

NetBackup™ for Kubernetes 管理者ガイド

リリース 11.2

最終更新日: 2026-06-23

法的通知と登録商標

Copyright © 2026 Cohesity, Inc. All rights reserved.

© 2026 Cohesity, Inc. All Rights Reserved. Cohesity、Cohesity ロゴ、その他の Cohesity マークは、Cohesity, Inc. の米国における、および/または国際的な商標です。ここで提供される情報は、Cohesity の機密情報および専有情報であり、(a) 意図された受信者により、(b) 有効なライセンスを受けた Cohesity ソフトウェアおよびサービスと共にのみ使用することができます。Cohesity ライセンスの条項については、www.cohesity.com/agreements を参照してください。

本書は、現状のままで提供されるものであり、その商品性、特定目的への適合性、または不侵害の暗黙的な保証を含む、明示的あるいは暗黙的な条件、表明、および保証はすべて免責されるものとします。ただし、これらの免責が法的に無効であるとされる場合を除きます。Cohesity およびその関連会社は、本書の提供、パフォーマンスまたは使用に関連する付随的または間接的損害に対して、一切責任を負わないものとします。本書に記載の情報は、予告なく変更される場合があります。

Cohesity サポート

Cohesity サポートへのお問い合わせ

Cohesity サポートケースを作成するには、いくつかの方法があります。

- [Cohesity サポート](#)に移動してベリタスのナレッジベースで検索するか、電話で問い合わせます (米国およびカナダ: 1-855-9CO-HESI (926-4374)、オプション 2)。
- [Cohesity サポートポータル](#)にログインして、新しいケースを作成します。
- Cohesity UI の (?) アイコンをクリックし、[サポートポータル (Support Portal)]を選択します。

サポートおよびサービスアシスタンス

最初に、サービスおよびサポートを契約したサービスプロバイダにお問い合わせください。Cohesity と直接連携している場合や、製品の保証や資格、修理の価格設定、またはテクニカルサポートに関連する質問がある場合は、以下のオプションを参照してください。

- 製品の問題に対するソリューションや、提案およびベストプラクティスを見つけるには、[Cohesity ナレッジベース](#)を参照してください。
- [Cohesity サポートポータル](#)にログインして、新しいケースを作成します。
- オープン中のケースを監視するには、ポータルにログインし、ホームページの[ケース (Cases)]タブをクリックします。このページには、すべてのケースの状態で更新が表示されるはずですが、個々のケースの状態を表示することもできます。

パートナーのハードウェアで実行されている Cohesity ソフトウェア

認定されたサードパーティ製ハードウェアで動作する Cohesity ソフトウェアの場合、次のサポートワークフローが適用されます。

1. ハードウェアの問題であると判断できない場合は、まず Cohesity サポートにお問い合わせください。

メモ: Cohesity はパートナーのハードウェアの交換要求を処理できません。

2. Cohesity サポートは問題の優先順位を判断します。ソフトウェアの問題の場合、Cohesity サポートは引き続きこの問題に取り組めます。
3. ハードウェアまたはファームウェアの問題であるかその可能性がある場合、Cohesity は問題に関する情報をお客様に提供し、適切なパートナーに対するサポートチケットを発行するようお客様に依頼します。
4. Cohesity サポートは必要に応じて、パートナーとお客様との三者通話に参加できます。
5. お客様がパートナーのケースの進捗状況を Cohesity サポートに通知します。

目次

第 1 章	NetBackup for Kubernetes の概要	8
	概要	8
	Kubernetes 用の NetBackup サポート機能	9
第 2 章	NetBackup Kubernetes Operator の配備と構成	12
	NetBackup Kubernetes Operator を配備するための前提条件	12
	NetBackup Kubernetes Operator でのサービスパッケージの配備	14
	Kubernetes 作業負荷のポート要件	17
	NetBackup Kubernetes Operator のアップグレード	18
	NetBackup Kubernetes Operator の削除	20
	NetBackup Kubernetes データムーバーの構成	21
	Kubernetes 用の NetBackup 保護の自動構成	22
	Kubernetes 作業負荷のカスタマイズ	27
	スナップショットからのバックアップ操作とバックアップからのリストア操 作の前提条件	32
	Kubernetes でサポートされる DTE クライアント設定	36
	datamover プロパティのカスタマイズ	36
	短縮名の付いた NetBackup サーバーのトラブルシューティング	38
	datamover ボードのスケジュールメカニズムのサポート	39
	アクセラレータストレージクラスの検証	48
第 3 章	NetBackup Kubernetes Operator での証明書の 配備	49
	Kubernetes Operator での証明書の配備	49
	ホスト ID ベースの証明書操作の実行	50
	ECA 証明書操作の実行	56
	証明書の種類の識別	62
第 4 章	Kubernetes 資産の管理	65
	Kubernetes クラスターの追加	65
	設定を行う	66
	Kuberentes リソース形式のリソース制限の変更	67

	自動検出の間隔の構成	68
	権限の構成	69
	資産のクリーンアップ	69
	資産への保護の追加	69
	マルウェアのスキャン	70
	作業負荷の種類ごとの資産	71
第 5 章	Kubernetes インテリジェントグループの管理	73
	インテリジェントグループについて	73
	インテリジェントグループの作成	74
	インテリジェントグループの削除	76
	インテリジェントグループの編集	76
第 6 章	Kubernetes ポリシーの管理	77
	ポリシーの作成	77
第 7 章	Kubernetes 資産の保護	79
	インテリジェントグループの保護	79
	インテリジェントグループからの保護の削除	80
	バックアップスケジュールの構成	80
	バックアップオプションの構成	82
	バックアップの構成	83
	A.I.R. (自動イメージレプリケーション) と複製の構成	85
	ストレージユニットの構成	88
	ボリュームモードのサポート	89
	アプリケーションの一貫したバックアップの構成	89
第 8 章	イメージグループの管理	94
	イメージグループについて	94
	イメージの期限切れ	94
	イメージのコピー	95
第 9 章	NetBackup でのランチャ管理クラスタの保護	97
	自動構成を使用した NetBackup へのランチャ管理 RKE クラスタの追加	97
	NetBackup でのランチャ管理 RKE クラスタの手動での追加	99

第 10 章	Kubernetes 資産のリカバリ	103
	リカバリポイントの検索と検証	103
	スナップショットからのリストア	104
	バックアップコピーからのリストア	107
	スナップショットまたはバックアップからの仮想マシンのリストア	110
第 11 章	増分バックアップとリストアについて	113
	Kubernetes の増分バックアップとリストアのサポート	113
第 12 章	アクセラレータベースのバックアップの有効化	117
	Kubernetes 作業負荷に対する NetBackup アクセラレータのサポートにつ いて	117
	プライマリサーバーにあるトラックログのディスク容量の制御	119
	ストレージクラスの動作がアクセラレータに与える影響	119
	アクセラレータ強制再スキャンについて	120
	アクセラレータバックアップのエラーに関する警告と考えられる理由	121
第 13 章	Kubernetes での FIPS モードの有効化	122
	Kubernetes での FIPS (連邦情報処理標準) モードの有効化	122
第 14 章	Openshift Virtualization のサポートについて	125
	Openshift Virtualization のサポート	125
	アプリケーションの一貫性がある仮想マシンのバックアップ	126
	仮想化のトラブルシューティング	128
第 15 章	Kubernetes の問題のトラブルシューティング	129
	プライマリサーバーのアップグレード時のエラー: NBCheck が失敗する	130
	古いイメージのリストア時のエラー: 操作が失敗する	130
	永続ボリュームのリカバリ API でのエラー	131
	リストア中のエラー: ジョブの最終状態で一部が失敗していると表示される	131
	同じ名前空間でのリストア時のエラー	132
	datamover ポッドが Kubernetes のリソース制限を超過	132
	リストア時のエラー: 高負荷のクラスターでジョブが失敗する	134
	特定のクラスター用に作成されたカスタムの Kubernetes の役割でジョブを表 示できない	134
	OperatorHub からインストールされたアプリケーションのリストア時に、選択 されていない空の PVC が Openshift によって作成される	135

Kubernetes ノードで PID の制限を超えると NetBackup Kubernetes	
Operator が応答しなくなる	136
NetBackup Kubernetes 10.1 におけるクラスタの編集集中のエラー	137
サイズの大きい PVC のバックアップまたはリストアが失敗する	137
名前空間ファイルモードの PVC を別のファイルシステムにリストアすると部	
分的に失敗する	138
バックアップコピーからのリストアがイメージの不整合エラーで失敗する	
.....	138
NetBackup プライマリサーバー、メディアサーバー、Kubernetes サーバー	
間の接続性チェック	139
トラックログに利用可能な領域がない場合のアクセラレータバックアップ中	
のエラー	139
トラックログ PVC の作成エラーによるアクセラレータバックアップ中のエラー	
.....	139
無効なアクセラレータストレージクラスによるアクセラレータバックアップ中の	
エラー	140
トラックログポッドの起動中に発生したエラー	140
トラックログ PVC 操作のためのデータムーバーインスタンスの設定に失敗	
する	140
configmap からトラックログのストレージクラスを読み取る際のエラー	141
異なるクラスタに単一の VM をリストアするときにリストア操作が失敗する	
.....	141
backupservercert の作成時に Kubernetes 作業負荷の自動配備が失	
敗する	141
Kubernetes 作業負荷の自動配備がタイムアウトエラーで失敗する	142
エラー「2890 K8s のデータのリストア操作が失敗しました (2890 K8s restore	
data operation failed)」が表示され、即座に終了リストア戦略が失敗	
しました。	142

NetBackup for Kubernetes の概要

この章では以下の項目について説明しています。

- [概要](#)
- [Kubernetes 用の NetBackup サポート機能](#)

