保护 **Kubernetes** 工作负载（预览版）

NetApp Backup and Recovery

NetApp
February 23, 2026

This PDF was generated from https://docs.netapp.com/zh-cn/data-services-backup-recovery/br-use-kubernetes-protect-overview.html on February 23, 2026. Always check docs.netapp.com for the latest.

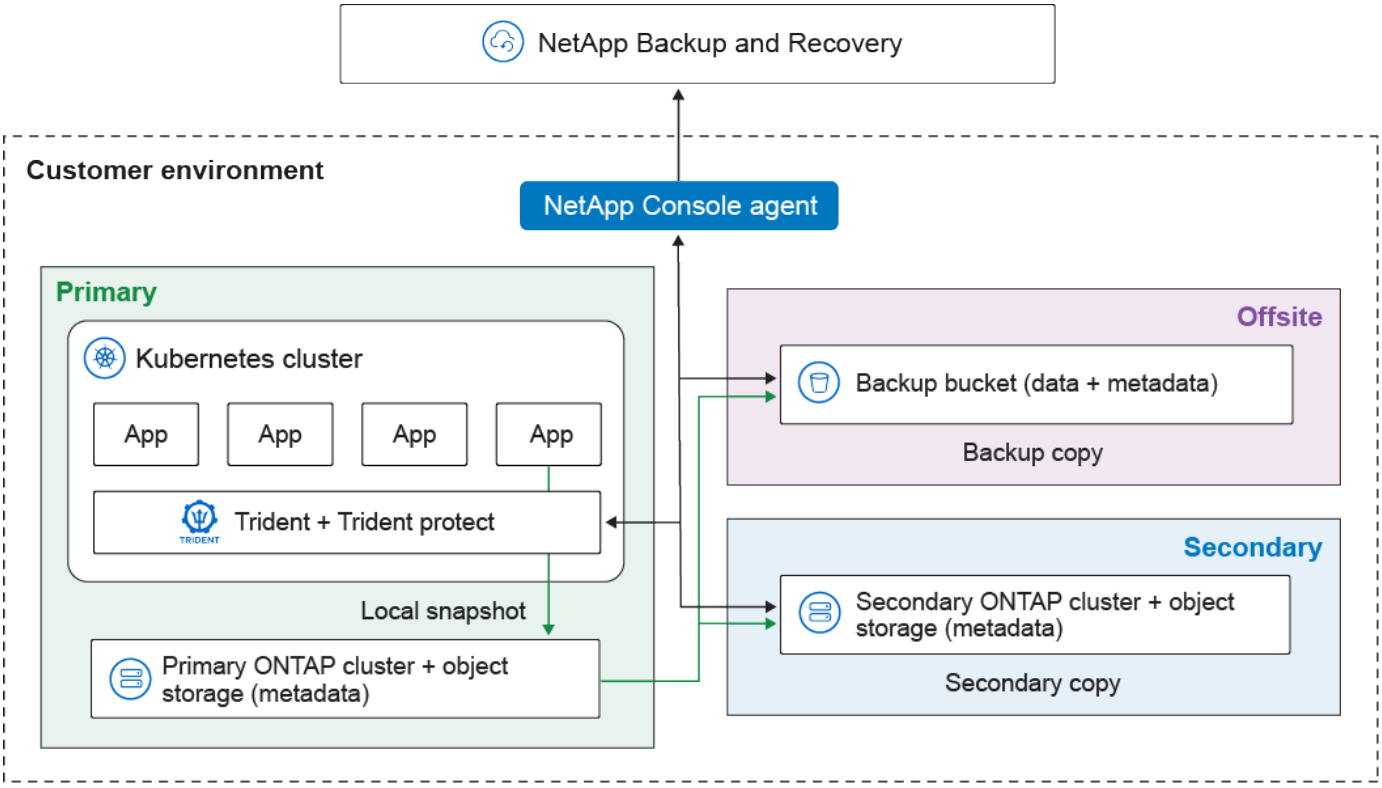
目录

保护 Kubernetes 工作负载（预览版）

管理 Kubernetes 工作负载概览

通过在NetApp Backup and Recovery中管理 Kubernetes 工作负载，您可以在一个地方发现、管理和保护您的 Kubernetes 集群和应用程序。您可以管理托管在 Kubernetes 集群上的资源和应用程序。您还可以创建保护策略并将其与 Kubernetes 工作负载关联，所有这些都使用单一界面完成。

下图展示了 Kubernetes 工作负载的备份和恢复的组件和基本架构，以及如何将数据的不同副本存储在不同位置：



NetApp Backup and Recovery为管理 Kubernetes 工作负载提供了以下优势：

- 单一控制平面，用于保护跨多个 Kubernetes 集群运行的应用程序。这些应用程序可以包括在 Kubernetes 集群上运行的容器或虚拟机。
- 与NetApp SnapMirror本机集成，为所有备份和恢复工作流程提供存储卸载功能。
- Kubernetes 应用程序的永久增量备份，转化为更低的恢复点目标 (RPO) 和恢复时间目标 (RTO)。

本文档作为技术预览提供。在预览期间，不建议将 Kubernetes 功能用于生产工作负载。对于此预览版产品， NetApp保留在正式发布之前修改产品详细信息、内容和时间表的权利。

您可以完成与管理 Kubernetes 工作负载相关的以下任务：

- "发现 Kubernetes 工作负载"。

- "管理 Kubernetes 集群"。
- "添加和保护 Kubernetes 应用程序"。
- "管理 Kubernetes 应用程序"。
- "恢复 Kubernetes 应用程序"。

探索NetApp Backup and Recovery中的 Kubernetes 工作负载

NetApp Backup and Recovery需要在保护 Kubernetes 工作负载之前发现它们。

所需的**NetApp Console**角色 备份和恢复超级管理员。了解详情"备份和恢复角色和权限"。 "了解所有服务的NetApp Console访问角色" 。

发现 **Kubernetes** 工作负载

在备份和恢复清单中，发现您环境中的 Kubernetes 工作负载。添加工作负载会将 Kubernetes 集群添加到NetApp Backup and Recovery。然后，您可以添加应用程序并保护集群资源。

> 当您发现当前受 Trident Protect 保护的群集时，在发现过程中将禁用与 Trident Protect 一起使用的任何备份计划（Trident Protect 备份计划与 Backup and Recovery 不兼容）。要保护群集的应用程序，"创建新的保护策略"或将应用程序与现有策略关联。然后，如果需要，您可以删除 Trident Protect 备份计划。

步骤

1. 执行以下操作之一：
   - 如果您是第一次发现 Kubernetes 工作负载，请在NetApp Backup and Recovery中，在“工作负载”下，选择“Kubernetes”磁贴。
   - 如果您已经发现 Kubernetes 工作负载，请在NetApp Backup and Recovery中选择 **Inventory** > **Workloads**，然后选择 **Discover resources**。
2. 选择 **Kubernetes** 工作负载类型。
3. 输入集群名称并选择与集群一起使用的连接器。
4. 按照出现的命令行说明进行操作：
   - 创建 Trident Protect 命名空间
   - 创建 Kubernetes 机密
   - 添加 Helm 存储库
   - 安装或升级 Trident Protect 和 Trident Protect 连接器

