



Kubernetesワークロードを保護する NetApp Backup and Recovery

NetApp
June 24, 2026

目次

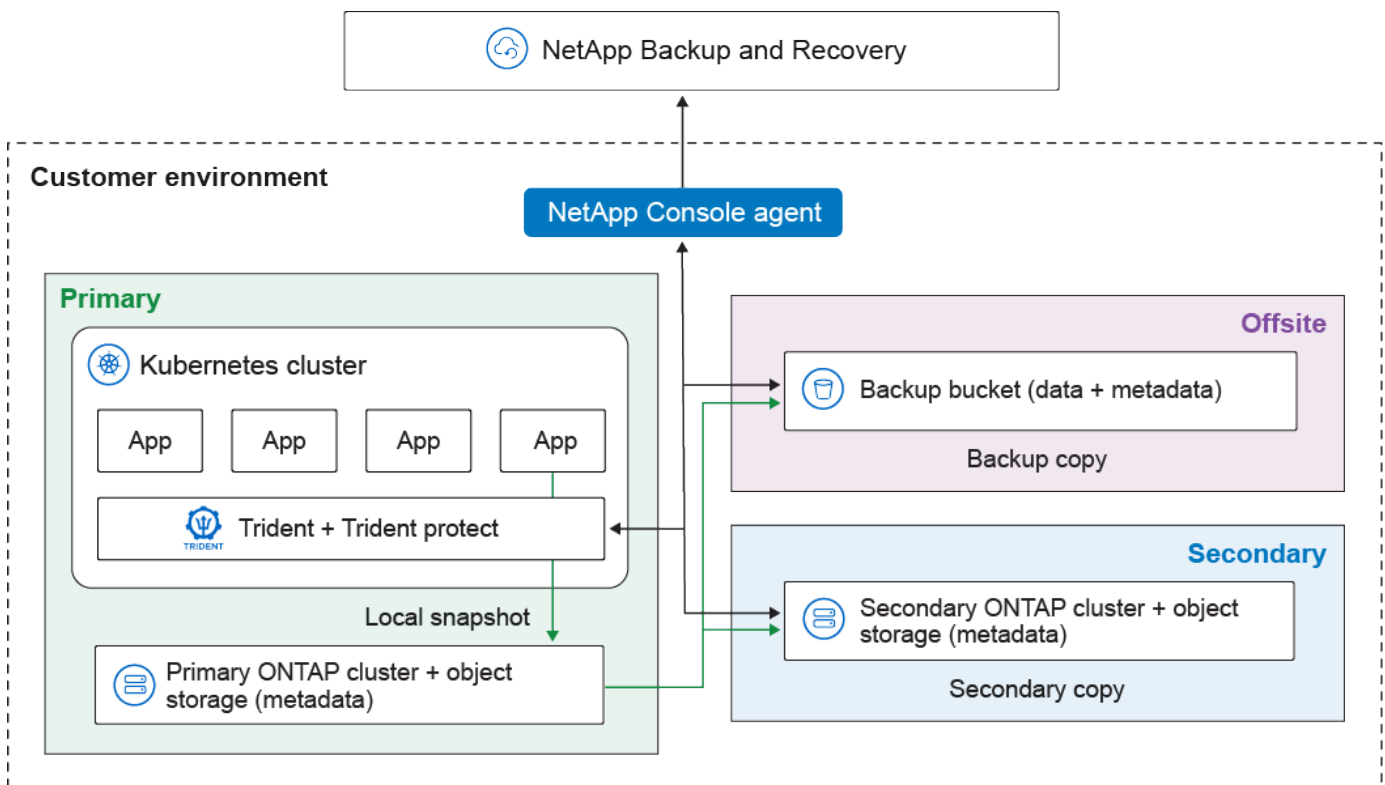
Kubernetesワークロードを保護する	1
Kubernetes ワークロードの管理の概要	1
NetApp Backup and Recoveryで Kubernetes ワークロードを発見	2
Kubernetes ワークロードを発見する	2
NetApp Backup and Recoveryダッシュボードに進みます	3
Kubernetes アプリケーションの追加と保護	3
Kubernetes アプリケーションの追加と保護	3
NetApp Backup and RecoveryでのKubernetesバックアップ ポリシーの作成と管理	8
Backup and Recovery Web UIを使用してKubernetesアプリケーションを今すぐバックアップ	12
Backup and Recoveryのカスタム リソースを使用して、 Kubernetesアプリケーションを今すぐバックアップ	13
Kubernetesアプリケーションを復元する	17
Web UI を使用して Kubernetes アプリケーションをリストアする	17
カスタムリソースを使用した Kubernetes アプリケーションのリストア	22
高度なカスタムリソースのリストア設定を使用する	33
カスタムリソースを使用して復元する際にリソースを変更します	36
Kubernetes クラスターを管理する	41
Kubernetes クラスターの情報を編集する	41
Kubernetes クラスターを削除する	41
Trident Protectのアップグレード	42
Kubernetes アプリケーションを管理する	42
Kubernetes アプリケーションの保護を解除する	42
Kubernetes アプリケーションを削除する	43
Kubernetes アプリケーションの復元ポイントを削除する	43
Kubernetes ワークロード用のNetApp Backup and Recovery実行フック テンプレートを管理する	43
実行フックの種類	44
カスタム実行フックに関する重要な注意事項	45
実行フックフィルター	45
実行フックの例	46
実行フックテンプレートを作成する	46
NetApp Backup and RecoveryでKubernetesワークロードの保護レポートを作成および管理する	46
保護レポートを作成する	47
保護レポートをダウンロードする	47
保護レポートを表示する	47
保護レポートを削除する	48

Kubernetesワークロードを保護する

Kubernetes ワークロードの管理の概要

NetApp Backup and RecoveryでKubernetesワークロードを管理すると、Kubernetesクラスターとアプリケーションを1か所で検出、管理、保護できます。Kubernetesクラスターでホストされているリソースとアプリケーションを管理できます。また、単一のインターフェイスを使用して、Kubernetesアプリケーションの保護ポリシーを作成して関連付けることもできます。

次の図は、Kubernetes ワークロードのバックアップとリカバリのコンポーネントと基本アーキテクチャ、およびデータの異なるコピーを異なる場所に保存する方法を示しています。



NetApp Backup and Recovery は、Kubernetes ワークロードの管理に次の利点をもたらします。

- 複数の Kubernetes クラスターで実行されるアプリケーションを保護するための単一のコントロールプレーン。これらのアプリケーションには、Kubernetes クラスターで実行されるコンテナや仮想マシンが含まれます。
- NetApp SnapMirrorとのネイティブ統合により、すべてのバックアップおよびリカバリ ワークフローのストレージ オフロード機能が有効になります。
- Kubernetes アプリケーションの永久増分バックアップにより、復旧ポイント目標 (RPO) と復旧時間目標 (RTO) が低下します。

Kubernetes ワークロードの管理に関連する次のタスクを実行できます。

- ["Kubernetes ワークロードを発見する"](#)。

- ["Kubernetes クラスターを管理する"](#)。
- ["Kubernetes アプリケーションの追加と保護"](#)。
- ["Kubernetes アプリケーションを管理する"](#)。
- ["Kubernetes アプリケーションを復元する"](#)。

NetApp Backup and Recoveryで Kubernetes ワークロードを発見

NetApp Backup and Recovery、Kubernetes ワークロードを保護する前に検出する必要があります。

必要なNetApp Consoleロール バックアップおよびリカバリのスーパー管理者。詳細はこちら["バックアップとリカバリの役割と権限"](#)。"すべてのサービスに対するNetApp Consoleのアクセスロールについて学習します"。

Kubernetes ワークロードを発見する

バックアップとリカバリのインベントリで、環境内の Kubernetes ワークロードを検出します。ワークロードを追加すると、NetApp Backup and Recoveryに Kubernetes クラスターが追加されます。その後、アプリケーションを追加し、クラスター リソースを保護できます。



現在Trident Protectで保護されているクラスタを検出すると、Trident Protectで使用されていたバックアップスケジュールは検出プロセス中に無効になります（Trident Protectのバックアップスケジュールはバックアップおよびリカバリと互換性がありません）。クラスタのアプリケーションを保護するには、["新しい保護ポリシーを作成する"](#)または、アプリケーションを既存のポリシーに関連付けます。その後、必要に応じてTrident Protectのバックアップスケジュールを削除できます。

手順

1. 次のいずれかを実行します。
 - Kubernetes ワークロードを初めて検出する場合は、NetApp Backup and Recovery の*ワークロード*で、**Kubernetes** タイルを選択します。
 - Kubernetes ワークロードをすでに検出している場合は、NetApp Backup and Recoveryで インベントリ > ワークロード を選択し、リソースの検出 を選択します。
2. **Kubernetes** ワークロード タイプを選択します。
3. クラスター名を入力し、クラスターで使用するコネクタを選択します。
4. 表示されるコマンド ラインの指示に従います。
 - Trident Protect ネームスペースを作成する
 - Kubernetesシークレットを作成する
 - Helmリポジトリを追加する
 - Trident Protectおよび Trident Protect コネクタのインストールまたはアップグレード

これらの手順により、NetApp Backup and Recovery がクラスターと対話できるようになります。

5. 手順を完了したら、[検出] を選択します。

クラスターがインベントリに追加されます。

6. 関連付けられている Kubernetes ワークロードで [表示] を選択すると、そのワークロードのアプリケーション、クラスター、および名前空間のリストが表示されます。

NetApp Backup and Recoveryダッシュボードに進みます

NetApp Backup and Recoveryダッシュボードを表示するには、次の手順に従います。

1. NetApp Consoleメニューから、保護 > バックアップとリカバリ を選択します。
2. ワークロード タイル (Microsoft SQL Server など) を選択します。
3. 「バックアップとリカバリ」メニューから、「ダッシュボード」を選択します。
4. データ保護の健全性を確認します。新しく検出され、保護され、バックアップされたワークロードに基づいて、危険にさらされているワークロードまたは保護されているワークロードの数が増加します。

["ダッシュボードに表示される内容を学ぶ"](#)。

Kubernetes アプリケーションの追加と保護

Kubernetes アプリケーションの追加と保護

NetApp Backup and Recoveryを使用すると、Web UIを介して、またはカスタムリソースファイルを適用して、Kubernetesアプリケーションを追加できます。アプリケーションは、標準のKubernetesリソースで構成される名前スペース、または1つ以上の仮想マシンで構成されるVMベースにすることができます。

必要なNetApp Consoleロール

組織管理者またはSnapCenter管理者。"[NetApp Backup and Recoveryのアクセス ロールについて学習します](#)"。
"[すべてのサービスに対するNetApp Consoleのアクセスロールについて学習します](#)"。

新しいKubernetesアプリケーションを追加して保護する

Kubernetesアプリケーションを保護する最初のステップは、NetApp Backup and Recovery内でアプリケーションを作成することです。アプリケーションを作成すると、Kubernetesクラスター上で実行されているアプリケーションがBackup and Recoveryに認識されるようになります。

