo de origem.
   - **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
       destinationApplicationName: my-new-app-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 arquivo `trident-protect-backup-restore-cr.yaml` com os valores corretos, aplique a CR:

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

Use o CLI

Passos

1. Restaure o backup para um namespace diferente, substituindo os valores entre colchetes pelas informações do seu ambiente. O `namespace-mapping` argumento usa namespaces separados por

dois pontos para mapear os 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> \
--destination-app-name<custom_app_name>\
-n <application_namespace>
```

Restaurar de um backup para o namespace original

Você pode restaurar um backup para o namespace original a qualquer momento. Ao realizar uma restauração no local, Trident Protect gerencia automaticamente os agendamentos de proteção e as operações em andamento para evitar pontos de recuperação inválidos:

- Todos os agendamentos de proteção ativados para o aplicativo são desativados antes do início da restauração. Isso impede que backups ou snapshots agendados sejam executados enquanto os recursos do aplicativo estão sendo restaurados.
- Após a restauração ser concluída com sucesso, somente os agendamentos que estavam ativados antes da restauração são reativados. Os agendamentos que já estavam desativados permanecem desativados.
- Quaisquer operações de backup ou snapshot em andamento são canceladas antes do início da restauração. Se uma operação não for cancelada em 5 minutos, a restauração prossegue e registra um aviso no status do CR de restauração.

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 do AWS API" 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 ou usando a CLI para controlar o comportamento do armazenamento temporário usado pelo Kopia. Consulte o "Documentação Kopia" para mais informações sobre as opções que você pode configurar. Use o comando `tridentctl-protect create --help` para mais informações sobre como especificar anotações com a Trident Protect CLI.

Use um CR

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
```

Use o CLI

Passos

1. Restaure o backup para o namespace original, substituindo os valores entre colchetes pelas 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>
```

Restaurar de um backup para 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 ou usando a CLI para controlar o comportamento do armazenamento temporário usado pelo Kopia. Consulte o "Documentação Kopia" para mais informações sobre as opções que você pode configurar. Use o comando `tridentctl-protect create --help` para mais informações sobre como especificar anotações com a Trident Protect CLI.
- Ao usar uma CR para restaurar para um novo namespace, você deve criar manualmente o namespace de destino antes de aplicar a CR. Trident Protect cria namespaces automaticamente apenas quando se usa a CLI.

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 seu ambiente local possa se conectar ao bucket de storage de objetos definido no AppVault CR ao executar o comando `tridentctl-protect get appvaultcontent`. Se as restrições de rede impedirem o acesso, execute o Trident Protect CLI a partir de um pod no cluster de destino.
- 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 do AWS API" 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 se o AppVault CR existe no cluster de destino usando o plugin Trident Protect CLI:

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

Se o AppVault CR não existir no cluster de destino, crie-o seguindo os passos em "Use objetos do Trident Protect AppVault para gerenciar buckets".

2. Visualize o conteúdo do backup disponível do AppVault no cluster de destino e anote `appArchivePath` do backup que deseja restaurar:

```
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:

> Ao usar uma CR, certifique-se de que o namespace destinado à restauração do aplicativo exista no cluster de destino.

Use um CR

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.appVaultRef**: (*Obrigatório*) O nome do AppVault onde o conteúdo do backup está armazenado.
   - **spec.appArchivePath**: (*Obrigatório*) O caminho dentro do AppVault onde o conteúdo do backup está armazenado. Use o comando da etapa 2 para visualizar o conteúdo do backup e encontrar `appArchivePath` o backup que deseja restaurar.
   - **spec.destinationApplicationName**: (*Opcional*) O nome para o aplicativo restaurado. Se fornecido, o aplicativo restaurado usa este nome. Se não for fornecido, o aplicativo restaurado usa o nome do aplicativo de origem.
   - **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
  destinationApplicationName: my-new-app-name
  namespaceMapping: [{"source": "my-source-namespace", "
destination": "my-destination-namespace"}]
```

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

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

Use o CLI

1. Use o seguinte comando para restaurar o aplicativo, substituindo os valores entre colchetes pelas informações do seu ambiente. O argumento namespace-mapping usa namespaces separados por dois pontos para mapear os 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> \
--destination-app-name <custom_app_name> \
--context <destination_cluster_name> \
-n <application_namespace>
```

Restaurar a partir 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 SnapshotRestore CR, Trident Protect 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 para um novo namespace, você deve criar manualmente o namespace de destino antes de aplicar a CR. Trident Protect cria namespaces automaticamente apenas 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 do AWS API" 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.

Use um CR

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.destinationApplicationName**: (*Opcional*) O nome para o aplicativo restaurado. Se fornecido, o aplicativo restaurado usa este nome. Se não for fornecido, o aplicativo restaurado usa o nome do aplicativo de origem.
   - **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
```

Use o CLI

Passos

1. Restaure o snapshot para um namespace diferente, substituindo os valores entre colchetes pelas informações do seu ambiente.

- O `snapshot` argumento usa um namespace e um nome de snapshot no formato `<namespace>/<name>`.
- O `namespace-mapping` argumento usa namespaces separados por dois pontos para mapear os 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> \
--destination-app-name <custom_app_name> \
-n <application_namespace>
```

Restaurar de um Snapshot para o namespace original

Você pode restaurar um snapshot para o namespace original a qualquer momento. Ao realizar uma restauração no local, Trident Protect gerencia automaticamente os agendamentos de proteção e as operações em andamento para evitar pontos de recuperação inválidos:

- Todos os agendamentos de proteção ativados para o aplicativo são desativados antes do início da restauração. Isso impede que backups ou snapshots agendados sejam executados enquanto os recursos do aplicativo estão sendo restaurados.
- Após a restauração ser concluída com sucesso, somente os agendamentos que estavam ativados antes da restauração são reativados. Os agendamentos que já estavam desativados permanecem desativados.
- Quaisquer operações de backup ou snapshot em andamento são canceladas antes do início da restauração. Se uma operação não for cancelada em 5 minutos, a restauração prossegue e registra um aviso no status do CR de restauração.

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 do AWS API" 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.

Use um CR

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 o CLI

Passos

1. Restaure o snapshot para o namespace original, substituindo os valores entre colchetes pelas 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, foi concluída ou falhou.

Passos

1. Use o seguinte comando para recuperar o status da operação de restauração, substituindo os valores entre colchetes pelas informações do seu ambiente:

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

Use as configurações avançadas de restauração do Trident Protect

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 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