     这些步骤确保NetApp Backup and Recovery可以与集群交互。
5. 完成这些步骤后，选择*发现*。

   该集群已添加到清单中。
6. 在关联的 Kubernetes 工作负载中选择“查看”以查看该工作负载的应用程序、集群和命名空间列表。

继续访问**NetApp Backup and Recovery**仪表板

按照以下步骤查看NetApp Backup and Recovery仪表板。

1. 从NetApp Console菜单中，选择 保护 > 备份和恢复。
2. 选择一个工作负载图块（例如，Microsoft SQL Server）。
3. 从备份和恢复菜单中，选择*仪表板*。
4. 审查数据保护的健康状况。处于危险中或受保护的工作负载的数量会根据新发现、受保护和备份的工作负载而增加。

   "了解仪表板显示的内容"。

添加和保护 **Kubernetes** 应用程序

添加和保护 **Kubernetes** 应用程序

NetApp Backup and Recovery使您能够轻松发现 Kubernetes 集群，而无需生成和上传 kubeconfig 文件。您可以使用从NetApp Console用户界面复制的简单命令连接 Kubernetes 集群并安装所需的软件。

所需的**NetApp Console**角色

组织管理员或SnapCenter管理员。"了解NetApp Backup and Recovery访问角色" 。 "了解所有服务的NetApp Console访问角色" 。

添加并保护新的 **Kubernetes** 应用程序

保护 Kubernetes 应用程序的第一步是在NetApp Backup and Recovery中创建应用程序。创建应用程序时，您会让控制台了解 Kubernetes 集群上正在运行的应用程序。

开始之前

在添加和保护 Kubernetes 应用程序之前，您需要"发现 Kubernetes 工作负载"。

使用 **Web UI** 添加应用程序

步骤

1. 在NetApp Backup and Recovery中，选择 **Inventory**。
2. 选择一个 Kubernetes 实例，然后选择“查看”以查看与该实例关联的资源。
3. 选择“应用程序”选项卡。
4. 选择*创建应用程序*。
5. 输入应用程序的名称。
6. 或者，选择以下任意字段来搜索您想要保护的资源：
   - 关联集群
   - 关联的命名空间
   - 资源类型
   - 标签选择器
7. （可选）选择“集群范围资源”以选择任何在集群级别范围限定的资源。如果包含这些资源，它们会在创建应用程序时添加到应用程序中。
8. 或者，选择“搜索”以根据您的搜索条件查找资源。

   控制台不存储搜索参数或结果；这些参数用于在选定的 Kubernetes 集群中搜索可包含在应用程序中的资源。

9. 控制台显示符合您的搜索条件的资源列表。
10. 如果列表包含您想要保护的资源，请选择“下一步”。
11. （可选）在“策略”区域中，选择现有保护策略来保护应用程序，或者创建新策略。如果不选择策略，则创建的应用程序将不带保护策略。你可以"添加保护策略"之后。
12. 在*Prescripts and postscripts*区域中，启用并配置您想要在备份操作之前或之后运行的任何prescript或postscript执行挂钩。要启用处方或附言，您必须至少已创建了一个"执行钩子模板"。
13. 选择“创建”。

结果

应用程序已创建并出现在 Kubernetes 清单的 应用程序 选项卡中的应用程序列表中。 NetApp Console根据您的设置启用对应用程序的保护，并且您可以在备份和恢复的*监控*区域中监控进度。

使用 **CR** 添加应用程序

步骤

1. 创建目标应用程序 CR 文件：
   a. 创建自定义资源 (CR) 文件并将其命名（例如， `my-app-name.yaml`）。
   b. 配置以下属性：
      - **metadata.name**：（必需）应用程序自定义资源的名称。请注意您选择的名称，因为保护操作所需的其他 CR 文件会引用此值。
      - **spec.includedNamespaces**: (*Required*) 使用命名空间和标签选择器指定应用程序使用的命名

空间和资源。应用程序命名空间必须是此列表的一部分。标签选择器是可选的，可用于筛选每个指定命名空间内的资源。

- **spec.includedClusterScopedResources**: (*Optional*) 使用此属性指定要包含在应用程序定义中的群集范围的资源。此属性允许您根据其组、版本、种类和标签选择这些资源。
  - **groupVersionKind**：（必需）指定集群范围内资源的 API 组、版本和种类。
  - **labelSelector**：（可选）根据集群范围资源的标签对其进行筛选。

c. 如果需要，请配置以下注释：

- **metadata.annotations.protect.trident.netapp.io/skip-vm-freeze**：（*Optional*）此批注仅适用于从虚拟机定义的应用程序，例如在 KubeVirt 环境中，快照之前会发生文件系统冻结。指定此应用程序是否可以在快照期间写入文件系统。如果设置为 true，应用程序将忽略全局设置，并且可以在快照期间写入文件系统。如果设置为 false，应用程序将忽略全局设置，并在快照期间冻结文件系统。如果指定，但应用程序在应用程序定义中没有虚拟机，则忽略批注。如果未指定，则应用程序遵循 "全局文件系统冻结设置"。
- **protect.trident.netapp.io/protection-command**：（可选）使用此注释指示 Backup and Recovery 保护或停止保护应用程序。可能的值为 `protect`或 `unprotect`。
- **protect.trident.netapp.io/protection-policy-name**: (可选) 使用此注释指定要用于保护此应用程序的 Backup and Recovery 保护策略的名称。此保护策略必须已存在于 Backup and Recovery 中。

如果需要在已创建应用程序后应用此批注，可以使用以下命令：

```
kubectl annotate application -n <application CR namespace> <application CR name> protect.trident.netapp.io/skip-vm-freeze="true"
```

+
示例 YAML：

+

```
apiVersion: protect.trident.netapp.io/v1
kind: Application
metadata:
  annotations:
    protect.trident.netapp.io/skip-vm-freeze: "false"
    protect.trident.netapp.io/protection-command: "protect"
    protect.trident.netapp.io/protection-policy-name: "policy-name"
  name: my-app-name
  namespace: my-app-namespace
spec:
  includedNamespaces:
    - namespace: namespace-1
      labelSelector:
        matchLabels:
          app: example-app
    - namespace: namespace-2
      labelSelector:
        matchLabels:
          app: another-example-app
  includedClusterScopedResources:
    - groupVersionKind:
        group: rbac.authorization.k8s.io
        kind: ClusterRole
        version: v1
      labelSelector:
        matchLabels:
          mylabel: test
```

1. （*Optional*）添加包含或排除标有特定标签的资源的筛选：

   - **resourceFilter.resourceSelectionCriteria**：（筛选时需要）使用 `Include`或 `Exclude`来包含或排除在 resourceMatchers 中定义的资源。添加以下 resourceMatchers 参数以定义要包括或排除的资源：

     - **resourceFilter.resourceMatchers**：resourceMatcher 对象数组。如果在此数组中定义多个元素，则它们将作为 OR 操作进行匹配，并且每个元素（组、种类、版本）内的字段将作为 AND 操作进行匹配。

       - **resourceMatchers[].group**：（*Optional*）要筛选的资源的组。
       - **resourceMatchers[].kind**：（*Optional*）要筛选的资源的类型。
       - **resourceMatchers[].version**：（*Optional*）要筛选的资源的版本。

- **resourceMatchers[].names**：（可选）要过滤的资源的 Kubernetes metadata.name 字段中的名称。
- **resourceMatchers[].namespaces**: (*Optional*) 要过滤的资源的 Kubernetes metadata.name 字段中的命名空间。
- **resourceMatchers[].labelSelectors**: (*Optional*) 资源的 Kubernetes metadata.name 字段中的标签选择器字符串，如 "Kubernetes 文档" 中所定义。例如："trident.netapp.io/os=linux"。