開始する前に

Kubernetesアプリケーションを追加して保護する前に、"[Kubernetesワークロードを発見する](#)"。

名前空間ベースのアプリ (Web UI) を追加する

手順

1. NetApp Backup and Recoveryで、**Inventory** を選択します。
2. ページ右上のワークロード一覧で、*Kubernetes*が選択されていることを確認してください。
3. ワークロードの項目で **View** を選択すると、Kubernetes リソースが表示されます。
4. *アプリケーション*タブを選択します。
5. *アプリケーションの作成*を選択します。
6. アプリケーションの名前を入力します。
7. *クラスター*リストから、アプリケーションをホストしているクラスターを選択します。
8. *フィルター*で*名前空間*を選択すると、アプリケーションを名前空間でフィルタリングできます。
9. 必要に応じて、次のいずれかのフィールドを選択して、保護するリソースを検索します。
 - 関連する名前空間
 - リソースの種類
 - ラベルセレクター
 - i. クラスターレベルでスコープが設定されたリソースを追加するには、*クラスタースコープのリソースを追加*を選択します。それらを含めると、アプリケーション作成時にアプリケーションに追加されます。
 - ii. 必要に応じて、「検索」を選択し、検索条件に基づいてリソースを検索します。



NetApp Backup and Recoveryでは、検索パラメータや検索結果は保存されません。これらのパラメータは、選択したKubernetesクラスタ内でアプリケーションに含めることができるリソースを検索するために使用されます。

10. Backup and Recoveryでは、検索条件に一致するリソースの一覧が表示されます。
11. 保護するリソースがリストに含まれている場合は、[次へ] を選択します。
12. オプションで、*ポリシー*領域で、既存の保護ポリシーを選択してアプリケーションを保護するか、新しいポリシーを作成します。ポリシーを選択しない場合、アプリケーションは保護ポリシーなしで作成されます。後で["保護ポリシーを追加する"](#)できます。
13. *プレスクリプトとポストスクリプト*領域で、バックアップ操作の前または後に実行するプレスクリプトまたはポストスクリプトの実行フックを有効にして構成します。プレスクリプトまたはポストスクリプトを有効にするには、少なくとも1つ作成しておく必要があります。["実行フックテンプレート"](#)。
14. *作成*を選択します。

結果

アプリケーションが作成され、Kubernetesインベントリの*Applications*タブにあるアプリケーション一覧に表示されます。NetApp Backup and Recoveryは、設定に基づいてアプリケーションを保護し、その進行状況は*Monitoring*エリアで確認できます。

VMベースのアプリ (Web UI) を追加する

手順

1. NetApp Backup and Recoveryで、**Inventory** を選択します。
2. ページ右上のワークロード一覧で、*Kubernetes*が選択されていることを確認してください。
3. ワークロードの項目で **View** を選択すると、Kubernetes リソースが表示されます。
4. *アプリケーション*タブを選択します。
5. *アプリケーションの作成*を選択します。
6. アプリケーションの名前を入力します。
7. *クラスター*リストから、アプリケーションをホストしているクラスターを選択します。
8. *フィルター*で*仮想マシン*を選択すると、VMベースのアプリケーションを作成できます。
9. 名前空間を選択し、必要に応じてラベルセレクタを含めることで、アプリケーションに追加する仮想マシンを検索します。



リストから仮想マシンを選択した場合、アプリケーション定義は静的になります。つまり、後から新しい仮想マシンをアプリケーションに追加することはできません（追加および保護するには、アプリケーションを編集する必要があります）。ラベルセレクタを使用する場合、個々の仮想マシンを選択したり、生成されたリストを編集したりすることはできませんが、後でセレクタに一致する仮想マシンはすべて自動的に含まれ、保護されます。

選択した仮想マシンが右側のリストに表示されます。

10. リストに保護する VM が含まれている場合は、*Next* を選択します。
11. オプションで、*ポリシー*領域で、既存の保護ポリシーを選択してアプリケーションを保護するか、新しいポリシーを作成します。ポリシーを選択しない場合、アプリケーションは保護ポリシーなしで作成されます。後で["保護ポリシーを追加する"](#)できます。
12. *プレスクリプトとポストスクリプト*領域で、バックアップ操作の前または後に実行するプレスクリプトまたはポストスクリプトの実行フックを有効にして構成します。プレスクリプトまたはポストスクリプトを有効にするには、少なくとも1つ作成しておく必要があります。["実行フックテンプレート"](#)。
13. *作成*を選択します。

結果

アプリケーションが作成され、Kubernetes インベントリの **アプリケーション** タブのアプリケーションリストに表示されます。NetApp Consoleは設定に基づいてアプリケーションの保護を有効にし、バックアップとリカバリの **監視** 領域で進行状況を監視できます。

名前空間ベースのアプリ (CR) を追加する

手順

1. デスティネーション アプリケーションの CR ファイルを作成します：
 - a. カスタムリソース (CR) ファイルを作成し、名前を付けます (例: `my-app-name.yaml`) 。
 - b. 次の属性を設定します：
 - **metadata.name** : (必須) アプリケーションのカスタム リソースの名前。保護操作に必要な他の CR ファイルはこの値を参照するため、選択した名前を書き留めておいてください。
 - **spec.includedNamespaces** : (必須) ネームスペースとラベルセレクタを使用して、アプ

リケーションが使用するネームスペースとリソースを指定します。アプリケーションネームスペースはこのリストに含まれている必要があります。ラベルセレクタはオプションであり、指定された各ネームスペース内のリソースをフィルタリングするために使用できます。

- **spec.includedClusterScopedResources** : (オプション) この属性を使用して、アプリケーション定義に含めるクラスタースコープのリソースを指定します。この属性を使用すると、グループ、バージョン、種類、ラベルに基づいてこれらのリソースを選択できます。
 - **groupVersionKind** : (必須) クラスタースコープのリソースのAPIグループ、バージョン、および種類を指定します。
 - **labelSelector** : (オプション) ラベルに基づいてクラスタースコープのリソースをフィルタリングします。

c. 必要に応じて、次のアノテーションを設定します :

- **metadata.annotations.protect.trident.netapp.io/skip-vm-freeze** : (オプション) このアノテーションは、KubeVirt環境など、スナップショットの前にファイルシステムのフリーズが発生する仮想マシンから定義されたアプリケーションにのみ適用されます。スナップショット作成中にこのアプリケーションがファイルシステムに書き込みできるかどうかを指定します。trueに設定すると、アプリケーションはグローバル設定を無視し、スナップショット作成中にファイルシステムへの書き込みが可能になります。falseに設定すると、アプリケーションはグローバル設定を無視し、スナップショット作成中はファイルシステムがフリーズします。指定されていても、アプリケーション定義に仮想マシンが含まれていない場合、このアノテーションは無視されます。指定されていない場合、アプリケーションは ["グローバルファイルシステム フリーズ設定"](#) に従います。
- **protect.trident.netapp.io/protection-command** : (オプション) この注釈を使用して、NetApp Backup and Recoveryにアプリケーションの保護または保護の停止を指示します。指定可能な値は `protect` または `unprotect` です。
- **protect.trident.netapp.io/protection-policy-name** : (オプション) このアノテーションを使用して、このアプリケーションを保護するために使用するNetApp Backup and Recovery保護ポリシーの名前を指定します。この保護ポリシーは、NetApp Backup and Recoveryにすでに存在している必要があります。

アプリケーションがすでに作成された後にこのアノテーションを適用する必要がある場合は、次のコマンドを使用できます :

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

+
YAMLの例：

+

```
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. (オプション) 特定のラベルでマークされたリソースを含めるか除外するかを指定するフィルタリングを追加します：

- **resourceFilter.resourceSelectionCriteria**：(フィルタリングに必須) `Include`または`Exclude`を使用して、resourceMatchersで定義されたリソースを含めるか除外します。含めるまたは除外するリソースを定義するには、以下のresourceMatchersパラメータを追加します：
 - **resourceFilter.resourceMatchers**：resourceMatcherオブジェクトの配列。この配列に複数の要素を定義すると、それらはOR演算として一致し、各要素内のフィールド(グループ、種類、バージョン)はAND演算として一致します。
 - **resourceMatchers[].group**：(オプション) フィルタリングするリソースのグループ。
 - **resourceMatchers[].kind**：(オプション) フィルタリングするリソースの種類。

- `resourceMatchers[].version` : (オプション) フィルタリングするリソースのバージョン。
- `resourceMatchers[].names` : (オプション) フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前。
- `resourceMatchers[].namespaces` : (オプション) フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前空間。
- `resourceMatchers[].labelSelectors` : (オプション) "[Kubernetesドキュメント](#)"で定義されているリソースの Kubernetes メタデータ.name フィールドのラベルセクタ文字列。
例: `"trident.netapp.io/os=linux"`。



`resourceFilter` と `labelSelector` の両方が使用される場合、`resourceFilter` が最初に実行され、次に `labelSelector` が結果のリソースに適用されます。

例:

```
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. 環境に合わせてアプリケーション CR を作成したら、CR を適用します。例:

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

NetApp Backup and RecoveryでのKubernetesバックアップポリシーの作成と管理

NetApp Backup and Recoveryでは、バックアップの頻度、バックアップの取得時間、保持するバックアップファイルの数などを制御する独自のKubernetesバックアップポリシーを作成できます。



これらのオプションと構成セクションの一部は、すべてのワークロードで使用できるわけではありません。

SnapCenterからリソースをインポートする場合、SnapCenterで使用されるポリシーとNetApp Backup and Recoveryで使用されるポリシーとの間に若干の違いが生じる可能性があります。見る["SnapCenterとNetApp Backup and Recoveryのポリシーの違い"](#)。

ポリシーに関連する次の目標を達成できます。

- ローカルスナップショットポリシーを作成する
- セカンダリストレージへのレプリケーションのポリシーを作成する
- オブジェクトストレージ設定のポリシーを作成する
- 詳細なポリシー設定を構成する
- ポリシーを編集する
- ポリシーの削除

ポリシーを表示

1. NetApp Backup and Recoveryメニューから、ポリシー を選択します。
2. ポリシーの詳細を確認してください。例：
 - ワークロード：例：Microsoft SQL Server、ONTAP ボリューム、VMware、KVM、Hyper-V、Oracle Database、または Kubernetes。
 - バックアップの種類：例としては、完全バックアップやログ バックアップなどがあります。
 - アーキテクチャ：例としては、ローカル スナップショット、ファンアウト、カスケード、ディスク間、ディスクからオブジェクト ストアなどがあります。
 - 保護されているリソース：そのワークロード上のリソースの合計数のうち、保護されているリソースの数を表示します。
 - ランサムウェア保護：ポリシーに、ローカル スナップショットのスナップショット ロック、セカンダリ ストレージのスナップショット ロック、またはオブジェクト ストレージの DataLock ロックが含まれているかどうかを示します。

ポリシーを作成します。

ローカル スナップショット、セカンダリ ストレージへのレプリケーション、オブジェクト ストレージへのバックアップを管理するポリシーを作成できます。3-2-1 戦略の一部として、プライマリ ストレージ システム上のインスタンス、データベース、アプリケーション、または VM のスナップショットを作成します。

必要な**NetApp Console**ロール ストレージ ビューアー、バックアップおよびリカバリ スーパー管理者、バックアップおよびリカバリ バックアップ管理者。詳細はこちら["バックアップとリカバリの役割と権限"](#)。"[すべてのサービスに対するNetApp Consoleのアクセスロールについて学習します](#)"。

開始する前に

セカンダリ ストレージに複製し、ローカル スナップショットまたはリモートONTAPセカンダリ ストレージでスナップショット ロックを使用する予定の場合は、まずクラスタ レベルでONTAPコンプライアンス クロックを初期化する必要があります。これは、ポリシーでスナップショット ロックを有効にするための要件です。