概要

NetBackup Web UI は、名前空間の形式で、Kubernetes アプリケーションのバックアップとリストアの機能を提供します。Kubernetes クラスタ内の保護可能な資産は NetBackup 環境内で自動的に検出され、管理者は必要なスケジュール、バックアップ、保持の各設定を含む 1 つ以上の保護計画を選択できます。

NetBackup Web UI では、次の操作を実行できます。

- 保護のための Kubernetes クラスタの追加
- 検出された名前空間の表示
- 役割の権限の管理
- リソース制限を設定してインフラとネットワークの負荷を最適化
- Kubernetes 資産を保護するための保護とインテリジェントグループの管理
- 名前空間と永続ボリュームを同じ、または代替の Kubernetes クラスタにリストアします。
- バックアップおよびリストア操作の監視
- 単一の仮想マシンのリストア
- イメージの有効期限、イメージのインポートおよびイメージのコピー操作

Kubernetes 用の NetBackup サポート機能

表 1-1 NetBackup for Kubernetes

機能	説明
自動 NetBackup Kubernetes エージェント の構成	Kubernetes クラスタが追加され、ストレージクラスとボリュームスナップショットクラスなどの構成が追加され、自動配備がサポートされた状態でデータムーバーの構成を実行できます。
NetBackup RBAC (役割 ベースのアクセス制御) と の統合	NetBackup Web UI は RBAC の役割を提供し、どの NetBackup ユーザーが NetBackup の Kubernetes 操作を管理できるかを制御します。ユーザーは Kubernetes 操作を管理するために NetBackup 管理者である必要はありません。
ライセンス	容量ベースのライセンス
保護計画	次の利点があります。 ■ 単一の保護計画を使用して、複数の Kubernetes 名前空間を保護します。複数のクラスタに 資産を分散できます。 ■ Kubernetes 資産を保護するために、Kubernetes コマンドを知る必要はありません。
Kubernetes 資産のインテ リジェントな管理	NetBackup は自動的に、Kubernetes クラスタ内の名前空間、永続ボリューム、永続ボリューム要 求などを検出します。また、手動検出を実行できます。資産が検出されると、Kubernetes 作業負 荷管理者は、資産を保護するために 1 つ以上の保護計画を選択できます。 メモ: AIR (自動イメージレプリケーション) の場合、ターゲットプライマリサーバーのインポート済 み名前空間にはインポートされた時間が [最終検出 (Last Discovered)] 時間として表示されま す。
Kubernetes 固有のクレデ ンシャル	クラスタの認証と管理に使用する Kubernetes サービスアカウント。
検出 ■ 完全検出 ■ 増分検出	[今すぐ検出 (Discover now)] オプションを使用した検出は常に完全検出です。 新しいクラスタが NetBackup に追加されたときの検出は常に完全検出です。 Kubernetes クラスタが追加されると、自動検出サイクルがトリガされ、Kubernetes クラスタで利用 可能なすべての資産が検出されます。その日最初の自動検出は完全検出で、以降の自動検出 は増分検出です。

機能	説明
バックアップ機能 ■ スナップショットのみのバックアップ ■ スナップショットからのバックアップ	バックアップでは次の機能を利用できます。 ■ バックアップは、NetBackup サーバーによって中央サイトから完全に管理されます。管理者は、さまざまな Kubernetes クラスタで、名前空間の自動的な無人バックアップをスケジュールできます。 ■ NetBackup Web UI は、1 つのインターフェースからの名前空間のバックアップとリストアをサポートします。 ■ 完全バックアップのバックアップスケジュールの構成。 ■ 手動バックアップとスナップショットのみのバックアップ。 ■ バックアップのパフォーマンスを向上させるための各クラスタのリソースのスロットル。 ■ NetBackup はスナップショット方式を使用して Kubernetes 名前空間のバックアップを実行し、リカバリ時間目標を短縮できます。
リストア機能 ■ スナップショットからのリストア ■ バックアップコピーからのリストア	リストアでは次の機能を利用できます。 ■ Kubernetes 名前空間と永続ボリュームを異なる場所にリストアします。 ■ 並列リストアジョブを含むバックアップコピーからのリストアを使用して、異なる Kubernetes クラスタプレーバーにリストアします。
クライアント側のデータ重複排除のサポート	Kubernetes でクライアント側のデータ重複排除のサポート機能が有効になっています。 詳しくは、『<i>NetBackup 重複排除ガイド</i>』の「クライアント側の重複排除について」セクションを参照してください。
自動イメージレプリケーション (AIR)	1 つの NetBackup Kubernetes クラスタで生成されたバックアップを、1 つ以上のターゲット NetBackup ドメインのストレージにレプリケートできます。これは AIR とも呼ばれます。他の NetBackup ドメインのストレージにバックアップをレプリケートする機能。 自動イメージレプリケーション (A.I.R) は、すべてのスケジュール形式でサポートされます。
ステートフルアプリケーションの保護	永続ボリュームを使用して状態を保持する Kubernetes アプリケーションを保護できます。次の機能をサポートする CSI (Container Storage Interface) プロバイダにおけるモードファイルシステムまたはブロック (あるいはその両方) の PVC (永続ボリューム要求) のバックアップとリストア: ■ PVC スナップショット機能 ■ NFS (Network File System) または他の非ブロックストレージに基づく PVC ボリュームプロビジョニング ■ ボリュームが混在する (VolumeMode: ファイルシステムとブロック) 名前空間のバックアップとリストアは、NetBackup 10.3 以降でサポートされます。
インポートと検証	インポートは 2 段階の操作です。第 1 段階では、指定したメディア上のバックアップに対するカタログエントリが再作成されます。第 2 段階のインポートが完了すると、それらのイメージによってバックアップされたファイルのカタログエントリが作成されます。 検証: NetBackup では、NetBackup カタログに記録されたものと内容を比較して、バックアップの内容を検証できます。

機能	説明
Red Hat プラットフォーム用の FIPS (連邦情報処理標準) サポート	Red Hat プラットフォームの NetBackup Kubernetes は、FIPS 準拠の通信をサポートします。
Kubernetes のアクセラレータバックアップのサポート	NetBackup は、Kubernetes 作業負荷のアクセラレータバックアップをサポートし、バックアップ時間を短縮します。
マルウェアスキャンのサポート	NetBackup バージョン 10.4 以降では、Kubernetes の作業負荷を介して Kubernetes 資産でマルウェアをスキャンするためのサポートが提供されます。
Kubernetes 作業負荷に対する OpenShift Virtualization のサポート	NetBackup バージョン 10.4.1 以降では、Kubernetes クラスタで実行されている 1 台以上の仮想マシンで、名前空間に対するバックアップとリストアがサポートされます。

警告: このガイドでは、Velero を参照用実装として使用し、Kubernetes アプリケーションと永続的なデータ保護を紹介します。Cohesity はベンダー中立を維持しており、複数のソリューション (Velero、Kasten K10、Trilio、Commvault、マネージドサービス (Backup for GKE など)) およびサポートされている CSI ベースの VolumeSnapshots を使用して保護できます。プラットフォーム、ストレージ、コンプライアンスのニーズに合わせてツールを選択してください。

NetBackup Kubernetes Operator の配備と構成

この章では以下の項目について説明しています。

- [NetBackup Kubernetes Operator](#) を配備するための前提条件
- [NetBackup Kubernetes Operator](#) でのサービスパッケージの配備
- [Kubernetes](#) 作業負荷のポート要件
- [NetBackup Kubernetes Operator](#) のアップグレード
- [NetBackup Kubernetes Operator](#) の削除
- [NetBackup Kubernetes](#) データムーバーの構成
- [Kubernetes](#) 用の [NetBackup](#) 保護の自動構成
- [Kubernetes](#) 作業負荷のカスタマイズ
- 短縮名の付いた [NetBackup](#) サーバーのトラブルシューティング
- [datamover](#) ポッドのスケジュールメカニズムのサポート
- アクセラレータストレージクラスの検証

NetBackup Kubernetes Operator を配備するための前提条件

[NetBackup Kubernetes Operator](#) を配備する前に、[Helm Chart](#) をインストールし、永続ボリューム用の領域を用意する必要があります。

[Helm](#) の最新バージョンをインストールするには、次のコマンドを実行します。

1. \$ curl -fsSL -o get_helm.sh <https://raw.githubusercontent.com/helm>
2. \$ chmod 700 get_helm.sh
3. \$./get_helm.sh

メモ: NetBackup を配備する各クラスタにオペレータを配備する必要があります。

新しい Helm Chart をインストールするには

- 1 名前空間内のすべての Helm Chart を一覧表示するには、次のコマンドを実行します。

```
- helm list -n <namespace>
```

- 2 古いプラグインをアンインストールするには、次のコマンドを実行します。

```
■ helm uninstall <plugin-name> -n <namespace>
```

- 3 新しいプラグインをインストールするには、次のコマンドを実行します。

```
■ helm install <plugin-name> <chart-path> -n <namespace>
```

Helm Chart とツリー構造のレイアウトを次に示します。

```
netbackupkops-helm-chart/
├─ charts
├─ Chart.yaml
├─ templates
│   └─ deployment.yaml
│   └─ _helpers.tpl
└─ values.yaml
```

ディレクトリ構造:

```
tar --list -f netbackupkops-10.3.tar.gz
veritas_license.txt
netbackupkops.tar
netbackupkops-helm-chart/
netbackupkops-helm-chart/Chart.yaml
netbackupkops-helm-chart/values.yaml
netbackupkops-helm-chart/.helmignore
netbackupkops-helm-chart/templates/
netbackupkops-helm-chart/templates/deployment.yaml
netbackupkops-helm-chart/templates/_helpers.tpl
netbackupkops-helm-chart/charts/
```