> 当同时使用 `resourceFilter`和 `labelSelector`时， `resourceFilter`首先运行，然后将 `labelSelector`应用于生成的资源。

例如：

```
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"]
```

2. 创建与环境匹配的应用程序 CR 后，应用 CR。例如：

```
kubectl apply -f my-app-name.yaml
```

立即使用 **Backup and Recovery Web UI** 备份 **Kubernetes** 应用程序

NetApp Backup and Recovery 使您能够使用 Web 界面手动备份 Kubernetes 应用程序。

所需的**NetApp Console**角色

组织管理员或SnapCenter管理员。"了解NetApp Backup and Recovery访问角色" 。 "了解所有服务的NetApp Console访问角色" 。

立即使用 **Web UI** 备份 **Kubernetes** 应用程序

手动创建 Kubernetes 应用程序的备份，为未来的备份和快照建立基线，或确保最新数据受到保护。

步骤

1. 在NetApp Backup and Recovery中，选择 **Inventory**。
2. 选择一个 Kubernetes 实例，然后选择“查看”以查看与该实例关联的资源。
3. 选择“应用程序”选项卡。
4. 在应用程序列表中，选择要备份的应用程序并选择相关的操作菜单。
5. 选择*立即备份*。
6. 确保选择了正确的应用程序名称。
7. 选择*备份*。

结果

控制台创建应用程序的备份并在备份和恢复的*监控*区域中显示进度。该备份是根据与应用程序关联的保护策略创建的。

现在使用 **Backup and Recovery** 中的自定义资源备份 **Kubernetes** 应用程序

NetApp Backup and Recovery 使您能够使用自定义资源 (CR) 手动备份 Kubernetes 应用程序。

现在使用自定义资源备份 **Kubernetes** 应用程序

手动创建 Kubernetes 应用程序的备份，为未来的备份和快照建立基线，或确保最新数据受到保护。

如果集群范围的资源在应用程序定义中显式引用，或者它们引用了任何应用程序命名空间，则会包含在备份、快照或克隆中。

开始之前

确保 AWS 会话令牌过期时间足以支持任何长时间运行的 s3 备份操作。如果令牌在备份操作期间过期，操作可能会失败。

- 有关检查当前会话令牌过期的详细信息，请参见 "AWS API 文档"。
- 有关 AWS 资源凭据的详细信息，请参见 "AWS IAM 文档"。

使用自定义资源创建本地快照

要创建 Kubernetes 应用程序的快照并将其存储在本地，请使用具有特定属性的 Snapshot 自定义资源。

步骤

1. 创建自定义资源 (CR) 文件并将其命名为 `local-snapshot-cr.yaml`。
2. 在创建的文件中，配置以下属性：
    - **metadata.name**：（*Required*）此自定义资源的名称；为您的环境选择一个唯一且合理的名称。

- **spec.applicationRef**：要快照的应用程序的 Kubernetes 名称。
- **spec.appVaultRef**：（必需）应存储快照内容（元数据）的 AppVault 的名称。
- **spec.reclaimPolicy**：（可选）定义删除快照 CR 时快照的 AppArchive 会发生什么情况。这意味着即使设置为 `Retain`，快照也将被删除。有效选项：
  - `Retain` (默认)
  - `Delete`

```
apiVersion: protect.trident.netapp.io/v1
kind: Snapshot
metadata:
  namespace: my-app-namespace
  name: local-snapshot-cr
spec:
  applicationRef: my-application
  appVaultRef: appvault-name
  reclaimPolicy: Retain
```

3. 使用正确的值填充 `local-snapshot-cr.yaml`文件后，应用 CR：

```
kubectl apply -f local-snapshot-cr.yaml
```

使用自定义资源将应用程序备份到对象存储

创建具有特定属性的 Backup CR，以将应用程序备份到对象存储。

步骤

1. 创建自定义资源 (CR) 文件并将其命名为 `object-store-backup-cr.yaml`。
2. 在创建的文件中，配置以下属性：
   - **metadata.name**：（*Required*）此自定义资源的名称；为您的环境选择一个唯一且合理的名称。
   - **spec.applicationRef**：（必需）要备份的应用程序的 Kubernetes 名称。
   - **spec.appVaultRef**：（必需，与 *spec.appVaultTargetsRef* 互斥）如果使用相同的存储桶存储快照和备份，则这是应存储备份内容的 AppVault 的名称。
   - **spec.appVaultTargetsRef**：（必需，与 *spec.appVaultRef* 互斥）如果您使用不同的存储桶来存储快照和备份，这是应存储备份内容的 AppVault 的名称。
   - **spec.dataMover**：（*Optional*）一个字符串，指示要用于备份操作的备份工具。该值区分大小写，必须为 `CBS`。
   - **spec.reclaimPolicy**：（可选）定义删除 Backup CR 时备份内容（元数据/卷数据）会发生什么。可能的值：
     - `Delete`

- `Retain` (默认)

- **spec.cleanupSnapshot**：（必需）确保备份 CR 创建的临时快照在备份操作完成后不被删除。建议值：`false`。

使用同一存储桶存储快照和备份时的示例 YAML：

```
apiVersion: protect.trident.netapp.io/v1
kind: Backup
metadata:
  namespace: my-app-namespace
  name: my-cr-name
spec:
  applicationRef: my-application
  appVaultRef: appvault-name
  dataMover: CBS
  reclaimPolicy: Retain
  cleanupSnapshot: false
```

使用不同存储桶存储快照和备份时的示例 YAML：

```
apiVersion: protect.trident.netapp.io/v1
kind: Backup
metadata:
  namespace: my-app-namespace
  name: object-store-backup-cr
spec:
  applicationRef: my-application
  appVaultTargetsRef: appvault-targets-name
  dataMover: CBS
  reclaimPolicy: Retain
  cleanupSnapshot: false
```

3. 使用正确的值填充 `object-store-backup-cr.yaml`文件后，应用 CR：

```
kubectl apply -f object-store-backup-cr.yaml
```

使用自定义资源创建 3-2-1 扇出备份

使用 3-2-1 扇出架构进行备份会将备份复制到辅助存储和对象存储。要创建 3-2-1 扇出备份，请创建具有特定属性的 Backup CR。

步骤

1. 创建自定义资源 (CR) 文件并将其命名为 `3-2-1-fanout-backup-cr.yaml`。

2. 在创建的文件中，配置以下属性：

   - **metadata.name**：（*Required*）此自定义资源的名称；为您的环境选择一个唯一且合理的名称。
   - **spec.applicationRef**：（必需）要备份的应用程序的 Kubernetes 名称。
   - **spec.appVaultTargetsRef**: (*Required*) 备份内容应存储的 AppVault 的名称。
   - **spec.dataMover**：（*Optional*）一个字符串，指示要用于备份操作的备份工具。该值区分大小写，必须为 `CBS`。
   - **spec.reclaimPolicy**：（可选）定义删除 Backup CR 时备份内容（元数据/卷数据）会发生什么。可能的值：
     - `Delete`
     - `Retain` (默认)
   - **spec.cleanupSnapshot**：（必需）确保备份 CR 创建的临时快照在备份操作完成后不被删除。建议值：`false`。
   - **spec.replicateSnapshot**：（*Required*）指示 Backup and Recovery 将快照复制到二级存储。必需值：`true`。
   - **spec.replicateSnapshotReclaimPolicy**: (*Optional*) 定义已复制快照在删除时会发生什么。可能的值：
     - `Delete`
     - `Retain` (默认)