これを行う方法については、以下を参照してください。"[ONTAPのコンプライアンスロックを初期化する](#)"。

スナップショットロック全般については、以下を参照してください。"[ONTAPのスナップショットロック](#)"。

手順

1. NetApp Backup and Recoveryメニューから、ポリシー を選択します。
2. [ポリシー] ページで、[新しいポリシーの作成] を選択します。

ポリシーページが表示されます。

3. * 詳細 * セクションに情報を入力します。
 - ワークロードの種類：*Kubernetes*を選択します。
 - ポリシー名を入力します。
 - エージェント リストからコンソール エージェントを選択します。
4. * Backup architecture * セクションに情報を入力します。リストからバックアップのデータフローを選択します：
 - **3-2-1** ファンアウト：プライマリストレージ（ディスク）からセカンダリストレージ（ディスク）を経てクラウド（オブジェクトストア）へ。ONTAP to ONTAP や ONTAP to オブジェクトストア構成など、異なるストレージシステム間でデータの複数のコピーを作成します。これは、クラウドハイパースケーラーのオブジェクトストア、またはプライベートオブジェクトストアのいずれかです。最適なデータ保護と災害復旧に最適です。このオプションは Amazon FSx for NetApp ONTAP では利用できません。
 - ディスク間：プライマリストレージ（ディスク）からセカンダリストレージ（ディスク）へ。ONTAP to ONTAP データ保護戦略は、2つのONTAPシステム間でデータをレプリケートし、高可用性とディザスタリカバリを確保します。これは通常、SnapMirrorを使用して実現され、同期レプリケーションと非同期レプリケーションの両方をサポートします。この方法により、データは最新の状態に保たれ、複数の場所で利用可能になり、強力なデータ保護が実現します。
 - ディスクからオブジェクトストレージ：プライマリストレージ（ディスク）からクラウド（オブジェクトストア）へ。ONTAP システムからオブジェクトストレージシステムにデータをレプリケートします。これは、クラウド ハイパースケーラー オブジェクト ストアまたは StorageGRID などのプライベート オブジェクト ストアです。この方法は、長期的なデータ保持とアーカイブに最適です。このオプションは、Amazon FSx for NetApp ONTAP では使用できません。
 - ローカルスナップショット：選択したボリューム上のローカルスナップショット。これにより、ワークロードが実行されている本番環境ボリュームの読み取り専用のポイントインタイムコピーが作成されます。ローカルスナップショットを使用すると、データの損失や破損から復旧できるだけでなく、ディザスタリカバリのためのバックアップを作成することもできます。
5. * ローカルスナップショット設定 * セクションの情報を入力します：
 - スナップショット スケジュールを選択するには、[スケジュールの追加] オプションを選択します。最大5つのスケジュールを設定できます。
 - スナップショットの頻度: 時間ごと、日ごと、週ごと、月ごと、または年ごとの頻度を選択します。年間頻度は Kubernetes ワークロードでは利用できません。
 - スナップショットの保持: 保持するスナップショットの数を入力します。
 - プロバイダー：Kubernetesアプリケーションリソースをホストするストレージプロバイダーを選択し、プロバイダーへの認証に必要な認証情報を入力します。

6. *セカンダリ設定*セクション（セカンダリストレージへのレプリケーション）の情報を入力します：
 - バックアップ: 時間ごと、日ごと、週ごと、月ごと、または年ごとの頻度を選択します。
 - バックアップ対象: バックアップの対象となるセカンダリ ストレージ上のターゲット システムを選択します。
 - 保持: 保持するスナップショットの数を入力します。
 - プロバイダー：Kubernetesアプリケーションリソースをホストするストレージプロバイダーを選択し、プロバイダーへの認証に必要な認証情報を入力します。
7. * Object store settings * セクション（オブジェクトストレージへのバックアップ）の情報を入力します。



表示されるフィールドは、選択したプロバイダーとアーキテクチャによって異なります。

- プロバイダー：オブジェクトストアのプロバイダーを選択し、適切なフィールドに認証情報を入力します（認証情報フィールドはプロバイダーによって異なります）。
- バックアップ対象: 登録済みのオブジェクト ストレージ対象を選択します。バックアップ環境内でターゲットにアクセスできることを確認します。
- **IPspace**: バックアップ操作に使用する IPspace を選択します。これは、複数の IPspace があり、どの IPspace をバックアップに使用するかを制御したい場合に便利です。
- スケジュール設定: ローカル スナップショットに設定されたスケジュールを選択します。スケジュールはローカル スナップショット スケジュールに従って設定されるため、削除することはできませんが、追加することはできません。
- 保持コピー数: 保持するスナップショットの数を入力します。
- 実行時間: データをオブジェクト ストレージにバックアップするためのONTAP転送スケジュールを選択します。
- オブジェクト ストアからアーカイブ ストレージにバックアップを階層化します: バックアップをアーカイブ ストレージ (AWS Glacier など) に階層化する場合は、階層オプションとアーカイブする日数を選択します。

ポリシーを編集する

バックアップアーキテクチャ、バックアップ頻度、保持ポリシー、およびポリシーに関するその他の設定を編集できます。Kubernetesワークロードポリシーについては、スケジュールと保持設定のみを編集できます。

ポリシーを編集するときに別の保護レベルを追加することはできませんが、保護レベルを削除することはできません。たとえば、ポリシーがローカル スナップショットのみを保護する場合は、セカンダリ ストレージへのレプリケーションやオブジェクト ストレージへのバックアップを追加できます。ローカル スナップショットとレプリケーションがある場合は、オブジェクト ストレージを追加できます。ただし、ローカル スナップショット、レプリケーション、およびオブジェクト ストレージがある場合は、これらのレベルのいずれかを削除することはできません。

オブジェクト ストレージにバックアップするポリシーを編集している場合は、アーカイブを有効にすることができます。

SnapCenterからリソースをインポートした場合、SnapCenterで使用されるポリシーとNetApp Backup and Recoveryで使用されるポリシーにいくつかの違いが生じる可能性があります。見る"[SnapCenterとNetApp Backup and Recoveryのポリシーの違い](#)"。

必要なNetApp Consoleロール

バックアップとリカバリのスーパー管理者。"[すべてのサービスに対するNetApp Consoleのアクセスロールについて学習します](#)"。

手順

1. NetApp Consoleで、保護 > バックアップとリカバリ に移動します。
2. ポリシー オプションを選択します。
3. 編集するポリシーを選択します。
4. *アクション*を選択します **...** アイコンをクリックし、[編集] を選択します。

ポリシーを削除する

不要になったポリシーは削除できます。



ワークロードに関連付けられているポリシーを削除することはできません。

手順

1. コンソールで、[保護] > [バックアップと復元] に移動します。
2. ポリシー オプションを選択します。
3. 削除するポリシーを選択します。
4. *アクション*を選択します **...** アイコンをクリックし、[削除] を選択します。
5. 操作を確認し、[削除] を選択します。

Backup and Recovery Web UIを使用してKubernetesアプリケーションを今すぐバックアップ

NetApp Backup and Recovery を使用すると、Web インターフェースを使用して Kubernetes アプリケーションを手動でバックアップできます。

必要なNetApp Consoleロール

組織管理者またはSnapCenter管理者。"[NetApp Backup and Recoveryのアクセス ロールについて学習します](#)"。
"[すべてのサービスに対するNetApp Consoleのアクセスロールについて学習します](#)"。

Web UI を使用して Kubernetes アプリケーションを今すぐバックアップする

将来のバックアップとスナップショットのベースラインを確立するため、または最新のデータが保護されていることを確認するために、Kubernetes アプリケーションのバックアップを手動で作成します。

手順

1. NetApp Backup and Recoveryで、**Inventory** を選択します。
2. Kubernetes インスタンスを選択し、[表示] を選択して、そのインスタンスに関連付けられているリソースを表示します。
3. *アプリケーション*タブを選択します。
4. アプリケーションのリストで、バックアップするアプリケーションを選択し、関連する [アクション] メニューを選択します。

5. *今すぐバックアップ*を選択します。
6. 正しいアプリケーション名が選択されていることを確認してください。
7. *バックアップ*を選択します。

結果

コンソールはアプリケーションのバックアップを作成し、バックアップとリカバリの 監視 領域に進行状況を表示します。バックアップは、アプリケーションに関連付けられた保護ポリシーに基づいて作成されます。

Backup and Recoveryのカスタム リソースを使用して、Kubernetesアプリケーションを今すぐバックアップ

NetApp Backup and Recovery では、カスタムリソース (CR) を使用して Kubernetes アプリケーションを手動でバックアップできます。

カスタムリソースを使用してKubernetesアプリケーションを今すぐバックアップ

将来のバックアップとスナップショットのベースラインを確立するため、または最新のデータが保護されていることを確認するために、Kubernetes アプリケーションのバックアップを手動で作成します。



クラスタを対象としたリソースは、アプリケーション定義で明示的に参照されている場合、またはいずれかのアプリケーション名前空間への参照がある場合、バックアップ、Snapshot、またはクローンに含まれます。

開始する前に

AWS セッショントークンの有効期限が、長時間実行される s3 バックアップ処理に十分であることを確認します。バックアップ処理中にトークンの有効期限が切れると、処理が失敗する可能性があります。

- 現在のセッショントークンの有効期限を確認する方法の詳細については、"[AWS API ドキュメント](#)"を参照してください。
- "[AWS IAM ドキュメント](#)"AWS リソースの認証情報の詳細については、こちらを参照してください。

カスタムリソースを使用してローカルスナップショットを作成する

Kubernetes アプリケーションのスナップショットを作成してローカルに保存するには、特定の属性を持つ Snapshot カスタム リソースを使用します。

手順

1. カスタムリソース (CR) ファイルを作成し、名前を付けます `local-snapshot-cr.yaml`。
2. 作成したファイルで、次の属性を設定します：
 - **metadata.name** : (必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
 - **spec.applicationRef** : スナップショットを作成するアプリケーションの Kubernetes 名。
 - **spec.appVaultRef** : (必須) スナップショットの内容 (メタデータ) を保存するAppVaultの名前。
 - **spec.reclaimPolicy** : (オプション) スナップショット CR が削除されたときに、スナップショットの AppArchive に対して何が起こるかを定義します。つまり、`Retain`に設定されている場合でも、スナップショットは削除されます。有効なオプション：

- Retain (デフォルト)
- 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. `local-snapshot-cr.yaml` ファイルに正しい値を入力したら、CRを適用します：

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

カスタム リソースを使用してアプリケーションをオブジェクト ストアにバックアップする

アプリケーションをオブジェクト ストアにバックアップするには、特定の属性を持つバックアップ CR を作成します。

手順

1. カスタムリソース (CR) ファイルを作成し、名前を `object-store-backup-cr.yaml` とします。
2. 作成したファイルで、次の属性を設定します：
 - **metadata.name**： (必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
 - **spec.applicationRef**： (必須) バックアップするアプリケーションの Kubernetes 名。
 - **spec.appVaultRef**： (必須、*spec.appVaultTargetsRef*とは相互排他) スナップショットとバックアップの保存に同じバケットを使用する場合、これはバックアップコンテンツを保存するAppVaultの名前です。
 - **spec.appVaultTargetsRef**： (必須、*spec.appVaultRef*とは相互排他) スナップショットとバックアップを保存するために別のバケットを使用する場合、これはバックアップコンテンツを保存するAppVaultの名前です。
 - **spec.dataMover**： (オプション、*Trident Protect*から移行したクラスタの場合は必須) バックアップ処理に使用するバックアップツールを示す文字列。このクラスタがTrident ProtectからBackup and Recoveryに移行されている場合、値は大文字と小文字が区別され、`CBS`である必要があります。
 - **spec.reclaimPolicy**： (オプション) Backup CR が削除されたときにバックアップコンテンツ (メタデータ / ボリュームデータ) に何が起こるかを定義します。使用可能な値：
 - Delete
 - Retain (デフォルト)