NetBackup Kubernetes Operator でのサービスパッケージの配備

Helm Chart の構成

Helm Chart を使用して、NetBackup Kubernetes Operator を配備できます。

NetBackup Kubernetes Operator をアップグレードするには、Helm Chart をアップグレードする必要があります。

メモ: 新しいプラグインをインストールする前に、古いプラグインをアンインストールする必要があります。

NetBackup Kubernetes Operator を配備するには:

- 1 Cohesity テクニカルサポート Web サイト (<https://www.veritas.com/content/support>) から tar パッケージをダウンロードします。
- 2 ホームディレクトリにパッケージを抽出します。netbackupkops-helm-chart フォルダは、ホームディレクトリに存在する必要があります。
- 3 すべてのクラスタコンテキストを一覧表示するには、コマンド `kubectl config get-contexts` を実行します。
- 4 オペレータサービスを配備するクラスタに切り替えるには、次のコマンドを実行します。

`kubectl config use-context <cluster-context-name>`
- 5 現在のディレクトリをホームディレクトリに変更するには、コマンド `cd ~` を実行します。
- 6 NetBackup は、OCI 標準に準拠したコンテナイメージレジストリをサポートしています。オペレータとデータムーバイメージをプッシュする任意のツールを使用できます。

プライベート Docker レジストリを使用している場合は、この手順の指示に従って、NetBackup 名前空間に **Secret** nb-docker-cred を作成します。それ以外の場合は、次の手順にスキップします。

- プライベート Docker レジストリにログインするには、コマンド `docker login -u <user name><repo-name>` を実行します。
ログイン後、認証トークンを含む `config.json` ファイルが作成または更新されます。`config.json` ファイルを表示するには、コマンド `cat ~/.docker/config.json` を実行します。
出力は次のようになります。

```
{
  "auths": {
    "https://index.docker.io/v1/": {
      "auth": "c3R...zE2"
    }
  }
}
```

- **NetBackup** 名前空間で `netbackupkops-docker-cred` という名前の **Secret** を作成するには、次のコマンドを実行します。

```
kubectl create secret generic netbackupkops-docker-cred \
--from-file=.dockerconfigjson=.docker/config.json \
--type=kubernetes.io/dockerconfigjson -n netbackup
```

Secret を作成する名前空間は任意に指定できます。

- **NetBackup** 名前空間で **Secret** `netbackupkops-docker-cred` が作成されたかどうかを確認するには、次のコマンドを実行します。

```
kubectl get secrets -n netbackup
```

- **Docker** キャッシュにイメージをロードして **Docker** イメージリポジトリにイメージをプッシュするには、次のコマンドを実行します。

- **NetBackup Kubernetes Operator** の **tar** ファイルをロードします。

```
<docker load -i <nameof the tar file> ./>
```

- 要件に従って、ロードされた **Docker** イメージにタグを付けます。

```
docker tag <imagename:tagof the loadedimage>
<repo-name/image-name:tag-name>
```

- **NetBackup Kubernetes Operator** の配備時に **Kubernetes** がイメージをフェッチできるリポジトリから、イメージをプッシュします。

```
docker push <repo-name/image-name:tag-name>
```

メモ: この例では、**Docker** が参照用に使用されています。同等の機能を提供する他の CLI ツールを使用できます。

7 テキストエディタで `netbackupkops-helm-chart/values.yaml` を編集します。

- マネージャセクションのイメージの値を、イメージ名とタグ `repo-name/image-name:tag-name` に置き換えます。
- レプリカの値を `0` に変更します。

メモ: NetBackup Kubernetes Operator を構成する手動の手順に従って、レプリカを 0 に設定します。

- 8 メタデータ永続ボリュームのサイズ調整が必要です。Kubernetes Operator のデフォルトの永続ボリュームサイズは 10Gi です。永続ボリュームサイズは構成可能です。プラグインを配備する前に、ストレージの値を 10Gi からより大きい値に変更できます。これにより、nbukops ポッドには、そのポッドでマウントされた PVC のサイズが設定されます。

メタデータの永続ボリュームサイズは values.yaml で指定できます。

helm-chart の deployment.yaml の永続ボリューム要求は次のようになります。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  labels:
    component: netbackup
    name: {{ .Release.Namespace }}-netbackupkops
    namespace: {{ .Release.Namespace }}
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
```

- 新規インストール時に Helm Chart を構成する際、netbackupkops-helm-chart の deployment.yaml で PVC ストレージのサイズを変更できます。これにより、初期の PVC サイズが作成されます。
- インストール後、PVC サイズの更新 (ダイナミックボリューム拡張) は一部のストレージベンダーによってサポートされます。詳しくは、<https://kubernetes.io/docs/concepts/storage/persistent-volumes> を参照してください。

メモ: 永続ボリュームのデフォルトサイズは、データを失うことなく、より大きい値にサイズ変更できます。ボリュームの拡張をサポートするストレージプロバイダを追加することをお勧めします。

- 9 NetBackup Kubernetes Operator サービスを配備するには、次のコマンドを実行します。

```
helm install <release name of the deployment>
./netbackupkops-helm-chart -n <namespace which runs NetBackup
operator service>

例: helm install veritas-netbackupkops ./netbackupkops-helm-chart
-n netbackup
```

- 必要に応じて配備のリリース名を変更できます。
- **NetBackup** オペレータサービスと **NetBackup** を実行する名前空間を指定するには、`-n` オプションが必要です。

10 配備の状態を確認するには、次のコマンドを実行します。

```
helm list -n <namespace which runs NetBackup operator service >

例:

helm list -n netbackup
```

11 リリース履歴を確認するには、次のコマンドを実行します。

```
helm history veritas-netbackupkops -n

<namespace which runs NetBackup operator service>。

例:

helm history veritas-netbackupkops -n netbackup
```

Kubernetes 作業負荷のポート要件

次の表は、Kubernetes Operator のポート要件を示しています。さまざまなホストの間にファイアウォールが存在する場合は、必要な通信ポートを開く必要があります。

表 2-1 NetBackup Kubernetes クラスタ環境で開く必要があるポート

ソース	ポート番号	宛先
NetBackup プライマリサーバー/メディアサーバー	Kubernetes API サーバーエンドポイントポート メモ: たとえば、 <code># kubectl cluster-info</code> のようにします。このコマンドは、API サーバーエンドポイントで使用されるポートを表示します。	Kubernetes クラスタ
Kubernetes クラスタ	Kubernetes API サーバーポート (NetBackup バージョン 9.1 が該当、バージョン 10.0 以降は該当せず)。	NetBackup プライマリサーバー

ソース	ポート番号	宛先
Kubernetes クラスタ	TCP ポート 1556 (アウトバウンド)	NetBackup プライマリサーバー/メディアサーバー
	TCP ポート 13724 (インバウンド) [耐性ネットワークを使用する場合にのみ必要です。]	メディアサーバー

NetBackup Kubernetes Operator のアップグレード

Helm コマンドを使用して NetBackup Kubernetes Operator の配備をアップグレードできます。

例:

```
helm upgrade veritas-netbackupkops ./netbackupkops-helm-chart -n netbackup
```

ローリングアップグレード

Kubernetes 保護は、標準的な NetBackup アップグレードアプローチに従い、ローリングアップグレードをサポートしています。

これは、すべてを同時にアップグレードするのではなく、Kubernetes Operator と NetBackup サーバーを段階的にアップグレードできることを意味します。アップグレード中に、Operator とデータムーバーは、メディアサーバーより 1 バージョン前、かつプライマリサーバーより最大 2 バージョン前のバージョンで実行できます。

異なるバージョンを使用する場合は、Operator のバージョンで利用可能な機能のみがサポートされます。

ローリングアップグレードのサポートは、NetBackup プライマリサーバーバージョン 11.0 以降から利用できます。

- 既存のポリシーはバックアップを作成し続けますが、Kubernetes Operator が更新されるまで手動でリストアする必要があります。

メモ: これは、NetBackup バージョン 9.1 から 10.x へのアップグレードに該当します。

NetBackup Kubernetes Operator をアップグレードするには

- 1 Cohesity テクニカルサポート Web サイト (<https://support.cohesity.com/s/>) から tar パッケージをダウンロードします。
- 2 ホームディレクトリにパッケージを抽出します。netbackupkops-helm-chart フォルダは、ホームディレクトリに存在する必要があります。
- 3 すべてのクラスタコンテキストを一覧表示するには、コマンド `kubectl config get-contexts` を実行します。
- 4 オペレータサービスを配備するクラスタに切り替えるには、次のコマンドを実行します。`kubectl config use-context <cluster-context-name>`
- 5 現在のディレクトリをホームディレクトリに変更するには、コマンド `cd ~` を実行します。
- 6 NetBackup は、OCI 標準に準拠したコンテナイメージリポジトリをサポートしています。オペレータとデータムーバイメージをプッシュする任意のツールを使用できます。プライベート Docker レジストリを使用している場合は、この手順の指示に従って、NetBackup 名前空間に **Secret** nb-docker-cred を作成します。それ以外の場合は、次の手順にスキップします。
 - Docker キャッシュにイメージをロードして Docker イメージリポジトリにイメージをプッシュするには、次のコマンドを実行します。
 - NetBackup Kubernetes Operator の tar ファイルをロードします。

```
<docker load -i <nameof the tar file> ./>
```
 - 要件に従って、ロードされた Docker イメージにタグを付けます。

```
docker tag <imagename:tagof the loadedimage>  
<repo-name/image-name:tag-name>
```
 - NetBackup Kubernetes Operator の配備時に Kubernetes がイメージをフェッチできるリポジトリに、イメージをプッシュします。