     示例 YAML：

```yaml
apiVersion: protect.trident.netapp.io/v1
kind: Backup
metadata:
  namespace: my-app-namespace
  name: 3-2-1-fanout-backup-cr
spec:
  applicationRef: my-application
  appVaultTargetsRef: appvault-targets-name
  dataMover: CBS
  reclaimPolicy: Retain
  cleanupSnapshot: false
  replicateSnapshot: true
  replicateSnapshotReclaimPolicy: Retain
```

3. 使用正确的值填充 `3-2-1-fanout-backup-cr.yaml`文件后，应用 CR：

```
kubectl apply -f 3-2-1-fanout-backup-cr.yaml
```

支持的备份注释

下表介绍了创建备份 CR 时可以使用的批注。

标注	类型	描述	默认值
protect.trident.netapp.io/full-backup	string	指定备份是否应为非增量备份。设置为 `true` 以创建非增量备份。最佳做法是定期执行完整备份，然后在完整备份之间执行增量备份，以最大限度地降低与恢复相关的风险。	"false"
protect.trident.netapp.io/snapshot-completion-timeout	string	整个快照操作完成所允许的最长时间。	“60 分钟”
protect.trident.netapp.io/volume-snapshots-ready-to-use-timeout	string	允许卷快照达到准备就绪状态的最长时间。	"30 分钟"
protect.trident.netapp.io/volume-snapshots-created-timeout	string	允许创建卷快照的最长时间。	"5 分钟"
protect.trident.netapp.io/pvc-bind-timeout-sec	string	在操作失败之前，等待任何新创建的 PersistentVolumeClaims (PVC) 到达 `Bound` 阶段的最长时间（秒）。	"1200"（20 分钟）

恢复 **Kubernetes** 应用程序

使用 **Web UI** 还原 **Kubernetes** 应用程序

NetApp Backup and Recovery使您能够恢复已通过保护策略保护的应用程序。要恢复应用程序，应用程序需要至少有一个可用的恢复点。恢复点由本地快照或对象存储备份（或两者）组成。您可以使用本地、辅助或对象存储存档来恢复应用程序。

开始之前

如果要还原使用 Trident Protect 备份的应用程序，请确保 Trident Protect 同时安装在源和目标集群上。

所需的**NetApp Console**角色

组织管理员或SnapCenter管理员。"了解NetApp Backup and Recovery访问角色" 。 "了解所有服务的NetApp Console访问角色" 。

步骤

1. 在 NetApp Backup and Recovery 菜单中，选择*恢复*。
2. 从列表中选择一个 Kubernetes 应用程序，并为该应用程序选择*查看和恢复*。

   出现还原点列表。

3. 选择要使用的还原点的 **Restore** 按钮。

常规设置

1. 选择要从中还原的源位置。

2. 从*Cluster*列表中选择目标集群。

   目前不支持将 Trident Protect 创建的本地快照还原到其他集群。

3. 选择还原到原始命名空间或新命名空间。
4. 如果选择还原到新命名空间，请输入要使用的目标命名空间。
5. 选择“下一步”。

资源选择

1. 选择是否要恢复与应用程序相关的所有资源，或者使用过滤器选择要恢复的特定资源：

恢复所有资源

1. 选择*恢复所有资源*。
2. 选择“下一步”。

恢复特定资源

1. 选择*选择性资源*。
2. 选择资源过滤器的行为。如果您选择“包括”，则会恢复您选择的资源。如果您选择“排除”，则您选择的资源将不会被恢复。
3. 选择*添加规则*来添加定义选择资源的过滤器的规则。您至少需要一条规则来过滤资源。

   每个规则都可以根据资源命名空间、标签、组、版本和种类等标准进行过滤。

4. 选择*保存*来保存每条规则。
5. 添加完所有需要的规则后，选择*搜索*即可查看备份档案中符合过滤条件的可用资源。

   显示的资源是集群上当前存在的资源。

6. 对结果满意后，选择*下一步*。

目的地设置

1. 展开 **Destination settings** 部分，然后选择恢复到默认存储类、其他存储类，或者如果要恢复到其他集群，则将存储类映射到目标集群。
2. 如果选择还原到其他存储类，请选择与每个源存储类匹配的目标存储类。
3. 或者，如果您要还原使用 Trident Protect 创建的备份或快照，请查看 AppVault 用作还原操作存储桶的详细信息。如果您的环境或 AppVault 状态发生变化，请选择 **Sync App Vault** 以刷新详细信息。

   如果需要在 Kubernetes 集群上创建 AppVault 以便还原使用 Trident Protect 创建的备份或快照，请参阅 "使用 Trident Protect AppVault 对象管理存储桶"。

4. （可选）展开 **Restore scripts** 部分，并启用 **Postscript** 选项以选择将在还原操作完成后运行的执行钩子模板。如果需要，请输入脚本需要的任何参数，并添加标签选择器以根据资源标签筛选资源。

5. 选择*恢复*。

使用自定义资源还原 **Kubernetes** 应用程序

您可以使用自定义资源从快照或备份还原应用程序。将应用程序还原到同一集群时，从现有快照还原将更快。

- 还原应用程序时，为应用程序配置的所有执行挂钩都会随应用程序一起还原。如果存在还原后执行挂钩，它将作为还原操作的一部分自动运行。
- qtree 卷支持从备份还原到其他命名空间或原始命名空间。但是，qtree 卷不支持从快照还原到其他命名空间或原始命名空间。
- 您可以使用高级设置自定义还原操作。要了解更多信息，请参阅 "使用高级自定义资源还原设置"。

将备份还原到其他命名空间

使用 BackupRestore CR 将备份还原到其他命名空间时，Backup and Recovery 会在新命名空间中还原应用程序，并为还原的应用程序创建应用程序 CR。要保护还原的应用程序，请创建按需备份或快照，或建立保护计划。

- 使用现有资源将备份还原到其他命名空间不会更改与备份中的名称共享的任何资源。要还原备份中的所有资源，请删除并重新创建目标命名空间，或将备份还原到新命名空间。
- 使用 CR 还原到新命名空间时，您必须在应用 CR 之前手动创建目标命名空间。Backup and Recovery 仅在使用 CLI 时自动创建命名空间。

开始之前

确保 AWS 会话令牌过期时间足以进行任何长期运行的 s3 还原操作。如果令牌在还原操作期间过期，则操作可能会失败。

- 有关检查当前会话令牌过期的详细信息，请参见 "AWS API 文档"。
- 有关 AWS 资源凭据的详细信息，请参见 "AWS IAM 文档"。

使用 Kopia 作为数据移动器还原备份时，可以选择在 CR 中指定注释，以控制 Kopia 使用的临时存储的行为。有关可以配置的选项的详细信息，请参见 "Kopia 文档"。

步骤

1. 创建自定义资源 (CR) 文件并将其命名为 `trident-protect-backup-restore-cr.yaml`。
2. 在创建的文件中，配置以下属性：
   - **metadata.name**：（*Required*）此自定义资源的名称；为您的环境选择一个唯一且合理的名称。
   - **spec.appArchivePath**：AppVault 中存储备份内容的路径。您可以使用以下命令查找此路径：