- **spec.cleanupSnapshot** : (必須) バックアップ CR によって作成された一時スナップショットが、バックアップ処理の完了後に削除されないようにします。推奨値 : `false`。

同じバケットを使用してスナップショットとバックアップを保存する場合の YAML の例 :

```
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
```

スナップショットとバックアップを保存するために異なるバケットを使用する場合の YAML の例 :

```
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. `object-store-backup-cr.yaml` ファイルに正しい値を入力したら、CRを適用します :

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

カスタムリソースを使用して **3-2-1** ファンアウト バックアップを作成する

3-2-1 ファンアウト アーキテクチャを使用してバックアップすると、バックアップがオブジェクトストアだけでなくセカンダリストレージにもコピーされます。3-2-1 ファンアウト バックアップを作成するには、特定の属性を持つバックアップ CR を作成します。

手順

1. カスタムリソース (CR) ファイルを作成し、名前を付けます `3-2-1-fanout-backup-cr.yaml`。

2. 作成したファイルで、次の属性を設定します：

- **metadata.name**：（必須）このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
- **spec.applicationRef**：（必須）バックアップするアプリケーションの Kubernetes 名。
- **spec.appVaultTargetsRef**：（必須）バックアップコンテンツを保存するAppVaultの名前。
- **spec.dataMover**：（オプション）バックアップ操作に使用するバックアップツールを示す文字列。値は大文字と小文字が区別され、`CBS`である必要があります。
- **spec.reclaimPolicy**：（オプション）Backup CR が削除されたときにバックアップコンテンツ（メタデータ / ボリュームデータ）に何が起こるかを定義します。使用可能な値：
 - Delete
 - Retain（デフォルト）
- **spec.cleanupSnapshot**：（必須）バックアップ CR によって作成された一時スナップショットが、バックアップ処理の完了後に削除されないようにします。推奨値： `false`。
- **spec.replicateSnapshot**：（必須）NetApp Backup and Recoveryにスナップショットをセカンダリストレージにレプリケートするように指示します。必要な値： `true`。
- **spec.replicateSnapshotReclaimPolicy**：（オプション）レプリケートされたSnapshotが削除されたときの動作を定義します。使用可能な値：
 - Delete
 - Retain（デフォルト）

YAMLの例：

```
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. `3-2-1-fanout-backup-cr.yaml` ファイルに正しい値を入力したら、CRを適用します：

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

サポートされているバックアップアノテーション

次の表では、バックアップ CR を作成するときを使用できる注釈について説明します。

注釈	タイプ	説明	デフォルト値
protect.trident.netapp.io/full-backup	string	バックアップを非増分にするかどうかを指定します。`true`に設定すると、非増分バックアップが作成されます。ベストプラクティスとしては、定期的にフルバックアップを実行し、フルバックアップの間に増分バックアップを実行して、リストアに伴うリスクを最小限に抑えることです。	"false"
protect.trident.netapp.io/snapshots-hot-completion-timeout	string	スナップショット処理全体が完了するまでに許容される最大時間。	"60m"
protect.trident.netapp.io/volume-snapshots-ready-to-use-timeout	string	ボリュームスナップショットが使用可能状態になるまでに許容される最大時間。	"30分"
protect.trident.netapp.io/ボリュームスナップショット作成タイムアウト	string	ボリュームスナップショットの作成に許可される最大時間。	"5m"
protect.trident.netapp.io/pvc-bind-timeout-sec	string	新しく作成されたPersistentVolumeClaims (PVC) が `Bound` フェーズに到達するまで待機する最大時間 (秒単位)。この時間を超えると操作は失敗します。	"1200" (20分)

Kubernetes アプリケーションを復元する

Web UI を使用して Kubernetes アプリケーションをリストアする

NetApp Backup and Recovery を使用すると、保護ポリシーで保護したアプリケーションを復元できます。アプリケーションを復元するには、少なくとも1つの復元ポイントが必要です。復元ポイントは、ローカルスナップショットまたはオブジェクトストアへのバックアップ（あるいはその両方）で構成されます。アプリケーションは、ローカルアーカイブ、セカンダリアーカイブ、またはオブジェクトストアアーカイブを使用して復元できます。

アプリケーションの復元ポイントの保護されたリソースを表示する

Backup and Recoveryで保護している各アプリケーションについて、特定の復元ポイント用にバックアップされたリソースを表示できます。

必要なNetApp Consoleロール

バックアップおよびリカバリビューア。"[NetApp Backup and Recoveryのアクセスロールについて学習します](#)". "[すべてのサービスに対するNetApp Consoleのアクセスロールについて学習します](#)".

手順

1. NetApp Backup and Recoveryで、インベントリ>*アプリケーション*を選択します。

2. アプリケーションの一覧からアプリケーションを選択し、アクションアイコン... > *表示と復元*を選択します。
3. 復元ポイントの一覧から復元ポイントを選択し、アクションアイコン... > *リソースを表示*を選択します。

リソースの一覧とその詳細が表示されます。名前空間またはクラスターごとによりリソースを表示し、将来の監査のためにリストを JSON ファイルとしてダウンロードできます。

4. 完了したら、*閉じる*を選択します。

Kubernetesアプリケーションを復元する

名前空間ベースのアプリケーションまたは仮想マシンベースのアプリケーションは、復元ポイントから復元できます。復元するには、すべてのリソースを復元するか、復元するリソースのサブセットを選択します。

開始する前に

Trident Protect を使用してバックアップされたアプリケーションを復元する場合は、Trident Protect がソースクラスターとデスティネーションクラスターの両方にインストールされていることを確認してください。

必要なNetApp Consoleロール

Backup and Recovery スーパー管理者または Backup and Recovery リストア管理者。"[NetApp Backup and Recoveryのアクセス ロールについて学習します](#)". "[すべてのサービスに対するNetApp Consoleのアクセスロールについて学習します](#)".

手順

1. NetApp Backup and Recoveryメニューで、*リストア*を選択します。
2. リストアから Kubernetes アプリケーションを選択し、そのアプリケーションの *表示と復元* を選択します。

復元ポイントのリストが表示されます。

3. 使用する復元ポイントの*Restore*ボタンを選択します。

データ復元ウィザードが開始され、_一般設定_ ページが表示されます。

4. 復元元のソースの場所を選択します。
5. クラスター リストアから宛先クラスターを選択します。
6. 元のネームスペースに復元するか、新しいネームスペースに復元するかを選択します。
7. 新しい名前空間に復元することを選択した場合は、以下の手順を実行してください。
 - a. 使用する宛先ネームスペースを入力してください。
 - b. 宛先アプリケーション名を入力してください。
 - c. 必要に応じて、*復元されたリソースのアプリケーションを作成しない*オプションを選択すると、アプリケーションのカスタムリソースオブジェクトを作成せずにリソースを復元できます。これにより、アプリケーションインベントリにおける不要なエントリが削減されます。

8. *次へ*を選択します。

Resource selection ページが表示されます。

9. アプリケーションに関連付けられているすべてのリソースを復元するか、フィルターを使用して復元する特定のリソースを選択するかを選択します。

すべてのリソース

- a. *すべてのリソースを復元*を選択します。

VMベースのアプリケーションをリストアする場合、NetApp Backup and Recoveryはリカバリポイント内のすべての仮想マシンを一覧表示します。

- b. *次へ*を選択します。

特定の名前空間ベースのアプリリソース

- a. * Selective resources * を選択し、選択したリソースをルールに基づいてフィルタリングするか、ネームスペースに基づいてフィルタリングするかを選択します。

リソース選択方法	手順
ルールを使用してリソースをフィルタリング	<ol style="list-style-type: none">i. *ルール*タブを選択します。ii. リソース フィルターの動作を選択します。 *含める*を選択すると、選択したリソースが復元されます。*除外*を選択した場合、選択したリソースは復元されません。iii. リソースを選択するためのフィルターを定義するルールを追加するには、[ルールの追加]を選択します。リソースをフィルタリングするには、少なくとも1つのルールが必要です。 各ルールは、リソースの名前空間、ラベル、グループ、バージョン、種類などの基準でフィルタリングできます。iv. 各ルールを保存するには、[保存]を選択します。v. 必要なルールをすべて追加したら、*リソースを表示*を選択して、フィルター条件に一致するバックアップアーカイブ内の利用可能なリソースを表示します。
リストからリソースを手動で選択する	<ol style="list-style-type: none">i. * Custom * タブを選択します。ii. 対応するリソースを表示するには、*Namespace-Scoped*または*Cluster-Scoped*を選択します。 バックアップとリカバリでは、リカバリポイント内のすべてのリソースが一覧表示されます。iii. 復元操作に含めるリソースを選択してください。



表示されるリソースは、クラスター上に現在存在するリソースです。

b. 完了したら、* Next *を選択します。

特定のVMベースのアプリケーションリソース

a. *選択的リソース*を選択します。

b. 次のいずれかを実行します。

- 仮想マシン全体を復元するには、「仮想マシン」タブを選択します。

バックアップとリカバリでは、リカバリポイントにあるすべての仮想マシンが一覧表示されます。リストア操作に含める仮想マシンを選択できます。

- 個々の永続ボリューム要求を復元するには、「永続ボリューム要求」タブを選択します。

バックアップとリカバリでは、リカバリポイント内のすべての永続ボリューム要求が一覧表示されます。復元操作に含める永続ボリューム要求を選択できます。

c. 完了したら、* Next *を選択します。

Destination settings ページが表示されます。

10. *Destination settings*セクションを展開し、デフォルトのストレージクラス、別のストレージクラスのいずれかに復元するか、別のクラスタに復元する場合は、ストレージクラスをデスティネーション クラスタにマッピングするかを選択します。
11. 別のストレージ クラスにリストアすることを選択した場合は、各ソース ストレージ クラスに一致するデスティネーション ストレージ クラスを選択します。
12. オプションで、Trident Protect を使用して作成されたバックアップまたは Snapshot をリストアする場合は、リストア処理のストレージバケットとして使用される AppVault の詳細を表示します。環境または AppVault のステータスに変更がある場合は、* App Vault を同期*を選択して詳細を更新します。



Trident Protectを使用して作成されたバックアップまたはスナップショットの復元を容易にするために、Kubernetesクラスタ上にAppVaultを作成する必要がある場合は、"[Trident Protect AppVaultオブジェクトを使用してバケットを管理する](#)"を参照してください。

13. 必要に応じて、[復元スクリプト]セクションを展開し、[Postscript] オプションを有効にして、復元操作の完了後に実行される実行フック テンプレートを選択します。必要に応じて、スクリプトに必要な引数を入力し、ラベル セレクタを追加して、リソース ラベルに基づいてリソースをフィルタします。
14. 必要に応じて、*リソース変換*セクションを展開して、復元プロセス中にリソース属性を追加、削除、または変更します。次に、以下の手順を実行します：