```
docker push <repo-name/image-name:tag-name>
```

メモ: この例では、Docker が参照用に使用されています。同等の機能を提供する他の CLI ツールを使用できます。

- 7 テキストエディタで netbackupkops-helm-chart/values.yaml を編集します。
 - **manager** セクションのイメージの値を、reponame/image-name:tag-name の形式でイメージ名とタグに置き換えます。

- `netbackup_config_pod` セクションのデータムーバーイメージを、データムーバーイメージの名前とタグに置き換えます。

8 NetBackup Kubernetes Operator をアップグレードするには、次のコマンドを実行します。

```
helm upgrade <plugin-name> <chart-path> -n <namespace>
```

例:

```
helm upgrade veritas-netbackupkops ./netbackupkops-helm-chart -n  
netbackup
```

メモ: NetBackup Kubernetes Operator をアップグレードすると、Helm の値がデフォルト値にリセットされます。アップグレード後に値が変更された場合は、古い `configmap` をバックアップし、パッチを再適用してください。

NetBackup Kubernetes Operator の削除

クラスタから NetBackup Kubernetes Operator の配備を削除できます。

```
helm uninstall <plugin-name> -n <Netbackup Kubernetes Operator  
Namespace>
```

メモ: プラグインをアンインストールすると、スナップショットベースのバックアップに関するメタデータが含まれる、NetBackup Kubernetes Operator PVC も削除されます。

NetBackup Kubernetes Operator を削除すると、スナップショットメタデータもホストするメタデータボリュームが失われます。スナップショットがすでに実行されている場合、メタデータがないと、スナップショットコピーからのリストア操作は失敗します。

NetBackup 9.1 では、古いスナップショットを手動で削除してから、関連付けられた Velero スナップショットを削除する必要があります。

NetBackup 10.0 では、NetBackup 9.1 を使用して作成された Velero 管理スナップショットを期限切れにできません。バックアップイメージが NetBackup で期限切れになると、カタログは自動的にクリアされます。ただし、Kubernetes サーバー 上のスナップショットは手動で削除する必要があります。

手動によるイメージの期限切れ操作について詳しくは、<https://support.cohesity.com/s/> を参照してください。

メモ: スナップショットを期限切れにしない、または永続ボリュームスナップショットを削除しない場合、Kubernetes Operator をアンインストールすると、孤立したボリュームスナップショットが Kubernetes クラスタに残ります。

NetBackup Kubernetes データムーバーの構成

NetBackup Kubernetes 作業負荷のデータムーバーを構成する必要があります。データムーバーイメージの正しいバージョンをダウンロードします。

スナップショットからのバックアップとバックアップからのリストアをサポートするには、NetBackup Kubernetes Operator 名前空間を構成する必要があります。(バックアップコピー)。ご使用のリリースバージョンに対応した正しいバージョンのデータムーバーイメージ veritasnetbackup-datamover-11.0.tar をダウンロードセンターからダウンロードしてください。<https://www.veritas.com/content/support> を参照してください。

データムーバーを構成する方法

- 1 イメージレジストリにデータムーバーイメージをプッシュするには、次のコマンドを実行します。

```
docker login -u <user name> <repo-name>
```

- 2 プロンプトでパスワードを入力します。ログイン済みの場合は、この手順をスキップします

- 3 `docker load -i <name of the datamover image file>` を実行します。

- 4 `docker tag <datamover image name:tag of the loaded datamover image> <repo-name/image-name:tag-name>` を実行します。

5 `docker push <repo-name/image-name:tag-name>`

メモ: この例では、**Docker** が参照用に使用されています。同等の機能を備えている CLI ツールを使用できます。

6 プライマリサーバー名の **ConfigMap** で、イメージ値が手順 4 でプッシュした `<repo-name/image-name:tag-name>` に設定されていることを確認します。

例:

```
apiVersion: v1
data:
  datamover.properties: image=<image-repo>/datamover:<datamover
tag>
  version: "1"
kind: ConfigMap
metadata:
  name: <Primary Server Name>
  namespace: <Netbackup Kubernetes Operator Namespace Name>
```

Configmap について詳しくは、『**NetBackup for Kubernetes 管理者ガイド**』の「**Kubernetes Operator** でサポートされる構成パラメータ」セクションを参照してください。

Kubernetes 用の NetBackup 保護の自動構成

前提条件

配備中に、ユーザーが **API** キーに対して、次のクレデンシャルを持っていることを確認します。自動配備の場合は、**API** キーを作成し、次のアクセス権が必要です。

1. **Kubernetes** クラスターで、次の内容を含む新しいシークレット `nb-config-deploy-secret.yaml` を作成します。

```
apiVersion: v1
kind: Secret
metadata:
  name: <kops-namespace>-nb-config-deploy-secret
  namespace: <kops-namespace>
type: Opaque
stringData:
  apikey: <Enter the value of API key from the earlier step>
```

2. シークレットを適用します。`kubectl apply -f nb-config-deploy-secret.yaml` コマンドを実行します。

API キーのクレデンシヤル

- 1 NetBackup Web UI を開きます。
- 2 左側で、[セキュリティ (Security)] をクリックし、[RBAC] を選択します。[追加 (Add)] をクリックします。
- 3 [カスタム役割 (Custom role)] を選択し、[次へ (Next)] をクリックします。
- 4 [役割の追加 (Add role)] ページで、[役割名 (Role name)] と [役割の説明 (Role description)] を指定します。[アクセス権 (Permissions)] で [割り当て (Assign)] をクリックします。
- 5 次のアクセス権が付与されていることを確認します。
 - [NetBackup 管理パネル (management panel)] > [NetBackup ホスト (hosts)]
 - 表示 (View)
 - 更新 (Update)
 - ホストマッピングを更新 (Update host mapping)
 - ホストマッピングの表示 (View host mapping)
 - [セキュリティ パネル (Security panel)] > [証明書管理 (Certificate management)] > [NetBackup セキュリティトークン (security tokens)]
 - [クレデンシヤル (Credentials)] タブ
 - 表示 (View)
 - 作成 (Create)
 - 更新 (Update)
 - 削除 (Delete)
 - クレデンシヤルの割り当て (Assign credentials)
 - [資産 (Assets)] タブ > [Kubernetes 資産 (Kubernetes assets)] > [Kubernetes クラスタと名前空間 (Kubernetes Cluster and Namespaces)]
 - 作成 (Create)
 - 更新 (Update)
 - 削除 (Delete)
 - 表示 (View)

Kubernetes 作業負荷で NetBackup を構成する前に、ポート 443、1556、および 13724 へのアクセス権を付与して NetBackup Server を実行する必要があります。

NetBackup の Kubernetes Operator とデータムーバーのイメージは、Kubernetes クラスタからアクセス可能なコンテナレジストリにアップロードする必要があります。

自動配備のために使用するシークレットを作成する必要があります。

API キーを作成するには

- 1 NetBackup Web UI を開きます。
- 2 左側で[セキュリティ (Security)]、[アクセスキー (Access keys)]の順に選択します。
- 3 [API キー (API keys)]タブをクリックします。
- 4 [追加 (Add)]をクリックします。

インストール前

- 1 `netbackupkops-helm-chart/values.yaml` の次のフィールドを編集します。
 - `containers.manager.image`: NetBackup Kubernetes コントローライメージをプルするためのコンテナレジストリ URL
 - `imagePullSecrets: name`: コンテナレジストリがイメージをプルするために認証を必要とする場合の、イメージプルシークレットの名前。
 - `nbprimaryserver`: NetBackup プライマリサーバーの構成された名前。
 - `nbsha256fingerprint`: NetBackup Web UI から sha256 の指紋をフェッチします。左側で[セキュリティ (Security)]、[証明書 (Certificates)]の順に選択します。[認証局 (Certificate Authority)]をクリックします。
 - `k8sCluster`: Kubernetes クラスタ API サーバーの FQDN。
 - `k8sPort`: Kubernetes API サーバーが一覧表示されるポート。
 - `datamoverProperties` (省略可能): `datamover` ポッドの `bp.conf` ファイルに指定する必要がある構成設定を含めます。

メモ: 詳しくは、p.38 の「[短縮名の付いた NetBackup サーバーのトラブルシューティング](#)」を参照してください。

この情報は、Kubernetes クラスタの UI コンソールで利用可能です。

- 2 存在しない場合は、次のコマンドを実行して Kubernetes クラスタと Kubernetes ポートを取得します。

`kubectl cluster-info` Kubernetes コントロールプレーンは、
<https://<Kubernetes FQDN>:6443> で実行されます

- `datamoverimage`: データムーバーイメージをプルするコンテナレジストリ URL。

- スナップショット操作とスナップショットからのバックアップ操作には、ストレージパラメータが必要です。ブロックまたはファイルシステムのストレージパラメータの少なくとも 1 つは必須です。

3 ストレージクラスを取得するには、次のコマンドを実行します。

```
# kubectl get storageclasses
```

- **storageclassblock**: ブロックボリュームのプロビジョニングに使用されるストレージクラス。
- **storageclassfilesystem**: ファイルシステムボリュームのプロビジョニングに使用されるストレージクラス。

4 ボリュームスナップショットクラスを取得するには、次のコマンドを実行します。