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

- **spec.appVaultRef**：（必需）存储备份内容的 AppVault 的名称。
- **spec.namespaceMapping**：还原操作的源命名空间到目标命名空间的映射。使用环境中的信息替换 `my-source-namespace`和 `my-destination-namespace`。

```
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. （可选）如果需要仅选择要还原的应用程序的某些资源，请添加包含或排除标有特定标签的资源的筛选：

> Trident Protect 会自动选择一些资源，因为它们与您选择的资源之间存在关系。例如，如果您选择了永久卷声明资源，并且它具有关联的 pod，则 Trident Protect 也将还原关联的 pod。

- **resourceFilter.resourceSelectionCriteria**：（筛选时需要）使用 `Include`或 `Exclude`来包含或排除在 resourceMatchers 中定义的资源。添加以下 resourceMatchers 参数以定义要包括或排除的资源：
  - **resourceFilter.resourceMatchers**：resourceMatcher 对象数组。如果在此数组中定义多个元素，则它们将作为 OR 操作进行匹配，并且每个元素（组、种类、版本）内的字段将作为 AND 操作进行匹配。
    - **resourceMatchers[].group**：（*Optional*）要筛选的资源的组。
    - **resourceMatchers[].kind**：（*Optional*）要筛选的资源的类型。
    - **resourceMatchers[].version**：（*Optional*）要筛选的资源的版本。
    - **resourceMatchers[].names**：（可选）要过滤的资源的 Kubernetes metadata.name 字段中的名称。
    - **resourceMatchers[].namespaces**: (*Optional*) 要过滤的资源的 Kubernetes metadata.name 字段中的命名空间。
    - **resourceMatchers[].labelSelectors**: (*Optional*) 资源的 Kubernetes metadata.name 字段中的标签选择器字符串，如 "Kubernetes 文档" 中所定义。例如：`"trident.netapp.io/os=linux"`。

      例如：

```
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. 使用正确的值填充 `trident-protect-backup-restore-cr.yaml`文件后，应用 CR：

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

将备份还原到原始命名空间

您可以随时将备份还原到原始命名空间。

开始之前

确保 AWS 会话令牌过期时间足以进行任何长期运行的 s3 还原操作。如果令牌在还原操作期间过期，则操作可能会失败。

- 有关检查当前会话令牌过期的详细信息，请参见 "AWS API 文档"。
- 有关 AWS 资源凭据的详细信息，请参见 "AWS IAM 文档"。

使用 Kopia 作为数据移动器还原备份时，可以选择在 CR 中指定注释，以控制 Kopia 使用的临时存储的行为。有关可以配置的选项的详细信息，请参见 "Kopia 文档"。

步骤

1. 创建自定义资源 (CR) 文件并将其命名为 `trident-protect-backup-ipr-cr.yaml`。
2. 在创建的文件中，配置以下属性：
   - **metadata.name**：（*Required*）此自定义资源的名称；为您的环境选择一个唯一且合理的名称。
   - **spec.appArchivePath**：AppVault 中存储备份内容的路径。您可以使用以下命令查找此路径：

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

- **spec.appVaultRef**：（必需）存储备份内容的 AppVault 的名称。

  例如：

```
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. （可选）如果需要仅选择要还原的应用程序的某些资源，请添加包含或排除标有特定标签的资源的筛选：

   Trident Protect 会自动选择一些资源，因为它们与您选择的资源之间存在关系。例如，如果您选择了永久卷声明资源，并且它具有关联的 pod，则 Trident Protect 也将还原关联的 pod。

   - **resourceFilter.resourceSelectionCriteria**：（筛选时需要）使用 `Include`或 `Exclude`来包含或排除在 resourceMatchers 中定义的资源。添加以下 resourceMatchers 参数以定义要包括或排除的资源：
     - **resourceFilter.resourceMatchers**：resourceMatcher 对象数组。如果在此数组中定义多个元素，则它们将作为 OR 操作进行匹配，并且每个元素（组、种类、版本）内的字段将作为 AND 操作进行匹配。
       - **resourceMatchers[].group**：（*Optional*）要筛选的资源的组。
       - **resourceMatchers[].kind**：（*Optional*）要筛选的资源的类型。
       - **resourceMatchers[].version**：（*Optional*）要筛选的资源的版本。
       - **resourceMatchers[].names**：（可选）要过滤的资源的 Kubernetes metadata.name 字段中的名称。
       - **resourceMatchers[].namespaces**: (*Optional*) 要过滤的资源的 Kubernetes metadata.name 字段中的命名空间。
       - **resourceMatchers[].labelSelectors**: (*Optional*) 资源的 Kubernetes metadata.name 字段中的标签选择器字符串，如 "Kubernetes 文档" 中所定义。例如：`"trident.netapp.io/os=linux"`。

         例如：

```
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. 使用正确的值填充 `trident-protect-backup-ipr-cr.yaml`文件后，应用 CR：

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

将备份还原到其他集群

如果原始群集出现问题，可以将备份还原到其他群集。

- 使用 Kopia 作为数据移动器还原备份时，可以选择在 CR 中指定注释，以控制 Kopia 使用的临时存储的行为。有关可以配置的选项的详细信息，请参见 "Kopia 文档"。
- 当使用 CR 还原到新的命名空间时，您必须在应用 CR 之前手动创建目标命名空间。

开始之前

确保满足以下先决条件：

- 目标集群已安装 Trident Protect。
- 目标集群可以访问与源集群相同的 AppVault 存储桶路径，备份存储在该路径中。
- 确保 AWS 会话令牌过期时间足以进行任何长期运行的还原操作。如果令牌在还原操作期间过期，则操作可能会失败。
  - 有关检查当前会话令牌过期的详细信息，请参见 "AWS API 文档"。
  - 有关 AWS 资源凭据的详细信息，请参见 "AWS 文档"。

步骤

1. 使用 Trident Protect CLI 插件检查目标集群上 AppVault CR 的可用性：

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

确保目标集群上存在用于应用程序还原的命名空间。

2. 从目标集群查看可用的 AppVault 的备份内容：

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

运行此命令会显示 AppVault 中的可用备份，包括其原始群集、相应的应用程序名称、时间戳和存档路径。

输出示例：

```
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
|   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. 使用 AppVault 名称和存档路径将应用程序还原到目标集群：
4. 创建自定义资源 (CR) 文件并将其命名为 `trident-protect-backup-restore-cr.yaml`。
5. 在创建的文件中，配置以下属性：
   - **metadata.name**：（*Required*）此自定义资源的名称；为您的环境选择一个唯一且合理的名称。
   - **spec.appVaultRef**：（必需）存储备份内容的 AppVault 的名称。
   - **spec.appArchivePath**：AppVault 中存储备份内容的路径。您可以使用以下命令查找此路径：

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

如果 BackupRestore CR 不可用，您可以使用步骤 2 中提到的命令查看备份内容。

- **spec.namespaceMapping**：还原操作的源命名空间到目标命名空间的映射。使用环境中的信息替换 `my-source-namespace`和 `my-destination-namespace`。

  例如：

```
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. 使用正确的值填充 `trident-protect-backup-restore-cr.yaml`文件后，应用 CR：

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

将快照还原到其他命名空间

您可以使用自定义资源 (CR) 文件将数据从快照还原到不同的命名空间或原始源命名空间。使用 SnapshotRestore CR 将快照还原到其他命名空间时，Backup and Recovery 会在新的命名空间中还原应用程序，并为还原的应用程序创建应用程序 CR。要保护还原的应用程序，请创建按需备份或快照，或建立保护计划。