PersistentVolumeClaimsと名前空間の変更は現在サポートされていません。

- a. 修飾子を変更するには、*リソース変換*オプションを有効にしてください。
- b. よく使う修飾子設定をすばやく適用するには、*テンプレート*リストからテンプレートを選択してください。このリストには、一般的なシナリオに対応した定義済みテンプレートと、ユーザーが作成したカスタムテンプレートが含まれています。



グローバル"設定"エリアでリソース変換テンプレートを作成します。

- c. 変更するリソースを指定するには、リソースグループ、バージョン、種類、および名前を入力してください。
- d. *操作*リストから操作を選択して、リソースに対して実行する操作を指定します。
- e. 変更する特定のキーのJSONパスを入力します。
- f. 該当する場合は、使用する新しい値を入力してください。*Value*フィールドは、特定の操作 (*Add* や *Replace* など) でのみ表示されます。
- g. 必要に応じて、さらにリソース変換を追加してください。

15. 完了したら、* Restore * を選択します。

カスタムリソースを使用した **Kubernetes** アプリケーションのリストア

カスタム リソースを使用して、スナップショットまたはバックアップからアプリケーションを復元できます。アプリケーションを同じクラスタに復元する場合、既存のスナップショットからの復元の方が高速になります。

- アプリケーションを復元すると、アプリケーションに対して設定されているすべての実行フックもアプリとともに復元されます。復元後の実行フックが存在する場合、復元操作の一部として自動的に実行されます。
- バックアップから別の名前空間または元の名前空間への復元は、qtree ボリュームでサポートされています。ただし、スナップショットから別の名前空間または元の名前空間への復元は、qtree ボリュームではサポートされていません。
- 詳細設定を使用して復元操作をカスタマイズできます。詳細については、["高度なカスタムリソースのリストア設定を使用する"](#)を参照してください。



バックアップを別の名前空間に復元する

BackupRestore CR を使用して異なる名前空間にバックアップを復元すると、NetApp Backup and Recovery はアプリケーションを新しい名前空間に復元し、復元されたアプリケーションのアプリケーション CR を作成します。復元されたアプリケーションを保護するには、オンデマンドバックアップまたはスナップショットを作成するか、保護スケジュールを確立します。

- 既存のリソースを含む別の名前空間にバックアップをリストアしても、バックアップ内のリソースと名前を共有するリソースは変更されません。バックアップ内のすべてのリソースをリストアするには、ターゲット名前空間を削除して再作成するか、バックアップを新しい名前空間にリストアします。
- CR を使用して新しい名前空間に復元する場合は、CR を適用する前に、デスティネーション名前空間を手動で作成する必要があります。NetApp Backup and Recovery では、CLI を使用する場合にのみ名前空間が自動的に作成されます。



開始する前に

AWS セッショントークンの有効期限が、長時間実行される s3 復元処理に十分であることを確認します。復元処理中にトークンの有効期限が切れると、処理が失敗する可能性があります。

- 現在のセッショントークンの有効期限を確認する方法の詳細については、"[AWS API ドキュメント](#)"を参照してください。
- "[AWS IAM ドキュメント](#)"AWS リソースの認証情報の詳細については、こちらを参照してください。



Kopia をデータムーバーとして使用してバックアップを復元する場合、オプションで CR に注釈を指定して、Kopia が使用する一時ストレージの動作を制御できます。"[Kopiaのドキュメント](#)"設定できるオプションの詳細については、こちらを参照してください。

手順

1. カスタムリソース (CR) ファイルを作成し、名前を付けます `trident-protect-backup-restore-cr.yaml`。
2. 作成したファイルで、次の属性を設定します：
 - **metadata.name** : (必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
 - **spec.appArchivePath** : AppVault内でバックアップコンテンツが保存されるパス。このパスを見つけるには、次のコマンドを使用できます：

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

- **spec.appVaultRef** : (必須) バックアップコンテンツが保存されるAppVaultの名前。
- **spec.namespaceMapping** : リストア処理のソースネームスペースからデスティネーションネームスペースへのマッピング。`my-source-namespace`と`my-destination-namespace`を環境の情報に置き換えます。

```
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. (オプション) 復元するアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルでマークされたリソースを含めるか除外するフィルタリングを追加します：



Trident Protect は、選択したリソースとの関係に基づいて、一部のリソースを自動的に選択します。たとえば、永続ボリュームクレームリソースを選択し、それに関連付けられたポッドがある場合、Trident Protect は関連付けられているポッドも復元します。

- **resourceFilter.resourceSelectionCriteria** : (フィルタリングに必須) `Include`または`Exclude`を

使用して、resourceMatchersで定義されたリソースを含めるか除外します。含めるまたは除外するリソースを定義するには、以下のresourceMatchersパラメータを追加します：

- **resourceFilter.resourceMatchers**：resourceMatcherオブジェクトの配列。この配列に複数の要素を定義すると、それらはOR演算として一致し、各要素内のフィールド（グループ、種類、バージョン）はAND演算として一致します。
 - **resourceMatchers[].group**：（オプション）フィルタリングするリソースのグループ。
 - **resourceMatchers[].kind**：（オプション）フィルタリングするリソースの種類。
 - **resourceMatchers[].version**：（オプション）フィルタリングするリソースのバージョン。
 - **resourceMatchers[].names**：（オプション）フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前。
 - **resourceMatchers[].namespaces**：（オプション）フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前空間。
 - **resourceMatchers[].labelSelectors**：（オプション） ["Kubernetesドキュメント"](#)で定義されているリソースのKubernetesメタデータ.nameフィールドのラベルセレクタ文字列。例：
"trident.netapp.io/os=linux"。

例：

```
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. `trident-protect-backup-restore-cr.yaml` ファイルに正しい値を入力したら、CRを適用します：

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

バックアップを元の名前空間に復元する

いつでもバックアップを元の名前空間に復元できます。

開始する前に

AWS セッショントークンの有効期限が、長時間実行される s3 復元処理に十分であることを確認します。復元処理中にトークンの有効期限が切れると、処理が失敗する可能性があります。

- 現在のセッショントークンの有効期限を確認する方法の詳細については、"[AWS API ドキュメント](#)"を参照してください。
- "[AWS IAM ドキュメント](#)"AWS リソースの認証情報の詳細については、こちらを参照してください。



Kopia をデータムーバーとして使用してバックアップを復元する場合、オプションで CR に注釈を指定して、Kopia が使用する一時ストレージの動作を制御できます。"[Kopiaのドキュメント](#)"設定できるオプションの詳細については、こちらを参照してください。

手順

1. カスタムリソース (CR) ファイルを作成し、名前を `trident-protect-backup-ipr-cr.yaml` とします。
2. 作成したファイルで、次の属性を設定します：
 - **metadata.name** : (必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
 - **spec.appArchivePath** : AppVault内でバックアップコンテンツが保存されるパス。このパスを見つけるには、次のコマンドを使用できます：

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

- **spec.appVaultRef** : (必須) バックアップコンテンツが保存されるAppVaultの名前。

例：

```
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. (オプション) 復元するアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルでマークされたリソースを含めるか除外するフィルタリングを追加します：



Trident Protect は、選択したリソースとの関係に基づいて、一部のリソースを自動的に選択します。たとえば、永続ボリュームクレームリソースを選択し、それに関連付けられたポッドがある場合、Trident Protect は関連付けられているポッドも復元します。

- **resourceFilter.resourceSelectionCriteria** : (フィルタリングに必須) `Include` または `Exclude` を

使用して、resourceMatchersで定義されたリソースを含めるか除外します。含めるまたは除外するリソースを定義するには、以下のresourceMatchersパラメータを追加します：

- **resourceFilter.resourceMatchers**：resourceMatcherオブジェクトの配列。この配列に複数の要素を定義すると、それらはOR演算として一致し、各要素内のフィールド（グループ、種類、バージョン）はAND演算として一致します。
 - **resourceMatchers[].group**：（オプション）フィルタリングするリソースのグループ。
 - **resourceMatchers[].kind**：（オプション）フィルタリングするリソースの種類。
 - **resourceMatchers[].version**：（オプション）フィルタリングするリソースのバージョン。
 - **resourceMatchers[].names**：（オプション）フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前。
 - **resourceMatchers[].namespaces**：（オプション）フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前空間。
 - **resourceMatchers[].labelSelectors**：（オプション） ["Kubernetesドキュメント"](#)で定義されているリソースのKubernetesメタデータ.nameフィールドのラベルセレクタ文字列。例：
"trident.netapp.io/os=linux"。

例：

```
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. `trident-protect-backup-ipr-cr.yaml`ファイルに正しい値を入力したら、CRを適用します：

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

バックアップを別のクラスタにリストアする

元のクラスタに問題がある場合は、バックアップを別のクラスタにリストアできます。



- Kopia をデータムーバーとして使用してバックアップを復元する場合、オプションで CR に注釈を指定して、Kopia が使用する一時ストレージの動作を制御できます。"[Kopiaのドキュメント](#)"設定できるオプションの詳細については、[こちらを参照してください](#)。
- CR を使用して新しい名前空間にリストアする場合は、CR を適用する前に、デスティネーション名前空間を手動で作成する必要があります。

開始する前に

次の前提条件が満たされていることを確認してください：

- デスティネーション クラスタに Trident Protect がインストールされている。
- デスティネーション クラスタは、バックアップが保存されているソース クラスタと同じAppVaultのバケットパスにアクセスできます。
- AWS セッション トークンの有効期限が、長時間実行される復元操作に十分であることを確認します。復元操作中にトークンの有効期限が切れると、操作が失敗する可能性があります。
 - 現在のセッショントークンの有効期限を確認する方法の詳細については、"[AWS API ドキュメント](#)"を参照してください。
 - AWS リソースの認証情報の詳細については、"[AWSのドキュメント](#)"を参照してください。

手順

1. Trident Protect CLIプラグインを使用して、デスティネーション クラスタ上のAppVault CRの可用性を確認します：

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



アプリケーションのリストアに使用する名前空間がデスティネーション クラスタに存在することを確認します。

2. デスティネーション クラスタから利用可能なAppVaultのバックアップコンテンツを表示します：

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

このコマンドを実行すると、AppVault内の利用可能なバックアップが表示されます。これには、元のクラスタ、対応するアプリケーション名、タイムスタンプ、アーカイブパスが含まれます。

出力例：

```

+-----+-----+-----+-----+
+-----+-----+-----+-----+
| 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. AppVault名前とアーカイブパスを使用して、アプリケーションをデスティネーション クラスタに復元します：
4. カスタムリソース (CR) ファイルを作成し、名前を付けます `trident-protect-backup-restore-cr.yaml`。
5. 作成したファイルで、次の属性を設定します：
 - **metadata.name**： (必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
 - **spec.appVaultRef**： (必須) バックアップコンテンツが保存されるAppVaultの名前。
 - **spec.appArchivePath**： AppVault内でバックアップコンテンツが保存されるパス。このパスを見つけるには、次のコマンドを使用できます：

```

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

```



BackupRestore CR が利用できない場合は、手順 2 に記載されているコマンドを使用してバックアップの内容を表示できます。

- **spec.namespaceMapping**： リストア処理のソース名前スペースからデスティネーション名前スペースへのマッピング。`my-source-namespace`と`my-destination-namespace`を環境の情報に置き換えます。

例：