```
# kubectl get volumesnapshotclasses
```

- **volumesnapshotclassblock**: ブロックボリュームスナップショットを作成するためのボリュームスナップショットクラス。
- **volumesnapshotclassfilesystem**: ファイルシステムボリュームスナップショットを作成するためのボリュームスナップショットクラス。

5 ストレージクラスとスナップショットクラス間のマッピングは、**storageMap** を介して管理されます。新しいストレージオプションがクラスタに追加された場合、インストール後に **backup-operator-configuration** の **configmap** で更新することもできます。

- **storageMap** はキーの辞書であり、値フィールドはキーがストレージクラスで、値が (**snapshotClass**、**storageClassForBackupDataMovement**、**storageClassForRestoreFromBackup**) で構成されるタプルです。このフィールドは、ストレージクラスとスナップショットクラス間のマッピングを指定する場合に必須です。
- **snapshotclass** はストレージクラスと同じプロビジョナを使用して作成する必要があり、ストレージクラスのスナップショットを作成できる必要があります。すべてのストレージクラスに **snapshotclass** のエントリが必要です。
- **storageClassForBackupDataMovement** は、データムーバー用の一時的な PVC を作成するために使用されます。このストレージクラスを使用して作成する場合は、元のストレージクラスのスナップショットを使用して作成された元のストレージクラス PVC と互換性がある必要があります。データムーバーはこの PVC からデータを読み込み、NetBackup メディアサーバーに送信します。**storageClassForRestoreFromBackup** はメディアサーバーのバックアップからリストアするために使用されます。元のストレージクラスと互換性があり、同じプロビジョナを使用している必要があります。
- 1 つのスナップショットクラスを使用して、互換性のある複数のストレージクラスのスナップショットを作成できます。
- テンプレート

```
storageMap:
  <key - storage class name>:
    snapshotClass: [mandatory field to specify
volumesnapshotclass for creating snapshot of given key storage
class]
    storageClassForBackupDataMovement: <optional, storage class
used to transfer pvc backup data from k8s cluster to
NetBackup media server>
    storageClassForRestoreFromBackup: <optional, storage class
used to restore pvc from NetBackup media server
to k8s cluster>
```

Note: storageClassForBackupDataMovement and storageClassForRestoreFromBackup are optional and must be compatible with key storage class if they are configured different from key storage class. If no value is specified for these fields original storage class would be used. These values can be changed later in backup-operator-configuration configmap

Example for openshift storage classes. cephfs storage class should have corresponding snapclass as cephfs as follows

```
storageMap:
  ocs-storagecluster-cephfs:
    storageClassForBackupDataMovement:
ocs-storagecluster-cephfs
    storageClassForRestoreFromBackup: ocs-storagecluster-cephfs

    snapshotClass: ocs-storagecluster-cephfspplugin-snapclass
  ocs-storagecluster-ceph-rbd:
    snapshotClass: ocs-storagecluster-rbdplugin-snapclass
```

インストール

helm をインストールするには、次のコマンドを実行します。

```
# helm install veritas-netbackupkops <path to
netbackupkops-helm-chart> -n <kops namespace>
```

デバッグ

Kubernetes Operator 名前空間から config-deploy ポッドを取得するには、次のコマンドを実行します。

```
# kubectl get pod -n <kops namespace> | grep "config-deploy"
```

ログ

ポッド <namespace>-netbackup-config-deploy からログを確認するには、次のコマンドを実行します。

```
# kubectl logs <pod-name> -n <kops namespace>
```

ログレベル

構成ポッドのログレベルを設定します。値には DEBUG、INFO、または ERROR を設定できます。デフォルト値は INFO に設定されます。

メモ: 詳しくは、『*NetBackup Kubernetes クイックスタートガイド*』を参照してください。

Kubernetes 作業負荷のカスタマイズ

構成値を取得するには、以下のコマンドを実行します。

```
kubectl get configmaps <namespace>-backup-operator-configuration -n  
<namespace> -o yaml > {local.file}。
```

構成を編集するには、以下のコマンドを実行します。

```
kubectl edit cm<backup-operator-configmap> -n <kops-namespace>。
```

資産の自動検出を無効にするには、VirtualMachine=false を設定します。

メモ: NetBackup では、[リソース (Resources)] セクションの VirtualMachine パラメータのみを編集できます。

メモ: 構成値を取得するには、コマンド `kubectl get configmaps <namespace>-backup-operator-configuration -n <namespace> -o yaml > {local.file}` を実行します。

表 2-2 Kubernetes Operator でサポートされる、
<namespace>-backup-operator-configuration の構成パラメータ

構成	説明	デフォルト 値	指定可能 な値
daemonsets	値が false に設定されている場合、このリソースは検出され、バックアップされます。ただし、 NetBackup Web UI で Kubernetes 名前空間をクリックした場合は、[リソース (Resource)]セクションにリソースは表示されません。	true	true、false
deployments	値が false に設定されている場合、このリソースは検出され、バックアップされます。ただし、 NetBackup Web UI で Kubernetes 名前空間をクリックした場合は、[リソース (Resource)]セクションにリソースは表示されません。	true	true、false
pods	値が false に設定されている場合、このリソースは検出され、バックアップされます。ただし、 NetBackup Web UI で Kubernetes 名前空間をクリックした場合は、[リソース (Resource)]セクションにリソースは表示されません。	true	true、false

構成	説明	デフォルト値	指定可能な値
replicasets	値が false に設定されている場合、このリソースは検出され、バックアップされます。ただし、 NetBackup Web UI で Kubernetes 名前空間をクリックした場合は、[リソース (Resource)] セクションにリソースは表示されません。	true	true、false
secrets	値が false に設定されている場合、このリソースは検出され、バックアップされます。ただし、 NetBackup Web UI で Kubernetes 名前空間をクリックした場合は、[リソース (Resource)] セクションにリソースは表示されません。	true	true、false
services	値が false に設定されている場合、このリソースは検出され、バックアップされます。ただし、 NetBackup Web UI で Kubernetes 名前空間をクリックした場合は、[リソース (Resource)] セクションにリソースは表示されません。	true	true、false

構成	説明	デフォルト値	指定可能な値
namespace	Kubernetes Operator は名前空間に配備されます。 値が false に設定されている場合、このリソースは検出され、バックアップされます。ただし、NetBackup Web UI で Kubernetes 名前空間をクリックした場合は、[リソース (Resource)] セクションにリソースは表示されません。	名前空間に指定された任意の名前。	NetBackup の名前空間。
cleanStaleCRDurationMinutes	古い CR をクリーニングするために CR ジョブが呼び出されてからの期間。古いカスタムリソースのクリーンアップジョブがトリガされてからの間隔。 長時間実行されるスナップショットからのバックアップとバックアップジョブからのリストアでは、cleanStaleCRDurationMinutes 値を増やす必要があります。	1,440 分	分単位の任意の値
ttlCRDurationMinutes	TTL CR の期間	30,240 分	30,240 分
fipsMode	NetBackup Kubernetes Operator とデータムーバーで FIPS_MODE を有効にする構成。	DISABLE	ENABLE、DISABLE
livenessProbeInitialDelay	精査の初期遅延期間。	60 秒	60 分

構成	説明	デフォルト値	指定可能な値
livenessProbeTimeoutInSeconds	読み込まれたマシンでは、Liveness Probe の実行に 1 秒より長くかかる場合があります。ユーザーはこの値を大きくして、Liveness Probe のタイムアウトエラーを原因とする障害を回避できます。	1 秒	1 秒から 5 秒。
livenessProbePeriodInSeconds	精査の期間。	180 秒	秒単位の任意の値
checkNbcertdaemonStatusDurationMinutes	NB 証明書デーモンの状態の期間。	1,440 分	1,440 分
collectDataMoverLogs	datamover ログの収集ではメモリ使用率が高くなるため、ポッドのデバッグ、トラブルシューティング、再起動を行う場合のみログを有効にすることをお勧めします。 datamover のログを有効にする前に、NetBackup Kubernetes ポッド用のメモリ上限を 2 GB 以上に増やしてください。デバッグまたはトラブルシューティングの完了後は、以前の値またはデフォルト値にリセットできます。 メモ: 失敗したジョブの場合にのみ、datamover ログを収集するための詳細なサポートが提供されます。これにより、より詳細なレベルとして All/FailedOnly/Off が提供されます。	Failed	All、Failed、None

構成	説明	デフォルト値	指定可能な値
maxRetentionDataMoverLogsInHours	datamover ログの最大保持期間。	24 時間	72 時間
maxRetentionDataMoverInHours	指定した時間より古い datamover リソースがすべて削除されます。	24 時間	時間単位の任意の値
cleanStaleCertFilesDurationMinutes	古い証明書ファイルのクリーンアップジョブがトリガされてからの間隔。	60 分	1440 分
maxRetentionInDiscoveryCacheHours	検出キャッシュを保持する時間間隔を決定する時間。	24 時間	48 時間
pollingTimeoutInMinutes	期限切れになり失敗するまで再試行し続けるタイムアウトです。	15 分	分単位の任意の値
pollingFrequencyInSecs	ポーリング間隔。	5 秒	秒単位の任意の値
nbcertPrerequisiteDirectoryAndFiles	NBCA の前提条件。	証明書名	証明書名

スナップショットからのバックアップ操作とバックアップからのリストア操作の前提条件

1. `storageMap` に追加されたストレージクラスで、ボリュームバインドモードが[即時 (Immediate)]に設定されていることを確認します。PVC ボリュームバインドモードが `WaitForFirstConsumer` である場合、PVC からのスナップショットの作成に影響します。この場合、バックアップジョブが失敗する可能性があります。

例: 次のコマンドを実行します: `# kubectl get sc`

2. スナップショットからのバックアップ操作とバックアップコピーからのリストア操作を実行する各プライマリサーバーは、プライマリサーバーの名前を使用して個別の **ConfigMap** を作成する必要があります。

次の `configmap.yaml` の例では、

- `backupserver.sample.domain.com` と `mediaserver.sample.domain.com` は、NetBackup プライマリサーバーとメディアサーバーのホスト名です。
- `IP: 10.20.12.13` と `IP: 10.21.12.13` は、NetBackup プライマリサーバーとメディアサーバーの IP アドレスです。