- SnapshotRestore 支持 `spec.storageClassMapping`属性，但仅当源和目标存储类使用相同的存储后端时。如果尝试还原到使用不同存储后端的 `StorageClass`，还原操作将失败。
- 当使用 CR 还原到新的命名空间时，您必须在应用 CR 之前手动创建目标命名空间。

开始之前

确保 AWS 会话令牌过期时间足以进行任何长期运行的 s3 还原操作。如果令牌在还原操作期间过期，则操作可能会失败。

- 有关检查当前会话令牌过期的详细信息，请参见 "AWS API 文档"。
- 有关 AWS 资源凭据的详细信息，请参见 "AWS IAM 文档"。

步骤

1. 创建自定义资源 (CR) 文件并将其命名为 `trident-protect-snapshot-restore-cr.yaml`。
2. 在创建的文件中，配置以下属性：
   - **metadata.name**：（*Required*）此自定义资源的名称；为您的环境选择一个唯一且合理的名称。
   - **spec.appVaultRef**：（必需）存储快照内容的 AppVault 的名称。

- **spec.appArchivePath**：AppVault 中存储快照内容的路径。您可以使用以下命令查找此路径：

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

- **spec.namespaceMapping**：还原操作的源命名空间到目标命名空间的映射。使用环境中的信息替换 `my-source-namespace`和 `my-destination-namespace`。

```
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. （可选）如果需要仅选择要还原的应用程序的某些资源，请添加包含或排除标有特定标签的资源的筛选：

> Trident Protect 会自动选择一些资源，因为它们与您选择的资源之间存在关系。例如，如果您选择了永久卷声明资源，并且它具有关联的 pod，则 Trident Protect 也将还原关联的 pod。

- **resourceFilter.resourceSelectionCriteria**：（筛选时需要）使用 `Include`或 `Exclude`来包含或排除在 resourceMatchers 中定义的资源。添加以下 resourceMatchers 参数以定义要包括或排除的资源：
  - **resourceFilter.resourceMatchers**：resourceMatcher 对象数组。如果在此数组中定义多个元素，则它们将作为 OR 操作进行匹配，并且每个元素（组、种类、版本）内的字段将作为 AND 操作进行匹配。
    - **resourceMatchers[].group**：（*Optional*）要筛选的资源的组。
    - **resourceMatchers[].kind**：（*Optional*）要筛选的资源的类型。
    - **resourceMatchers[].version**：（*Optional*）要筛选的资源的版本。
    - **resourceMatchers[].names**：（可选）要过滤的资源的 Kubernetes metadata.name 字段中的名称。
    - **resourceMatchers[].namespaces**: (*Optional*) 要过滤的资源的 Kubernetes metadata.name 字段中的命名空间。
    - **resourceMatchers[].labelSelectors**: (*Optional*) 资源的 Kubernetes metadata.name 字段中的标签选择器字符串，如 "Kubernetes 文档" 中所定义。例如：`"trident.netapp.io/os=linux"`。

      例如：

```
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. 使用正确的值填充 `trident-protect-snapshot-restore-cr.yaml`文件后，应用 CR：

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

将快照还原到原始命名空间

您可以随时将快照还原到原始命名空间。

开始之前

确保 AWS 会话令牌过期时间足以进行任何长期运行的 s3 还原操作。如果令牌在还原操作期间过期，则操作可能会失败。

- 有关检查当前会话令牌过期的详细信息，请参见 "AWS API 文档"。
- 有关 AWS 资源凭据的详细信息，请参见 "AWS IAM 文档"。

步骤

1. 创建自定义资源 (CR) 文件并将其命名为 `trident-protect-snapshot-ipr-cr.yaml`。
2. 在创建的文件中，配置以下属性：
   - **metadata.name**：（*Required*）此自定义资源的名称；为您的环境选择一个唯一且合理的名称。
   - **spec.appVaultRef**：（必需）存储快照内容的 AppVault 的名称。
   - **spec.appArchivePath**：AppVault 中存储快照内容的路径。您可以使用以下命令查找此路径：

     ```
     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. （可选）如果需要仅选择要还原的应用程序的某些资源，请添加包含或排除标有特定标签的资源的筛选：

> Trident Protect 会自动选择一些资源，因为它们与您选择的资源之间存在关系。例如，如果您选择了永久卷声明资源，并且它具有关联的 pod，则 Trident Protect 也将还原关联的 pod。

- **resourceFilter.resourceSelectionCriteria**：（筛选时需要）使用 `Include`或 `Exclude`来包含或排除在 resourceMatchers 中定义的资源。添加以下 resourceMatchers 参数以定义要包括或排除的资源：
  - **resourceFilter.resourceMatchers**：resourceMatcher 对象数组。如果在此数组中定义多个元素，则它们将作为 OR 操作进行匹配，并且每个元素（组、种类、版本）内的字段将作为 AND 操作进行匹配。
    - **resourceMatchers[].group**：（*Optional*）要筛选的资源的组。
    - **resourceMatchers[].kind**：（*Optional*）要筛选的资源的类型。
    - **resourceMatchers[].version**：（*Optional*）要筛选的资源的版本。
    - **resourceMatchers[].names**：（可选）要过滤的资源的 Kubernetes metadata.name 字段中的名称。
    - **resourceMatchers[].namespaces**: (*Optional*) 要过滤的资源的 Kubernetes metadata.name 字段中的命名空间。
    - **resourceMatchers[].labelSelectors**: (*Optional*) 资源的 Kubernetes metadata.name 字段中的标签选择器字符串，如 "Kubernetes 文档" 中所定义。例如：`"trident.netapp.io/os=linux"`。

      例如：

```
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. 使用正确的值填充 `trident-protect-snapshot-ipr-cr.yaml`文件后，应用 CR：

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

使用高级自定义资源还原设置

您可以使用高级设置（如注释、命名空间设置和存储选项）自定义还原操作，以满足您的特定要求。

还原和故障转移操作期间的命名空间注释和标签

在恢复和故障转移操作期间，目标命名空间中的标签和注释将与源命名空间中的标签和注释匹配。将添加目标命名空间中不存在的源命名空间中的标签或注释，并覆盖已存在的任何标签或注释，以匹配源命名空间中的值。仅存在于目标命名空间上的标签或注释保持不变。

> 如果使用 Red Hat OpenShift，请务必注意命名空间注释在 OpenShift 环境中的关键作用。命名空间注释可确保还原的 Pod 遵守 OpenShift 安全上下文约束 (SCC) 定义的适当权限和安全配置，并且可以访问卷而不会出现权限问题。有关详细信息，请参见 "OpenShift 安全上下文约束文档"。

在执行还原或故障转移操作之前，可以通过设置 Kubernetes 环境变量 `RESTORE_SKIP_NAMESPACE_ANNOTATIONS`来防止目标命名空间中的特定注释被覆盖。例如：