```
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. `trident-protect-backup-restore-cr.yaml` ファイルに正しい値を入力したら、CRを適用します：

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

スナップショットを別のネームスペースにリストアする

カスタム リソース (CR) ファイルを使用して、スナップショットからデータを別の名前空間または元のソース名前空間に復元できます。SnapshotRestore CR を使用してスナップショットを別の名前空間に復元すると、Backup and Recovery はアプリケーションを新しい名前空間に復元し、復元されたアプリケーションのアプリケーション CR を作成します。復元されたアプリケーションを保護するには、オンデマンド バックアップまたはスナップショットを作成するか、保護スケジュールを設定します。



- SnapshotRestoreは `spec.storageClassMapping` 属性をサポートしていますが、ソース ストレージ クラスとデスティネーション ストレージ クラスが同じストレージ バックエンドを使用する場合のみです。異なるストレージ バックエンドを使用する `StorageClass` に復元しようとする、復元処理は失敗します。
- CR を使用して新しいネームスペースにリストアする場合は、CR を適用する前に、デスティネーションネームスペースを手動で作成する必要があります。

開始する前に

AWS セッショントークンの有効期限が、長時間実行される s3 復元処理に十分であることを確認します。復元処理中にトークンの有効期限が切れると、処理が失敗する可能性があります。

- 現在のセッショントークンの有効期限を確認する方法の詳細については、"[AWS API ドキュメント](#)"を参照してください。
- "[AWS IAM ドキュメント](#)"AWS リソースの認証情報の詳細については、こちらを参照してください。

手順

1. カスタムリソース (CR) ファイルを作成し、名前を `trident-protect-snapshot-restore-cr.yaml` とします。
2. 作成したファイルで、次の属性を設定します：
 - **metadata.name** : (必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
 - **spec.appVaultRef** : (必須) スナップショットの内容が保存されるAppVaultの名前。

- **spec.appArchivePath** : AppVault内のパスで、スナップショットの内容が保存される場所。このパスを見つけるには、次のコマンドを使用できます :

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

- **spec.namespaceMapping** : リストア処理のソースネームスペースからデスティネーションネームスペースへのマッピング。`my-source-namespace`と`my-destination-namespace`を環境の情報に置き換えます。

```
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. (オプション) 復元するアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルでマークされたリソースを含めるか除外するフィルタリングを追加します :



Trident Protect は、選択したリソースとの関係に基づいて、一部のリソースを自動的に選択します。たとえば、永続ボリュームクレームリソースを選択し、それに関連付けられたポッドがある場合、Trident Protect は関連付けられているポッドも復元します。

- **resourceFilter.resourceSelectionCriteria** : (フィルタリングに必須) `Include`または`Exclude`を使用して、resourceMatchersで定義されたリソースを含めるか除外します。含めるまたは除外するリソースを定義するには、以下のresourceMatchersパラメータを追加します :
 - **resourceFilter.resourceMatchers** : resourceMatcherオブジェクトの配列。この配列に複数の要素を定義すると、それらはOR演算として一致し、各要素内のフィールド (グループ、種類、バージョン) はAND演算として一致します。
 - **resourceMatchers[].group** : (オプション) フィルタリングするリソースのグループ。
 - **resourceMatchers[].kind** : (オプション) フィルタリングするリソースの種類。
 - **resourceMatchers[].version** : (オプション) フィルタリングするリソースのバージョン。
 - **resourceMatchers[].names** : (オプション) フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前。
 - **resourceMatchers[].namespaces** : (オプション) フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前空間。
 - **resourceMatchers[].labelSelectors** : (オプション) "[Kubernetesドキュメント](#)"で定義されているリソースのKubernetesメタデータ.nameフィールドのラベルセレクト文字列。例 :
"trident.netapp.io/os=linux"。

例：

```
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. `trident-protect-snapshot-restore-cr.yaml` ファイルに正しい値を入力したら、CRを適用します：

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

スナップショットを元の名前空間に復元する

いつでもスナップショットを元の名前空間にリストアできます。



ローカルスナップショットからのVMベースのアプリケーションのインプレース復元（元の名前空間および元のクラスタへの復元）は、現在サポートされていません。

開始する前に

AWS セッショントークンの有効期限が、長時間実行される s3 復元処理に十分であることを確認します。復元処理中にトークンの有効期限が切れると、処理が失敗する可能性があります。

- 現在のセッショントークンの有効期限を確認する方法の詳細については、["AWS API ドキュメント"](#)を参照してください。
- ["AWS IAM ドキュメント"](#) AWS リソースの認証情報の詳細については、こちらを参照してください。

手順

1. カスタムリソース（CR）ファイルを作成し、`trident-protect-snapshot-restore-cr.yaml` という名前を付けます。
2. 作成したファイルで、次の属性を設定します：
 - **metadata.name**：（必須）このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。

- **spec.appVaultRef** : (必須) スナップショットの内容が保存されるAppVaultの名前。
- **spec.appArchivePath** : AppVault内のパスで、スナップショットの内容が保存される場所。このパスを見つけるには、次のコマンドを使用できます :

```
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. (オプション) 復元するアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルでマークされたリソースを含めるか除外するフィルタリングを追加します :



Trident Protect は、選択したリソースとの関係に基づいて、一部のリソースを自動的に選択します。たとえば、永続ボリュームクレームリソースを選択し、それに関連付けられたポッドがある場合、Trident Protect は関連付けられているポッドも復元します。

- **resourceFilter.resourceSelectionCriteria** : (フィルタリングに必須) `Include` または `Exclude` を使用して、resourceMatchersで定義されたリソースを含めるか除外します。含めるまたは除外するリソースを定義するには、以下のresourceMatchersパラメータを追加します :
 - **resourceFilter.resourceMatchers** : resourceMatcherオブジェクトの配列。この配列に複数の要素を定義すると、それらはOR演算として一致し、各要素内のフィールド (グループ、種類、バージョン) はAND演算として一致します。
 - **resourceMatchers[].group** : (オプション) フィルタリングするリソースのグループ。
 - **resourceMatchers[].kind** : (オプション) フィルタリングするリソースの種類。
 - **resourceMatchers[].version** : (オプション) フィルタリングするリソースのバージョン。
 - **resourceMatchers[].names** : (オプション) フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前。
 - **resourceMatchers[].namespaces** : (オプション) フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前空間。
 - **resourceMatchers[].labelSelectors** : (オプション) "[Kubernetesドキュメント](#)"で定義されているリソースのKubernetesメタデータ.nameフィールドのラベルセクタ文字列。例 : "trident.netapp.io/os=linux"。

例 :

```
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. `trident-protect-snapshot-ipr-cr.yaml` ファイルに正しい値を入力したら、CRを適用します：

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

高度なカスタムリソースのリストア設定を使用する

注釈、名前空間設定、ストレージ オプションなどの詳細設定を使用して、特定の要件を満たすように復元操作をカスタマイズできます。

リストアおよびフェイルオーバー処理中の名前空間のアノテーションとラベル

リストアおよびフェイルオーバー時には、宛先の名前空間ラベルと注釈がソースに合わせて更新されます（ソースのキーは宛先のキーに追加または上書きされますが、宛先にのみ存在するキーは変更されません）。



Red Hat OpenShiftでは、名前空間アノテーションは、リストアされたポッドが正しいセキュリティコンテキスト制約と権限を受け取ることを保証し、ボリュームへのアクセスや権限エラーなしでの実行を可能にするため重要です。詳細については、"[OpenShiftセキュリティコンテキスト制約ドキュメント](#)"を参照してください。

Kubernetes環境変数を設定する

```
RESTORE_SKIP_NAMESPACE_ANNOTATIONS
```

復元またはフェイルオーバーの前に、特定の宛先名前空間の注釈が上書きされないようにします。例：

```
helm upgrade trident-protect -n trident-protect netapp-trident-protect/trident-protect \
  --set-string
restoreSkipNamespaceAnnotations="{<annotation_key_to_skip_1>,<annotation_key_to_skip_2>}" \
  --reuse-values
```



リストアまたはフェイルオーバー中に、`restoreSkipNamespaceAnnotations`および`restoreSkipNamespaceLabels`で指定された名前空間アノテーションとラベルは、リストアまたはフェイルオーバー処理から除外されます。これらの設定は、Helmの初回インストール時に必ず構成してください。詳細については、["追加のTrident Protectヘルムチャート設定を構成する"](#)を参照してください。

Helm を `--create-namespace` フラグとともに使用してソースアプリケーションをインストールした場合、Trident Protect は名前ラベルをデスティネーション名前スペースにコピーします。ラベルの値がソース名前スペース名と一致する場合は、デスティネーション名前スペース名に置き換えられます。一致しない場合は、そのまま残されます。

例

次の例では、異なるラベルと注釈を持つソース名前スペースと宛先名前スペースを示し、操作の前後の宛先名前スペースを表示することで、キーがどのように追加、マージ、または上書きされるかを説明しています。

リストアまたはフェイルオーバー処理の前に

次の表は、リストアまたはフェイルオーバー処理前のサンプルのソース名前スペースとデスティネーション名前スペースの状態を示しています：

名前スペース	アノテーション	ラベル
名前スペースns-1 (ソース)	<ul style="list-style-type: none"> • annotation.one/key : "updatedvalue" • annotation.two/key : "true" 	<ul style="list-style-type: none"> • environment=production • コンプライアンス=hipaa • name=ns-1
名前スペースns-2 (宛先)	<ul style="list-style-type: none"> • annotation.one/key : "true" • annotation.three/key : "false" 	<ul style="list-style-type: none"> • ロール=database

リストア処理後

次の表は、復元またはフェイルオーバー操作後の宛先名前スペースの状態を示しています。いくつかのキーが追加され、いくつかは上書きされ、`name`ラベルは宛先名前スペースと一致するように更新されました：

ネームスペース	アノテーション	ラベル
ネームスペースns-2 (宛先)	<ul style="list-style-type: none"> • annotation.one/key: "updatedvalue" • annotation.two/key: "true" • annotation.three/key: "false" 	<ul style="list-style-type: none"> • name=ns-2 • コンプライアンス=hipaa • environment=production • ロール=database

サポートされているフィールド

このセクションでは、復元操作に使用できる追加のフィールドについて説明します。

ストレージクラスのマッピング

`spec.storageClassMapping``属性は、ソース アプリケーションに存在するストレージクラスからターゲット クラス上の新しいストレージクラスへのマッピングを定義します。異なるストレージクラスを持つクラス間でアプリケーションを移行する場合や、BackupRestore操作のストレージバックエンドを変更する場合にこれを使用できます。

例：

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

サポートされているアノテーション

このセクションでは、システム内のさまざまな動作を構成するためにサポートされているアノテーションを一覧表示します。ユーザーがアノテーションを明示的に設定しない場合、システムはデフォルト値を使用します。

注釈	タイプ	説明	デフォルト値
protect.trident.netapp.io/data-mover-timeout-sec	string	データムーバー処理が停止する最大時間（秒単位）。	"300"
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	string	Kopia コンテンツ キャッシュの最大サイズ制限（メガバイト単位）。	"1000"