```
apiVersion: v1
data:
  datamover.hostaliases: |
    10.20.12.13=backupserver.sample.domain.com
    10.21.12.13=mediaserver.sample.domain.com
  datamover.properties: |
    image=reg.domain.com/datamover/image:latest
    version: "1"
kind: ConfigMap
metadata:
  name: backupserver.sample.domain.com
  namespace: kops-ns
```

- configmap.yaml ファイルの詳細をコピーします。
- テキストエディタを開き、yaml ファイルの詳細を貼り付けます。
- ファイルに yaml というファイル拡張子を付けて、Kubernetes クラスタにアクセスできるホームディレクトリに保存します。

3. 正しいデータムーバイメージで datamover.properties: image=reg.domain.com/datamover/image:latest を指定します。
4. プライマリサーバーとプライマリサーバーに接続されているメディアサーバーで短縮名が使用されていて、データムーバからのホストの解決が失敗している場合は、datamover.hostaliases を指定します。プライマリサーバーとメディアサーバーのすべてのホスト名と IP のマッピングを指定します。
5. プライベート Docker レジストリを使用するには、「NetBackup Kubernetes Operator でのサービスパッケージの配備」セクションのポイント 6 の説明に従って、シークレットを作成します。

シークレットが作成されたら、configmap.yaml ファイルの作成中に次の属性を追加します。

```
datamover.properties: |
  image=repo.azurecr.io/netbackup/datamover:10.0.0049
  imagePullSecret=secret_name
```

6. configmap.yaml ファイルを作成します。kubectl create -f configmap.yaml コマンドを実行します。
7. Kubernetes Operator がプライマリサーバーを短縮名で解決できない場合は、次のガイドラインを参照してください。
 - 証明書のフェッチ中に **[EXIT STATUS 8500: Web サービスとの接続が確立されませんでした (EXIT STATUS 8500: Connection with the web service was not established)]** というメッセージが表示された場合は、ホスト名の解決の状態を nbcert ログで確認します。

- ホスト名解決が失敗した場合は、hostAliases で values.yaml ファイルを更新します。
- 次の hostAliases の例では、
 - backupserver.sample.domain.com と mediaserver.sample.domain.com は、NetBackup プライマリサーバーとメディアサーバーのホスト名です。
 - IP: 10.20.12.13 と IP: 10.21.12.13 は、NetBackup プライマリサーバーとメディアサーバーの IP アドレスです。

```
hostAliases:
- hostnames:
  - backupserver.sample.domain.com
  ip: 10.20.12.13
- hostnames:
  - mediaserver.sample.domain.com
  ip: 10.21.12.13
```

hostAliases の例の詳細をコピーしてテキストエディタに貼り付け、配備で hostAliases に追加します。

メモ: hostAliases セクションは、デフォルトのファイル
./netbackupkops-helm-chart/values.yaml に追加する必要があります。

hostAliases の例:

```
2104 hostAliases;
- ip:10.15.206.7
hostnames:
- lab02-linsvr-01.demo.sample.domain.com
- lab02-linsvr-01
- ip:10.15.206.8
hostnames:
- lab02-linsvr-02.demo.sample.domain.com
- lab02-linsvr-02
imagePullSecrets:
- name:  {{ .values.netbackupKops.imagePullSecrets.name }}
```

8. nbcertcmdtool の TLS 関連の構成を更新するには、deployment.yaml ファイル内にある {{ .Release.Namespace }}-certconfigscript という名前の configmap を必要な設定で更新します。

例:

```
To set TLS_MAX_VERSION,
apiVersion: v1
data:
  nbcert.sh: |
    #!/bin/sh
    mkdir -p /usr/opensv/kops
    mkdir -p /usr/opensv/fingerprint-dir
    mkdir -p /usr/opensv/tmp
    mkdir -p /usr/opensv/netbackup/logs/nbcert
    mkdir -p /usr/opensv/netbackup/logs/nbcert/nobody
    mkdir -p /usr/opensv/var/global
    mkdir -p /usr/opensv/var/vxss
    cp -r /nbcertcmdtool /usr/opensv/nbcertcmdtool
    touch /usr/opensv/var/global/nbcl.conf
    touch /usr/opensv/netbackup/bp.conf
    chown -R nobody:nobody /usr/opensv
    echo "CLIENT_KEEP_LOG_DAYS = 90" >>
  /usr/opensv/netbackup/bp.conf
    echo "SERVICE_USER=nobody" >> /usr/opensv/netbackup/bp.conf
    echo "MACHINE_NBU_TYPE = KUBERNETES_CLUSTER" >>
  /usr/opensv/netbackup/bp.conf
    echo "TLS_MAX_VERSION = TLSv1.3" >>
  /usr/opensv/netbackup/bp.conf
kind: ConfigMap
metadata:
  labels:
    component: netbackup
  name: {{ .Release.Namespace }}-certconfigscript
  namespace: {{ .Release.Namespace }}
```

9. 指紋と認証トークンを使用して **Secret** を作成します。

シークレットと **backupservercert** の作成について詳しくは、『**NetBackup for Kubernetes 管理者ガイド**』の「**NetBackup Kubernetes Operator** での証明書の配備」セクションを参照してください。

10. 証明書をフェッチするための **backupservercert** 要求を作成します。

詳しくは、『**NetBackup for Kubernetes 管理者ガイド**』の「**NetBackup Kubernetes Operator** での証明書の配備」を参照してください。

詳しくは『**NetBackup セキュリティおよび暗号化ガイド**』を参照してください。

メモ: この手順は、スナップショットからのバックアップとバックアップコピーからのリストアを正常に実行するために必須です。

Kubernetes でサポートされる DTE クライアント設定

DTE_CLIENT_MODE オプションは、バックアップサーバー固有の **configmap** によって **datamover** に設定される移動中のデータの暗号化 (DTE) モードを指定します。バックアップイメージの移動中のデータの暗号化は、グローバル DTE モードとクライアント DTE モードに基づいて実行されます。

バックアップサーバー固有の **configmap** を更新し、DTE_CLIENT_MODE キーを追加します。このキーは次の値を取ることができます。

- AUTOMATIC
- ON
- OFF

DTE_CLIENT_MODE について詳しくは、『NetBackup 管理者ガイド Vol. 1』の「クライアント用 DTE_CLIENT_MODE」のセクションを参照してください。

次に、DTE_CLIENT_MODE 設定を追加した **configmap** を示します。

```
apiVersion: v1
data:
  datamover.hostaliases: |
    10.20.12.13=backupserver.sample.domain.com
    10.21.12.13=mediaserver.sample.domain.com
  datamover.properties: |
    image=reg.domain.com/datamover/image:latest
    DTE_CLIENT_MODE=ON
  version: "1"
kind: ConfigMap
metadata:
  name: backupserver.sample.domain.com
  namespace: kops-ns
```

datamover プロパティのカスタマイズ

キーと値のペアをバックアップサーバー固有の **configmap** に渡すことによって、**datamover** プロパティをカスタマイズできます。

表 2-3 datamover のプロパティ

キー名	指定可能な値
VXMS_VERBOSE	範囲: [0,99]
VERBOSE	範囲: [0,5]
DTE_CLIENT_MODE	■ AUTOMATIC ■ ON ■ OFF
USE_CTIME_FOR_INCREMENTALS	YES / NO
USE_CTIME_FOR_DIRECTORY_INCRS	YES / NO
DO_NOT_RESET_FILE_ACCESS_TIME	YES / NO

メモ: NetBackup クライアントでサポートされているその他の構成設定は、**datamover configmap** の **datamover.properties** キーの下に追加することで、データムーバー用に設定できます。これらの構成は、データムーバー内の `bp.conf` ファイルに追加されません。

configmap を更新するには、次のようにキーと値のペアを追加します。

```
apiVersion: v1
data:
  datamover.properties: |
    image=reg.domain.com/datamover/image:latest
    VERBOSE=5
    DTE_CLIENT_MODE=OFF
    VXMS_VERBOSE=5
  version: "1"
kind: ConfigMap
metadata:
  name: backupserver.sample.domain.com
  namespace: kops-ns
```

NetBackup 10.5 以降では、TLS 1.3 プロトコルがサポートされます。NetBackup 10.5 以降のバージョンでは、次のように安全な通信ワークフローでデフォルトで TLS 1.3 プロトコルを使用します。

- セキュアプロキシ
- DTE (移動中のデータの暗号化)
- 外部 KMS サーバー

- AD/LDAP サーバー
- MSDP
- cURL を使用した HTTPS 通信

TLS 関連のプロパティを構成するには、この構成マップで必要な設定を更新できます。
TLS 設定について詳しくは、この[記事](#)を参照してください。

短縮名の付いた NetBackup サーバーのトラブルシューティング

- 1 NetBackup Kubernetes Operator が短縮名を基にバックアップサーバーまたはメディアサーバーを解決できない場合は、次の手順を実行します。
 - 証明書のフェッチ中に **[EXIT STATUS 8500: Web サービスとの接続が確立されませんでした (EXIT STATUS 8500: Connection with the web service was not established)]** というメッセージが表示された場合。次に、ホスト名解決が成功したかどうかを `nbcert` ログから確認します。失敗した場合は、次の手順を実行します。
 - Kubernetes Operator の `deployment.yaml` を更新し、その配備に `hostAliases` を追加します。
 - 次の `hostAliases` の例では、
 - `backupserver.sample.domain.com` と `mediaserver.sample.domain.com` は、NetBackup プライマリサーバーとメディアサーバーのホスト名です。
 - IP: `10.20.12.13` と IP: `10.21.12.13` は、NetBackup プライマリサーバーとメディアサーバーの IP アドレスです。