```
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
```

执行还原或故障切换操作时，在 `restoreSkipNamespaceAnnotations`和 `restoreSkipNamespaceLabels`中指定的任何命名空间批注和标签都将从还原或故障切换操作中排除。确保在初始 Helm 安装过程中配置了这些设置。要了解更多信息，请参阅"配置其他 Trident Protect helm 图表设置"。

如果使用带有 `--create-namespace` 标志的 Helm 安装源应用程序，则会对 `name` 标签键进行特殊处理。在还原或故障转移过程中，Trident Protect 会将此标签复制到目标命名空间，但如果来自源的值与源命名空间匹配，则将值更新为目标命名空间值。如果此值与源命名空间不匹配，则将其复制到目标命名空间，不进行任何更改。

示例

以下示例显示了源和目标命名空间，每个命名空间都有不同的注释和标签。您可以查看操作之前和之后的目标命名空间的状态，以及如何在目标命名空间中组合或覆盖注释和标签。

还原或故障转移操作之前

下表显示了还原或故障转移操作之前的示例源和目标命名空间的状态：

命名空间	标注	标签
命名空间 ns-1（源）	• annotation.one/key: "updatedvalue" • annotation.two/key: "true"	• environment=production • 合规性=hipaa • name=ns-1
命名空间 ns-2（目标）	• annotation.one/key: "true" • annotation.three/key: "false"	• 角色=数据库

还原操作后

下表显示了还原或故障转移操作后示例目标命名空间的状态。已添加一些键，一些键已被覆盖，并且已更新 `name` 标签以匹配目标命名空间：

命名空间	标注	标签
命名空间 ns-2（目标）	• annotation.one/key: "updatedvalue" • annotation.two/key: "true" • annotation.three/key: "false"	• name=ns-2 • 合规性=hipaa • environment=production • 角色=数据库

支持的字段

本节描述可用于还原操作的其他字段。

存储类映射

该 `spec.storageClassMapping` 属性定义从源应用程序中存在的存储类到目标集群上的新存储类的映射。您可以在具有不同存储类的集群之间迁移应用程序或更改 BackupRestore 操作的存储后端时使用此功能。

示例：

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

支持的注释

本节列出了用于在系统中配置各种行为的支持注释。如果用户未明确设置注释，系统将使用默认值。

标注	类型	描述	默认值
protect.trident.netapp.io/data-mover-timeout-sec	string	允许数据移动器操作停止的最长时间（以秒为单位）。	"300"
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	string	Kopia 内容缓存的最大大小限制（以兆字节为单位）。	"1000"
protect.trident.netapp.io/pvc-bind-timeout-sec	string	在操作失败之前，等待任何新创建的 PersistentVolumeClaims (PVC) 到达 `Bound`阶段的最长时间（以秒为单位）。适用于所有还原 CR 类型（BackupRestore、BackupInplaceRestore、SnapshotRestore、SnapshotInplaceRestore）。如果您的存储后端或集群通常需要更多时间，请使用更高的值。	"1200"（20 分钟）

管理 Kubernetes 集群

NetApp Backup and Recovery使您能够发现和管理 Kubernetes 集群，以便您可以保护集群托管的资源。

所需的**NetApp Console**角色

组织管理员或SnapCenter管理员。"了解NetApp Backup and Recovery访问角色" 。 "了解所有服务的NetApp Console访问角色" 。

要发现 Kubernetes 集群，请参阅"发现 Kubernetes 工作负载"。

编辑 Kubernetes 集群信息

如果需要更改集群名称，您可以编辑集群。

步骤

1. 在NetApp Backup and Recovery中，选择 **Inventory** > **Clusters**。
2. 在集群列表中，选择要编辑的集群并选择相关的操作菜单。
3. 选择*编辑集群*。
4. 对集群名称进行任何必要的更改。集群名称需要与您在发现过程中使用 Helm 命令的名称相匹配。
5. 选择*完成*。

删除 Kubernetes 集群

要停止保护 Kubernetes 集群，请禁用保护并删除相关应用程序，然后从NetApp Backup and Recovery中删除该集群。 NetApp Backup and Recovery不会删除集群或其资源；它只会从NetApp Console清单中删除集群。

步骤

1. 在NetApp Backup and Recovery中，选择 **Inventory** > **Clusters**。
2. 在集群列表中，选择要编辑的集群并选择相关的操作菜单。
3. 选择*删除集群*。
4. 查看确认对话框中的信息，然后选择*删除*。

管理 Kubernetes 应用程序

NetApp Backup and Recovery使您能够取消保护并删除 Kubernetes 应用程序及相关资源。

所需的**NetApp Console**角色

组织管理员或SnapCenter管理员。"了解NetApp Backup and Recovery访问角色" 。 "了解所有服务的NetApp Console访问角色" 。

取消保护 **Kubernetes** 应用程序

如果您不再需要保护某个应用程序，可以取消保护。当您取消保护应用程序时， NetApp Backup and Recovery 会停止保护该应用程序，但保留所有相关的备份和快照。

当保护操作仍在运行时，您无法取消对应用程序的保护。要么等待操作完成，要么作为解决方法，删除还原点正在运行的保护操作正在使用的。然后，您可以取消对应用程序的保护。

步骤

1. 在NetApp Backup and Recovery中，选择 **Inventory**。
2. 选择一个 Kubernetes 实例，然后选择“查看”以查看与该实例关联的资源。
3. 选择“应用程序”选项卡。
4. 在应用程序列表中，选择要取消保护的应用程序并选择相关的操作菜单。
5. 选择*取消保护*。
6. 阅读通知，准备好后，选择*取消保护*。

删除 **Kubernetes** 应用程序

删除不再需要的应用程序。 NetApp Backup and Recovery停止保护并删除已删除应用程序的所有备份和快照。

步骤

1. 在NetApp Backup and Recovery中，选择 **Inventory**。
2. 选择一个 Kubernetes 实例，然后选择“查看”以查看与该实例关联的资源。
3. 选择“应用程序”选项卡。
4. 在应用程序列表中，选择要删除的应用程序并选择相关的操作菜单。
5. 选择*删除*。
6. 启用*删除快照和备份*以删除应用程序的所有快照和备份。

   您将无法再使用这些快照和备份恢复应用程序。

7. 确认操作并选择*删除*。

删除 **Kubernetes** 应用程序的还原点

如果需要取消对应用程序的保护，并且保护操作当前正在运行，则可能需要删除该应用程序的还原点。

步骤

1. 在 NetApp Backup and Recovery 菜单中，选择*恢复*。
2. 从列表中选择一个 Kubernetes 应用程序，并为该应用程序选择*查看和恢复*。

   出现还原点列表。

3. 选择需要删除的恢复点，然后选择操作图标 ••• > 删除恢复点 将其删除。

管理适用于 Kubernetes 工作负载的NetApp Backup and Recovery执行挂钩模板

执行钩子是一种自定义操作，它与托管 Kubernetes 应用程序中的数据保护操作一起运行。例如，通过使用执行挂钩在快照之前暂停数据库事务并在之后恢复它们来创建应用程序一致的快照。创建执行钩子模板时，指定钩子类型、要运行的脚本以及目标容器的过滤器。使用模板将执行挂钩链接到您的应用程序。