注釈	タイプ	説明	デフォルト値
protect.trident.netapp.io/pvc-bind-timeout-sec	string	新しく作成されたPersistentVolumeClaims (PVC) が `Bound` フェーズに到達するまで待機する最大時間 (秒単位)。この時間を超えると処理は失敗します。環境：すべてのリストアCRタイプ (BackupRestore、BackupInplaceRestore、SnapshotRestore、SnapshotInplaceRestore)。ストレージバックエンドまたはクラスターで処理に時間がかかることが多い場合は、より大きい値を使用してください。	「1200」 (20分)

カスタムリソースを使用して復元する際にリソースを変更します

リソース変換機能を使用すると、リソースの復元中にリソースを変更できます。これは、復元したバージョンを元のバージョンと異なるものにする必要がある場合に役立ちます。たとえば、仮想マシンを別のネットワークに復元する際に、そのIPアドレスを変更する場合などです。また、["Web UIを使用してリソースを復元する際に、リソースを変更します"](#)。

必須**NetApp Console**ロール Backup and Recovery super adminまたはBackup and Recovery restore admin。["NetApp Backup and Recoveryのアクセス ロールについて学習します"](#)。"すべてのサービスに対するNetApp Consoleのアクセスロールについて学習します"。

リソース変更の仕組み

``transformations`` フィールドは、``SnapshotRestore``、``BackupRestore``、``AppMirrorRelationship``、およびその他のリストアリソースで、リストアプロセス中にKubernetesリソースを変更できるようにします。これは、ホスト名、レジストリURL、リソース制限、または環境変数を変更することにより、アプリケーションや仮想マシンを新しいクラスターに適応させる際に役立ちます。

リソース変換では、["RFC 6902"](#) JSON Patch操作と ["RFC 6901"](#) JSON Pointerパスを使用して、Kubernetesリソース内の特定のフィールドをターゲットにして変更します。

リソース変換を含む復元オブジェクトの基本的な構造は以下のとおりです：

```

apiVersion: protect.trident.netapp.io/v1
kind: SnapshotRestore
metadata:
  name: my-restore
  namespace: target-namespace
spec:
  appVaultRef: my-vault
  appArchivePath: /path/to/snapshot
  namespaceMapping:
    - source: source-ns
      destination: target-ns
  transformations:
    - resource:
        kind: ConfigMap           # Required: resource kind
        group: ""                # Optional: API group (empty for core
resources)
        version: ""              # Optional: API version
        name: ""                  # Optional: specific resource name
      operations:
        - op: replace            # Operation type
          path: "/data/key"      # JSON Pointer path
          value: "new-value"     # New value (for add/replace/test)

```

サポートされているリソース

以下の条件に一致するリソースに対して、リソース変換を使用できます：

- kind (必須) : Kubernetesリソースの種類 (例: ConfigMap、Deployment、Pod)
- group (オプション) : API グループ (例: apps、route.openshift.io) - コアリソースの場合は省略
- version (オプション) : APIバージョン (例: v1、v1beta1)
- name (オプション) : 名前で指定した特定のリソースにのみ適用します



PersistentVolumeClaimsと名前空間の変更は現在サポートされていません。

サポートされている操作

リソースを変更するには、以下の操作を使用できます。

- add : リソースに値を追加します。
- copy : あるパスから別のパスへ値をコピーします。
- move : リソース内で値を移動します。
- remove : リソースから値を削除します。

- `replace` : リソース内の値を置換します。
- `test` : 操作を実行する前にテストします。

リソースに値を追加する

`add` 操作を使用して、指定されたパスに新しいフィールドまたは値を追加します。オブジェクトや配列にデータを追加できます。次の例では、Deploymentリソースにノードセレクタを追加します：

```
transformations:
  - resource:
      kind: Deployment
    operations:
      - op: add
        path: "/spec/template/spec/nodeSelector"
        value:
          "topology.kubernetes.io/zone": "us-east-1a"
          disktype: "ssd"
```

コマンドラインを使用してこの変換を実行するには、次のコマンドを使用します：

```
tridentctl-protect --transformation
'apps,v1,Deployment:add{"path":"/spec/template/spec/nodeSelector","value":
{"topology.kubernetes.io/zone":"us-east-1a","disktype":"ssd"}}'
```

リソース内の値をコピーする

`copy` 操作を使用して、同じリソース内のあるパスから別のパスに値をコピーします。ソースは変更されません。次の例では、ConfigMapオブジェクトのデータキーを複製します：

```
transformations:
  - resource:
      kind: ConfigMap
    operations:
      - op: copy
        from: "/data/source-key"
        path: "/data/backup-key"
```

コマンドラインを使用してこの変換を実行するには、次のコマンドを使用します：

```
tridentctl-protect --transformation
',v1,ConfigMap:copy{"from":"/data/source-key","path":"/data/backup-key"}'
```

リソース内で値を移動する

`move` 操作を使用して、同じリソース内であるパスから別のパスに値を移動します。ソースが削除され、値が宛先に配置されます。次の例では、ConfigMapオブジェクトのデータキーの名前を変更します：

```
transformations:
- resource:
  kind: ConfigMap
  operations:
  - op: move
    from: "/data/OLD_KEY"
    path: "/data/NEW_KEY"
```

コマンドラインを使用してこの変換を実行するには、次のコマンドを使用します：

```
tridentctl-protect --transformation
',v1,ConfigMap:move{"from":"/data/OLD_KEY","path":"/data/NEW_KEY"}'
```

リソースから値を削除する

`remove` 操作を使用して、指定されたパスにあるフィールドまたは値を削除します。次の例では、ConfigMapリソースから注釈を削除します：

```
transformations:
- resource:
  kind: ConfigMap
  operations:
  - op: remove
    path: "/metadata/annotations/kubect1.kubernetes.io~1last-applied-configuration"
```



上記の例のパスでは、`~1`は`/`のJSONポインタースケープシーケンスです。

コマンドラインを使用してこの変換を実行するには、次のコマンドを使用します：

```
tridentctl-protect --transformation
',v1,ConfigMap:remove{"path":"/metadata/annotations/kubect1.kubernetes.io~
l1last-applied-configuration"}'
```

リソース内の値を置き換える

`replace`操作を使用して、指定されたパスにあるリソース内の既存の値を置き換えます。JSONパスは既に存在している必要があります。次の例は、Routeオブジェクトのホスト名を変更します：

```
transformations:
- resource:
  kind: Route
  group: route.openshift.io
  operations:
  - op: replace
    path: "/spec/host"
    value: "prod.example.com"
```

コマンドラインを使用してこの変換を実行するには、次のコマンドを使用します：

```
tridentctl-protect --transformation
'route.openshift.io,v1,Route:replace{"path":"/spec/host","value":"prod.exa
mple.com"}'
```

リソースの変更をテストする

パスの値が想定される値と一致するかどうかをテストするには、`test`オペレーションを使用します。テストが失敗した場合、変更全体がロールバックされます。次の例では、`environment`が`staging`である場合にのみ`database-host`を更新します：

```
transformations:
- resource:
  kind: ConfigMap
  operations:
  - op: test
    path: "/data/environment"
    value: "staging"
  - op: replace
    path: "/data/database-host"
    value: "prod-db.example.com"
```

コマンドラインを使用してこの変換を実行するには、次のコマンドを使用します：

```
tridentctl-protect --transformation
',v1,ConfigMap:test{"path":"/data/environment","value":"staging"},replace{
"path":"/data/database-host","value":"prod-db.example.com"}'
```

Kubernetes クラスターを管理する

NetApp Backup and Recovery を使用すると、Kubernetes クラスターを検出して管理できるため、クラスターによってホストされるリソースを保護できます。

必要なNetApp Consoleロール

組織管理者またはSnapCenter管理者。["NetApp Backup and Recoveryのアクセス ロールについて学習します"](#)。["すべてのサービスに対するNetApp Consoleのアクセスロールについて学習します"](#)。



Kubernetesクラスターを発見するには、以下を参照してください。["Kubernetes ワークロードを発見する"](#)。

Kubernetes クラスターの情報を編集する

名前を変更する必要がある場合は、クラスターを編集できます。

手順

1. NetApp Backup and Recoveryで、インベントリ > クラスタ を選択します。
2. クラスターのリストで、編集するクラスターを選択し、関連する [アクション] メニューを選択します。
3. *クラスターの編集*を選択します。
4. クラスター名に必要な変更を加えます。クラスター名は、検出プロセス中にHelmコマンドで使用した名前と一致する必要があります。
5. *完了*を選択します。

Kubernetes クラスターを削除する

Kubernetes クラスターの保護を停止するには、保護を無効にして関連付けられているアプリケーションを削除し、NetApp Backup and Recoveryからクラスターを削除します。NetApp Backup and Recovery、クラスターまたはそのリソースは削除されません。NetAppNetApp Consoleインベントリからクラスターが削除されるだけです。

手順

1. NetApp Backup and Recoveryで、インベントリ > クラスタ を選択します。
2. クラスターのリストで、編集するクラスターを選択し、関連する [アクション] メニューを選択します。
3. *クラスターの削除*を選択します。
4. 確認ダイアログボックスの情報を確認し、「削除」を選択します。

Trident Protectのアップグレード

Trident Protect 26.05以降を実行しているKubernetesクラスタの場合、NetApp Backup and RecoveryからTrident Protectを直接アップグレードできます。アップグレードオプションは、新しいバージョンが利用可能な場合にのみ表示されます。

手順

1. NetApp Backup and Recoveryで、インベントリ > クラスタ を選択します。
2. アップグレードするKubernetesクラスタを選択し、アクションアイコン **...** > *クラスタのアップグレード* を選択します。
3. アップグレードダイアログで、以下の操作を行います：
 - a. リストから、アップグレードするバージョンを選択します。
 - b. * Upgrade * を選択します。

Kubernetes アプリケーションを管理する

NetApp Backup and Recovery を使用すると、Kubernetes アプリケーションと関連リソースの保護を解除したり削除したりできます。

必要なNetApp Consoleロール

組織管理者またはSnapCenter管理者。"[NetApp Backup and Recoveryのアクセス ロールについて学習します](#)"。
。["すべてのサービスに対するNetApp Consoleのアクセスロールについて学習します"](#)。

Kubernetes アプリケーションの保護を解除する

アプリケーションを保護する必要がなくなった場合は、保護を解除できます。アプリケーションの保護を解除すると、NetApp Backup and Recovery はアプリケーションの保護を停止しますが、関連するすべてのバックアップとスナップショットは保持されます。



保護操作が実行中の間は、アプリケーションの保護を解除することはできません。操作が完了するまで待つか、回避策として、実行中の保護操作が使用している [リストアポイントを削除する](#) を削除します。その後、アプリケーションの保護を解除できます。

手順

1. NetApp Backup and Recoveryで、**Inventory** を選択します。
2. Kubernetes インスタンスを選択し、[表示] を選択して、そのインスタンスに関連付けられているリソースを表示します。
3. *アプリケーション*タブを選択します。
4. アプリケーションのリストで、保護を解除するアプリケーションを選択し、関連する [アクション] メニューを選択します。
5. *保護解除*を選択します。
6. 通知を読み、準備ができたなら [保護解除] を選択します。

Kubernetes アプリケーションを削除する

不要になったアプリケーションを削除します。NetApp Backup and Recovery は保護を停止し、削除されたアプリケーションのバックアップとスナップショットをすべて削除します。

手順

1. NetApp Backup and Recoveryで、**Inventory** を選択します。
2. Kubernetes インスタンスを選択し、[表示] を選択して、そのインスタンスに関連付けられているリソースを表示します。
3. *アプリケーション*タブを選択します。
4. アプリケーションのリストで、削除するアプリケーションを選択し、関連する [アクション] メニューを選択します。
5. *削除*を選択します。
6. スナップショットとバックアップの削除 を有効にすると、アプリケーションのすべてのスナップショットとバックアップが削除されます。



これらのスナップショットとバックアップを使用してアプリケーションを復元することはできなくなります。


7. 操作を確認し、[削除] を選択します。

Kubernetes アプリケーションの復元ポイントを削除する

アプリケーションの保護を解除する必要があり、現在保護操作が実行中の場合は、アプリケーションの復元ポイントを削除する必要がある場合があります。

手順

1. NetApp Backup and Recoveryメニューで、*リストア*を選択します。
2. リストから Kubernetes アプリケーションを選択し、そのアプリケーションの *表示と復元* を選択します。

復元ポイントのリストが表示されます。
3. 削除するリカバリポイントを選択し、アクションアイコン  *リカバリポイントの削除* を選択して削除します。

Kubernetes ワークロード用のNetApp Backup and Recovery 実行フック テンプレートを管理する

実行フックは、管理された Kubernetes アプリケーションのデータ保護操作で実行されるカスタム アクションです。たとえば、実行フックを使用してスナップショットの前にデータベース トランザクションを一時停止し、スナップショット後に再開することで、アプリケーション整合性のあるスナップショットを作成します。実行フック テンプレートを作成するときは、フックの種類、実行するスクリプト、およびターゲット コンテナのフィルターを指定します。テンプレートを使用して、実行フックをアプリケーション

ンにリンクします。



NetApp Backup and Recoveryは、データ保護中にKubeVirtなどのアプリケーションのファイルシステムをフリーズおよびアンフリーズします。Trident Protectのドキュメントを使用して、この動作をグローバルに無効にすることも、特定のアプリケーションに対して無効にすることもできます：

- すべてのアプリケーションでこの動作を無効にするには、"[KubeVirt VM によるデータ保護](#)"。
- 特定のアプリケーションでこの動作を無効にするには、"[アプリケーションを定義する](#)"。

必要なNetApp Consoleロール

組織管理者またはSnapCenter管理者。["NetApp Backup and Recoveryのアクセス ロールについて学習します"](#)。["すべてのサービスに対するNetApp Consoleのアクセスロールについて学習します"](#)。

実行フックの種類

NetApp Backup and Recovery は、実行できるタイミングに基づいて、次のタイプの実行フックをサポートしています。

- 事前スナップショット
- スナップショット後
- バックアップ前
- バックアップ後
- 復元後

実行順序

データ保護操作が実行されると、実行フック イベントが次の順序で発生します。

1. 適用可能なカスタム操作前実行フックは、適切なコンテナで実行されます。複数のカスタム事前操作フックを作成できますが、その実行順序は保証されず、構成もできません。
2. 該当する場合、ファイルシステムのフリーズが発生します。
3. データ保護操作が実行されます。
4. 該当する場合、凍結されたファイルシステムは凍結解除されます。
5. NetApp Backup and Recovery は、適切なコンテナで適用可能なカスタム操作前実行フックを実行します。複数のカスタム操作後フックを作成できますが、その実行順序は保証されず、構成もできません。

同じタイプのフックを複数作成した場合、それらの実行順序は保証されません。異なるタイプのフックは常に指定された順序で実行されます。たとえば、さまざまな種類のフックがすべて含まれる構成の実行順序は次のとおりです。

1. スナップショット前のフックが実行されました
2. スナップショット後のフックが実行されました
3. バックアップ前のフックが実行されました

4. バックアップ後のフックが実行されました



実行フック スクリプトを本番環境で有効にする前にテストします。「`kubectl exec`」を使用してスクリプトをテストし、アプリを一時的な名前空間に複製して復元することで、スナップショットとバックアップを検証します。



スナップショット前の実行フックによって Kubernetes リソースが追加、変更、または削除された場合、それらの変更はスナップショットまたはバックアップと、その後のすべての復元操作に含まれます。

カスタム実行フックに関する重要な注意事項

アプリの実行フックを計画するときは、次の点を考慮してください。

- 実行フックは、アクションを実行するためにスクリプトを使用する必要があります。複数の実行フックが同じスクリプトを参照できます。
- 実行フックは、実行可能なシェル スクリプトの形式で記述する必要があります。
- スクリプトのサイズは 96 KB に制限されています。
- 実行フックの設定と一致基準は、スナップショット、バックアップ、または復元操作に適用可能なフックを決定するために使用されます。



実行フックにより、アプリケーションの機能が低下したり、無効になったりする可能性があります。カスタムフックをできるだけ早く実行するようにします。関連する実行フックを使用してバックアップまたはスナップショット操作を開始したが、その後キャンセルした場合でも、バックアップまたはスナップショット操作がすでに開始されている場合は、フックは引き続き実行できます。つまり、バックアップ後の実行フックで使用されるロジックでは、バックアップが完了したと想定することはできません。

実行フックフィルター

アプリケーションの実行フックを追加または編集するときに、実行フックにフィルターを追加して、フックが一致するコンテナを管理できます。フィルターは、すべてのコンテナで同じコンテナ イメージを使用するが、各イメージを異なる目的で使用する可能性があるアプリケーション (Elasticsearch など) に役立ちます。フィルターを使用すると、実行フックが必ずしもすべての同一コンテナではなく一部のコンテナで実行されるシナリオを作成できます。単一の実行フックに対して複数のフィルターを作成すると、それらは論理 AND 演算子で結合されます。実行フックごとに最大 10 個のアクティブ フィルターを設定できます。

実行フックに追加する各フィルターは、正規表現を使用してクラスター内のコンテナを照合します。フックがコンテナに一致すると、フックはそのコンテナ上で関連付けられたスクリプトを実行します。フィルターの正規表現では正規表現 2 (RE2) 構文が使用されますが、一致リストからコンテナを除外するフィルターの作成はサポートされていません。NetApp Backup and Recoveryが実行フックフィルターの正規表現でサポートする構文については、以下を参照してください。"[正規表現2 \(RE2\) 構文のサポート](#)"。



復元またはクローン操作後に実行される実行フックに名前空間フィルターを追加し、復元またはクローンのソースと宛先が異なる名前空間にある場合、名前空間フィルターは宛先の名前空間にのみ適用されます。

実行フックの例

訪問 ["NetApp Verda GitHub プロジェクト"](#) Apache Cassandra や Elasticsearch などの一般的なアプリの実際の実行フックをダウンロードします。また、例を参照したり、独自のカスタム実行フックを構成するためのアイデアを入手したりすることもできます。



Verda GitHub リポジトリ内のスクリプトは現状のまま提供されており、NetApp による公式サポートの対象外です。Trident Protect および Backup and Recovery 内の実行フックフレームワークのみが、NetApp によって公式にサポートされています。

実行フックテンプレートを作成する

アプリケーションでのデータ保護操作の前後にアクションを実行するために使用できるカスタム実行フックテンプレートを作成できます。



ここで作成するテンプレートは、Kubernetes ワークロードを保護する場合にのみ使用できません。

手順

1. コンソールで、[保護] > [バックアップと復元] に移動します。
2. *設定*タブを選択します。
3. 実行フック テンプレート セクションを展開します。
4. *実行フックテンプレートの作成*を選択します。
5. 実行フックの名前を入力します。
6. 必要に応じて、フックの種類を選択します。たとえば、復元後フックは復元操作が完了した後に実行されます。
7. スクリプト テキスト ボックスに、実行フック テンプレートの一部として実行する実行可能シェル スクリプトを入力します。必要に応じて、*スクリプトのアップロード*を選択して、代わりにスクリプト ファイルをアップロードすることもできます。
8. *作成*を選択します。

テンプレートを作成すると、実行フック テンプレート セクションのテンプレートのリストに表示されます。

NetApp Backup and RecoveryでKubernetesワークロードの保護レポートを作成および管理する

NetApp Backup and Recovery では、Kubernetes ワークロードの保護レポートを作成して、成功したバックアップと失敗したバックアップの数、バックアップの種類、クラスタの正常性情報など、保護の状態と詳細を表示します。

必須**NetApp Console** ロール Backup and Recovery super admin、Backup and Recovery backup admin、または Backup and Recovery restore admin。"[バックアップとリカバリの役割と権限](#)"について説明します "[すべてのサービスに対するNetApp Consoleのアクセスロールについて学習します](#)"。

保護レポートを作成する

保護レポートを作成して、クラスタの保護ステータスを表示します。


手順

1. NetApp Backup and Recoveryメニューから、レポート オプションを選択します。
2. *レポートの作成*を選択します。
3. レポート範囲の詳細を入力します:
 - レポート名: レポートの一意の名前を入力します。
 - レポート タイプ: アカウント別またはワークロード別のレポートのどちらが必要かを選択します (リストから Kubernetes を選択します)。
 - クラスタを選択: ワークロード別に選択した場合は、レポートを生成するクラスタをリストから選択し、*Accept*を選択します。すべてのクラスタのレポートを生成するには、*Select all*を選択します。
4. レポート範囲を入力: レポートに過去 1 日、過去 7 日間、過去 30 日間、前四半期、または前年度のデータを含めるかどうかを選択します。
5. レポート設定の詳細を入力します: レポートを 1 回だけ実行するか、定期的なレポート生成をスケジュールするかを選択します。スケジュールされたレポートの場合は、繰り返しの頻度を選択し、開始日を選択します。
 - a. 電子メール配信の詳細を入力します: (スケジュールされたレポートの場合のみ) レポートを電子メールで配信する場合は、スケジュールされたレポートを受信する1つ以上の電子メールアドレスを入力します。設定ページで電子メール通知を構成します。電子メール通知の設定の詳細については、"[設定を構成する](#)"。
6. *作成*を選択します。

保護レポートをダウンロードする

生成された保護レポートを JSON ファイルまたは PDF ドキュメントとしてダウンロードして、表示および共有できます。

手順

1. NetApp Backup and Recoveryメニューから、レポート オプションを選択します。
2. *レポート*ページで、*レポート*メニューを選択して、生成された保護レポートのリストを表示します。
3. ダウンロードするレポートで、Actionsアイコン  > *Download*を選択します。
 - レポートを JSON 形式でダウンロードするには、**Download JSON** を選択します。
 - レポートを PDF ドキュメントとしてダウンロードするには、* Download PDF * を選択します。

保護レポートを表示する

NetApp Backup and Recovery内で保護レポートのインタラクティブな詳細を素早く表示できます。ジョブの概要情報、データ保護ステータス、構成の詳細などを確認できます。

手順

1. NetApp Backup and Recoveryメニューから、レポート オプションを選択します。
2. *レポート*ページで、*レポート*メニューを選択して、生成された保護レポートのリストを表示します。
3. 表示するレポートについて、アクションアイコン...> *レポートを表示*を選択します。

レポートの詳細が表示されます。

保護レポートを削除する

不要になった保護レポートを削除します。

手順

1. NetApp Backup and Recoveryメニューから、レポート オプションを選択します。
2. *レポート*ページで、*レポート*メニューを選択して、生成された保護レポートのリストを表示します。
3. 削除するレポートについて、アクションアイコン...> *削除*を選択します。
4. *削除*を選択して操作を確認します。

著作権に関する情報

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および/または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。