NetApp Backup and Recovery 会在数据保护期间冻结和解冻应用程序的文件系统，例如 KubeVirt。您可以全局禁用此行为，也可以使用 Trident Protect 文档对特定应用程序禁用此行为：

- 要为所有应用程序禁用此行为，请参阅 "使用 KubeVirt 虚拟机保护数据"。
- 要针对特定应用程序禁用此行为，请参阅 "定义应用程序"。

所需的**NetApp Console**角色

组织管理员或SnapCenter管理员。"了解NetApp Backup and Recovery访问角色" 。 "了解所有服务的NetApp Console访问角色" 。

执行钩子的类型

NetApp Backup and Recovery根据运行时间支持以下类型的执行挂钩：

- 预快照
- 快照后
- 预备份
- 备份后
- 恢复后

执行顺序

当运行数据保护操作时，执行挂钩事件按以下顺序发生：

1. 任何适用的自定义预操作执行挂钩都在适当的容器上运行。您可以创建多个自定义预操作挂钩，但它们的执行顺序无法保证或配置。
2. 如果适用，则会发生文件系统冻结。
3. 执行数据保护操作。
4. 如果适用，冻结的文件系统将被解冻。
5. NetApp Backup and Recovery在适当的容器上运行任何适用的自定义操作前执行挂钩。您可以创建多个自定义后操作挂钩，但它们的执行顺序无法保证或配置。

如果创建多个相同类型的钩子，则无法保证它们的执行顺序。不同类型的钩子总是按照指定的顺序运行。例如，以下是具有所有不同类型钩子的配置的执行顺序：

1. 快照前钩子执行

2. 快照后钩子执行
3. 执行备份前挂钩
4. 执行备份后钩子

在生产中启用执行挂钩脚本之前对其进行测试。使用“kubectl exec”测试脚本，然后通过将应用程序克隆到临时命名空间并恢复来验证快照和备份。

如果快照前执行钩子添加、更改或删除 Kubernetes 资源，则这些更改将包含在快照或备份以及任何后续恢复操作中。

关于自定义执行钩子的重要说明

在为您的应用程序规划执行挂钩时，请考虑以下事项。

- 执行钩子必须使用脚本来执行操作。许多执行钩子可以引用同一个脚本。
- 执行钩子需要以可执行shell脚本的格式编写。
- 脚本大小限制为 96KB。
- 执行挂钩设置和任何匹配标准用于确定哪些挂钩适用于快照、备份或恢复操作。

执行挂钩可以减少或禁用应用程序功能。让您的自定义钩子尽快运行。如果您启动带有相关执行挂钩的备份或快照操作，但随后取消它，则如果备份或快照操作已经开始，则仍允许挂钩运行。这意味着备份后执行挂钩中使用的逻辑不能假定备份已完成。

执行钩子过滤器

当您为应用程序添加或编辑执行挂钩时，您可以向执行挂钩添加过滤器来管理该挂钩将匹配哪些容器。过滤器对于在所有容器上使用相同容器镜像但可能将每个镜像用于不同目的的应用程序（例如 Elasticsearch）很有用。过滤器允许您创建执行挂钩在某些（但不一定是所有）相同的容器上运行的场景。如果为单个执行挂钩创建多个过滤器，它们将通过逻辑 AND 运算符组合在一起。每个执行挂钩最多可以有 10 个活动过滤器。

添加到执行挂钩的每个过滤器都使用正则表达式来匹配集群中的容器。当钩子与容器匹配时，钩子将在该容器上运行其关联的脚本。过滤器的正则表达式使用正则表达式 2 (RE2) 语法，该语法不支持创建从匹配列表中排除容器的过滤器。有关NetApp Backup and Recovery在执行钩子过滤器中支持的正则表达式的语法的信息，请参见 "正则表达式 2 (RE2) 语法支持"。

如果将命名空间过滤器添加到在恢复或克隆操作后运行的执行挂钩，并且恢复或克隆源和目标位于不同的命名空间中，则命名空间过滤器仅适用于目标命名空间。

执行钩子示例

访问 "NetApp Verda GitHub 项目"下载流行应用程序（如 Apache Cassandra 和 Elasticsearch）的真实执行挂钩。您还可以查看示例并获得构建您自己的自定义执行挂钩的想法。

创建执行钩子模板

您可以创建自定义执行挂钩模板，用于在应用程序上执行数据保护操作之前或之后执行操作。

此处创建的模板仅在保护 Kubernetes 工作负载时可用。

步骤

1. 在控制台中，转到*保护*>*备份和恢复*。
2. 选择“设置”选项卡。
3. 展开*执行钩子模板*部分。
4. 选择*创建执行钩子模板*。
5. 输入执行挂钩的名称。
6. （可选）选择一种钩子类型。例如，还原后钩子会在还原操作完成后运行。
7. 在 **Script** 文本框中，输入要作为执行挂钩模板的一部分运行的可执行 shell 脚本。或者，您可以选择*上传脚本*来上传脚本文件。
8. 选择“创建”。

   创建模板后，它将出现在*执行挂钩模板*部分的模板列表中。

在 NetApp Backup and Recovery 中创建和管理 Kubernetes 工作负载的保护报告

在 NetApp Backup and Recovery 中，为 Kubernetes 工作负载创建保护报告，以查看保护状态和详细信息，包括成功和失败备份的计数、备份类型、集群健康信息等。

必需的 **NetApp Console** 角色 Backup and Recovery 超级管理员、Backup and Recovery 备份管理员或 Backup and Recovery 还原管理员。了解有关"备份和恢复角色和权限"的信息 "了解所有服务的NetApp Console访问角色"。

创建保护报告

创建保护报告以查看集群的保护状态。

步骤

1. 从NetApp Backup and Recovery菜单中，选择 报告 选项。
2. 选择*创建报告*。
3. 输入报告范围详细信息：
   - 报告名称：输入报告的唯一名称。
   - 报告类型：选择是按帐户报告还是按工作负载报告（从列表中选择 Kubernetes）。
   - **Select cluster**：如果您按工作负载选择，请从要为其生成报告的列表中选择集群，然后选择 **Accept**。选择 **Select all** 为所有集群生成报告。
4. 输入报告范围：选择是否希望报告包括过去一天、过去 7 天、过去 30 天、最后一个季度或去年的数据。
5. 输入报告配置详细信息：选择是仅运行一次报告，还是计划定期生成报告。对于计划报告，请选择重复频率并选择开始日期。

a. 输入电子邮件传递详细信息：（仅适用于计划报告）如果要通过电子邮件传递报告，请输入应接收计划报告的一个或多个电子邮件地址。

在设置页面配置电子邮件通知。有关配置电子邮件通知的详细信息，请参阅"配置设置"。

6. 选择“创建”。

下载保护报告

下载生成的保护报告，可以是 JSON 文件或 PDF 文档，以便您查看和共享。

步骤

1. 从NetApp Backup and Recovery菜单中，选择 报告 选项。
2. 在 **Reports** 页面上，选择 **Reports** 菜单以查看生成的保护报告列表。
3. 对于要下载的报告，请选择操作图标 ••• > 下载。
   - 选择 **Download JSON** 以 JSON 格式下载报告。
   - 选择 **Download PDF** 将报告下载为 PDF 文档。

查看防护报告

快速查看 NetApp Backup and Recovery 中保护报告的交互式详细信息。您可以查看作业摘要信息、数据保护状态、配置详细信息等。

步骤

1. 从NetApp Backup and Recovery菜单中，选择 报告 选项。
2. 在 **Reports** 页面上，选择 **Reports** 菜单以查看生成的保护报告列表。
3. 对于要查看的报告，请选择操作图标 ••• > 查看报告。

   此时将显示报告详细信息。

删除保护报告

当您不再需要防护报告时，请将其删除。

步骤

1. 从NetApp Backup and Recovery菜单中，选择 报告 选项。
2. 在 **Reports** 页面上，选择 **Reports** 菜单以查看生成的保护报告列表。
3. 对于要删除的报告，请选择"操作"图标 ••• > **Delete**。
4. 选择 删除 以确认操作。

版权信息

版权所有 © 2026 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。 美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 http://www.netapp.com/TM 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。