```
hostAliases:
- hostnames:
  - backupserver.sample.domain.com
  ip: 10.20.12.13
- hostnames:
  - mediaserver.sample.domain.com
  ip: 10.21.12.13
```

`hostAliases` の例の詳細をコピーしてテキストエディタに貼り付け、配備で `hostAliases` に追加します。

- 2 データムーバーがバックアップサーバーまたはメディアサーバーの短縮名を解決できない場合。この問題を解決するには、次の手順を実行します。
 - バックアップサーバー名を使用して `configmap` を更新します。

- `datamover.hostaliases` フィールドを追加し、ホスト名に IP アドレスをマップします。
- 次の `configmap.yaml` の例では、
 - `backupserver.sample.domain.com` と `mediaserver.sample.domain.com` は、NetBackup プライマリサーバーとメディアサーバーのホスト名です。
 - IP: `10.20.12.13` と IP: `10.21.12.13` は、NetBackup プライマリサーバーとメディアサーバーの IP アドレスです。

```
apiVersion: v1

data:
  datamover.hostaliases: |
    10.20.12.13=backupserver.sample.domain.com
    10.21.12.13=mediaserver.sample.domain.com
  datamover.properties: |
    image=reg.domain.com/datamover/image:latest
    version: "1"
kind: configmap
metadata:
  name: backupserver.sample.domain.com
  namespace: kops-ns
```

- `configmap.yaml` ファイルの詳細をコピーします。
- テキストエディタを開き、`yaml` ファイルの詳細を貼り付けます。
- その後、それに `yaml` ファイル拡張子を付けて、Kubernetes クラスタにアクセスできるホームディレクトリに保存します。
- `configmap.yaml` ファイルを作成するには、コマンド `kubectl create -f configmap.yaml` を実行します。
- すでに作成されている `configmap.yaml` を更新する場合は、コマンドを実行して `configmap` を更新します。 `kubectl apply -f configmap.yaml`

datamover ポッドのスケジュールメカニズムのサポート

NetBackup では、ユーザーは `datamover` ポッドに注釈を付けて、バックアップサーバー固有の `ConfigMap` を使用して `datamover` ポッドにネットワーク接続定義を適用することで、追加のネットワークの使用を有効にできます。

新規インストール中に NAD (ネットワーク接続定義) で datamover ポッドに注釈を付ける方法

NetBackup Kubernetes Operator Helm のインストール中に、ユーザーは `netbackupkops-helm-chart/values.yaml` ファイルで注釈を指定できます。これは、NetBackup Kubernetes Operator の配備と datamover ポッドに適用されます。これは省略可能なパラメータです。

インストール後に NAD (ネットワーク接続定義) で datamover ポッドに注釈を付ける方法

ユーザーは、バックアップサーバー固有の ConfigMap を編集して、注釈を付けたり既存の注釈を修正したりできます。

例

```
# kubectl get cm -n <kops-namespace>

# kubectl edit cm/<backup-server-name> -n <kops-namespace>

datamover.annotations: |

k8s.v1.cni.cncf.io/networks: whereabouts-ipvlan-conf-1
```

制限事項

datamover ポッドで専用ネットワークが使用されている場合、ポッド内のホストエイリアスは複数の専用ネットワークに対しては機能しません。最初のインターフェースが停止しても、接続は 2 つ目の専用バックアップネットワークにフォールバックされません。

datamover に 2 つの専用ネットワークインターフェースを個別に接続する代わりに、両方のインターフェースを使用してブリッジを作成し、このブリッジの上に multus を構成することをお勧めします。

バックアップサーバーの ConfigMap で次のフィールドを指定して、ノード上の datamover ポッドをスケジュールします。

1. **nodeSelector:** nodeSelector は、ポッドを特定のラベルを持つノードに制約する簡単な方法です。

例:

```
apiVersion: v1

kind: ConfigMap

metadata:

  name: backupserver.sample.domain.com
```

```
namespace: netbackup

data:

  datamover.hostaliases: |

    10.20.12.13=backupserver.sample.domain.com

    10.21.12.13=mediaserver.sample.domain.com

  datamover.properties: |

    image=reg.domain.com/datamover/image:latest

  datamover.nodeSelector: |

    kubernetes.io/hostname: test1-194jm-worker-k49vj

    topology.rook.io/rack: rack1

version: "1"
```

2. **nodeName: nodeName** は、親和性または **nodeSelector** よりも直接的にノードを選択する形式です。ポッドがバックアップ用にスケジュールされているノードを指定でき、デフォルトのスケジュールメカニズムを上書きできます。

例:

```
apiVersion: v1

kind: ConfigMap

metadata:

  name: backupserver.sample.domain.com

  namespace: netbackup

data:

  datamover.hostaliases: |

    10.20.12.13=backupserver.sample.domain.com
```

```
10.21.12.13=mediaserver.sample.domain.com

datamover.properties: |

    image=reg.domain.com/datamover/image:latest

datamover.nodeName : test1-194jm-worker-hbblk

version: "1"
```

3. **Taint と Toleration:** Toleration を使用すると、類似する Taint を使用してポッドをスケジュールすることができます。Taint と Toleration を連携して使用することで、ポッドを適切なノードに確実にスケジュールできます。1 つ以上の Taint がノードに適用される場合を考えます。そのノードは、Taint を容認 (tolerate) しないポッドを受け入れることはできません。

例:

```
apiVersion: v1

kind: ConfigMap

metadata:

    name: backupserver.sample.domain.com

    namespace: netbackup

data:

    datamover.hostaliases: |

        10.20.12.13=backupserver.sample.domain.com

        10.21.12.13=mediaserver.sample.domain.com

    datamover.properties: |

        image=reg.domain.com/datamover/image:latest

    datamover.tolerations: |

        - key: "dedicated"
```

```
operator: "Equal"

value: "experimental"

effect: "NoSchedule"

version: "1"
```

4. 親和性と反親和性: ノード親和性は **nodeSelector** フィールドのように機能しますが、より表現力が高く、柔軟なルールを指定できます。ポッド間の親和性/反親和性を使用すると、他のポッドのラベルに対してポッドを制約できます。

例:

■ ノード親和性:

```
apiVersion: v1

kind: ConfigMap

metadata:

  name: backupserver.sample.domain.com

  namespace: netbackup

data:

  datamover.hostaliases: |

    10.20.12.13=backupserver.sample.domain.com

    10.21.12.13=mediaserver.sample.domain.com

  datamover.properties: |

    image=reg.domain.com/datamover/image:latest

  datamover.affinity: |

    nodeAffinity:

      requiredDuringSchedulingIgnoredDuringExecution:
```

```
nodeSelectorTerms:

- matchExpressions:

  - key: kubernetes.io/hostname

    operator: In

    values:

      - test1-194jm-worker-hbblk

preferredDuringSchedulingIgnoredDuringExecution:

- weight: 1

  preference:

    matchExpressions:

      - key: beta.kubernetes.io/arch

        operator: In

        values:

          - amd64

version: "1"
```

■ ポッド親和性

```
apiVersion: v1

kind: ConfigMap

metadata:

  name: backupserver.sample.domain.com

  namespace: netbackup
```

```
data:

  datamover.hostaliases: |

    10.20.12.13=backupserver.sample.domain.com

    10.21.12.13=mediaserver.sample.domain.com

  datamover.properties: |

    image=reg.domain.com/datamover/image:latest

  datamover.affinity: |

    podAffinity:

      requiredDuringSchedulingIgnoredDuringExecution:

        - labelSelector:

            matchExpressions:

              - key: component

                operator: In

                values:

                  - netbackup

            topologyKey: kubernetes.io/hostname

    version: "1"
```

5. **topologySpreadConstraints:** トポロジー分散制約は、地域、ゾーン、ノード、その他のユーザー定義トポロジードメインなどの障害ドメイン間でクラスタ全体に分散するポッドの動作を制御するために使用されます。

例:

```
apiVersion: v1

kind: ConfigMap
```

```
metadata:

  name: backupserver.sample.domain.com

  namespace: netbackup

data:

  datamover.hostaliases: |

    10.20.12.13=backupserver.sample.domain.com

    10.21.12.13=mediaserver.sample.domain.com

  datamover.properties: |

    image=reg.domain.com/datamover/image:latest

  datamover.topologySpreadConstraints : |

    - maxSkew: 1

    topologyKey: kubernetes.io/hostname

    whenUnsatisfiable: DoNotSchedule

version: "1"
```

- ラベル: ラベルは、ポッドなどのオブジェクトに関連付けられているキーと値のペアです。ラベルは、オブジェクトの重要でユーザーに関連する属性を識別することを意図しています。ラベルを使用することで、オブジェクトのサブセットを整理して選択できます。ラベルは作成時にオブジェクトにアタッチでき、その後いつでも追加および変更できます。

例:

```
apiVersion: v1

kind: ConfigMap

metadata:

  name: backupserver.sample.domain.com

  namespace: netbackup
```

```
data:

  datamover.hostaliases: |

    10.20.12.13=backupserver.sample.domain.com

    10.21.12.13=mediaserver.sample.domain.com

  datamover.properties: |

    image=reg.domain.com/datamover/image:latest

  datamover.labels: |

    env: test

    pod: datamover

  version: "1"
```

- 注釈: ユーザーはラベルまたは注釈を使用して、**Kubernetes** オブジェクトにメタデータをアタッチできます。注釈を使用してオブジェクトを識別および選択することはできません。

例:

```
apiVersion: v1

kind: ConfigMap

metadata:

  name: backupserver.sample.domain.com

  namespace: netbackup

data:

  datamover.hostaliases: |

    10.20.12.13=backupserver.sample.domain.com

    10.21.12.13=mediaserver.sample.domain.com
```

```
datamover.properties: |

    image=reg.domain.com/datamover/image:latest

datamover.annotations: |

    buildinfo: |-

        [{

            "name": "test",

            "build": "1"

        }]

    imageregistry: "https://reg.domain.com/"

version: "1"
```

アクセラレータストレージクラスの検証

NetBackup は、アクセラレータが有効なバックアップをサポートしています。これは、インストールまたはアップグレード中に、`values.yaml` の **`acceleratorTracklogPvcStorageClass`** キーを適切なストレージクラスに設定することで有効にできます。

ストレージクラスは、ファイルモード PVC の作成を許可する必要があります。

例: **`acceleratorTracklogPvcStorageClass: ocs-storagecluster-ceph-rbd`**

インストールおよびアップグレード中に、NetBackup Kubernetes Operator がファイルモード PVC とポッドを作成し、指定したストレージクラスが有効かどうかを確認します。

- ストレージクラスのボリュームバインドモードが **Immediate** の場合は PVC のみが作成され、PVC がバインド状態の場合はインストールが成功します。
- ストレージクラスのボリュームバインドモードが **WaitForFirstConsumer** の場合は、PVC を使用して **datamover** ポッドが作成されます。
- PVC がバインド状態でポッドが実行中状態の場合、NetBackup Kubernetes Operator のインストールは成功です。

NetBackup Kubernetes Operator での証明書の配備

この章では以下の項目について説明しています。

- [Kubernetes Operator](#) での証明書の配備
- [ホスト ID ベースの証明書操作の実行](#)
- [ECA 証明書操作の実行](#)
- [証明書の種類の識別](#)

Kubernetes Operator での証明書の配備

`datamover` と NetBackup メディアサーバー間で安全に通信するために証明書を配備する必要があります。

メモ: スナップショットからのバックアップ操作とバックアップからのリストア操作を実行する前に、証明書を配備する必要があります。

`BackupServerCert` を作成する前に、クラスタの追加と検出が正常に行われる必要があります。これは、状態を成功として設定するために、NetBackup がいくつかの `clusterInfo` を渡すことに依存しているためです。

`datamover` 通信でサポートされる証明書

`datamover` は、NetBackup 環境内のデータ移動を容易にし、TLS (Transport Layer Security) を介してメディアサーバーと通信します。詳しくは、『NetBackup™ セキュリティおよび暗号化ガイド』の「NetBackup での安全な通信について」セクションを参照してください。

ださい。**datamover** が通信するためには、ホスト ID ベースの証明書、または **NetBackup** プライマリサーバーによって発行され、**ECA** が署名した証明書が必要です。**NBCA** (**NetBackup** 認証局) または **ECA** (外部認証局) モードで証明書配備操作を実行できるように、新しいカスタムリソース定義 **BackupServerCert** が導入されました。

カスタムリソースの指定は次のようになります。

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupservercert-sample-nbca
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: primary.server.sample.com
  certificateOperation: Create | Update | Remove
  certificateType: NBCA | ECA
  nbcaAttributes:
    nbcaCreateOptions:
      secretName: "Secret name consists of token and fingerprint"
    nbcaUpdateOptions:
      secretName: "Secret name consists of token and fingerprint"
      force: true | false
    nbcaRemoveOptions:
      hostID: "hostID of the nbca certificate. You can view on Netbackup UI"
  ecaAttributes:
    ecaCreateOptions:
      ecaSecretName: "Secret name consists of cert, key, passphrase, cacert"
      copyCertsFromSecret: true | false
      isKeyEncrypted: true | false
    ecaUpdateOptions:
      ecaCrlCheck: DISABLE | LEAF | CHAIN
      ecaCrlRefreshHours: [0,4380]
```

ホスト ID ベースの証明書操作の実行

プライマリサーバーが **NBCA** モードで構成されていることを確認します。**NBCA** モードがオンかどうかを確認するには、コマンド `/usr/opensv/netbackup/bin/nbcertcmd -getSecConfig -caUsage` を実行します。

出力は次のようになります。

NBCA: ON
ECA: OFF

ホスト ID ベースの証明書の指定は次のようになります。

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupservercert-sample
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: primaryserver.sample.domain.com
  certificateOperation: Create | Update | Remove
  certificateType: NBCA
  nbcaAttributes:
    nbcaCreateOptions:
      secretName: "Secret name consists of token and fingerprint"
    nbcaUpdateOptions:
      secretName: "Secret name consists of token and fingerprint"
      force: true
    nbcaRemoveOptions:
      hostID: "hostId of the nbca certificate. You can view on
Netbackup UI"
```

表 3-1 ホスト ID ベースの証明書操作

操作形式	オプションとコメント
作成 (Create)	secretName: トークンと指紋を含む Secret の名前。
削除 (Remove)	hostID: NBCA 証明書のホスト ID。
更新 (Update)	secretName: トークンと指紋を含む Secret の名前。

Kubernetes Operator 用ホスト ID ベースの証明書の作成

次の手順を使用して、Kubernetes Operator 用にホスト ID ベースの証明書を作成できます。

Kubernetes Operator 用ホスト ID ベースの証明書を作成するには

- 1 バックアップサーバーで、次のコマンドを実行し、SHA-256 指紋を取得します。
- 2 認証トークンを作成するには、『NetBackup™ セキュリティおよび暗号化ガイド』の「認証トークンの作成」を参照してください。
- 3 再発行トークンを作成するには、必要に応じて、『NetBackup™ セキュリティおよび暗号化ガイド』の「再発行トークンの作成」を参照してください。
- 4 トークンと指紋を使用して **Secret** を作成します。
- 5 セキュリティレベルに関係なく必須のため、トークンを指定します。

Token-fingerprint-secret.yaml は次のようになります。

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-name
  namespace: kops-ns
type: Opaque
stringData:
  token: "Authorization token | Reissue token"
  fingerprint: "SHA256 Fingerprint"
```

- Token-fingerprint-secret.yaml ファイルのテキストをコピーします。
- テキストエディタを開き、yaml ファイルのテキストを貼り付けます。
- その後、そのテキストに **yaml** ファイル拡張子を付けて、Kubernetes クラスタにアクセスできるホームディレクトリに保存します。

- 6 Token-fingerprint-secret.yaml ファイルを作成するには、コマンド `kubectl create -f Token-fingerprint-secret.yaml` を実行します。

- 7 nbcaCreateOptions を使用して

backupservercert オブジェクトを作成し、**Secret** 名を指定します。

nbca-create-backupservercert.yaml は次のようになります。

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupserver-nbca-create
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
```

```
backupServer: backupserver.sample.domain.com
certificateOperation: Create
certificateType: NBCA
nbcaAttributes:
  nbcaCreateOptions:
    secretName: nbcaSecretName with token and fingerprint
```

- nbca-create-backupservercert.yaml ファイルのテキストをコピーします。
 - テキストエディタを開き、yaml ファイルのテキストを貼り付けます。
 - その後、そのテキストに yaml ファイル拡張子を付けて、Kubernetes クラスタにアクセスできるホームディレクトリに保存します。
- 8 nbca-create-backupservercert.yaml ファイルを作成するには、コマンド `kubectl create -f nbca-create-backupservercert.yaml` を実行します。
- 9 証明書が作成されたら、カスタムリソースの状態を確認します。カスタムリソースの状態が正常の場合は、スナップショットからのバックアップジョブを実行できます。

メモ: スナップショットからのバックアップ操作またはバックアップコピーからのリストア操作を開始する前に、BackupServerCert カスタムリソースの状態が正常であることを確認する必要があります。

メモ: ホスト ID ベースの証明書を更新するには、NetBackup のホスト ID 証明書で、24 時間後に更新が予定されているかどうかを確認します。証明書は、有効期限の 180 日 (6 カ月) 前に自動的に更新されます。

メモ: NetBackup プライマリサーバーのクロックと NetBackup Kubernetes Operator のクロックが同期しているかどうかを確認します。CheckClockSkew のエラーについて詳しくは、『NetBackup™ セキュリティおよび暗号化ガイド』の「証明書の有効期間に対するクロックスキューの意味」セクションを参照してください。

Kubernetes Operator からのプライマリサーバー証明書の削除

プライマリサーバーがバックアップおよびリストア操作の実行に使用されていない場合は、そのサーバーから証明書を削除できます。

Kubernetes Operator からプライマリサーバー証明書を削除するには

1 NetBackup Web UI にログインし、削除する証明書のホスト ID を取得します。

証明書のホスト ID を取得するには、『NetBackup™ セキュリティおよび暗号化ガイド』の「ホスト ID ベースの証明書の詳細の表示」セクションを参照してください。

2 操作形式を削除に設定して backupservercert を作成します。

nbca-remove-backupservercert.yaml ファイルは次のようになります。

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupserver-nbca-domain.com
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: backupserver.sample.domain.com
  certificateOperation: Remove
  certificateType: NBCA
  nbcaAttributes:
    nbcaRemoveOptions:
      hostID: nbcahostID
```

- nbca-remove-backupservercert.yaml ファイルのテキストをコピーします。
- テキストエディタを開き、yaml ファイルのテキストを貼り付けます。
- その後、そのテキストに yaml ファイル拡張子を付けて、Kubernetes クラスタにアクセスできるホームディレクトリに保存します。

3 nbca-remove-backupservercert.yaml ファイルを作成するには、コマンド `kubectl create -f nbca-remove-backupservercert.yaml` を実行します。

4 証明書を無効にするには、『NetBackup™ セキュリティおよび暗号化ガイド』の「ホスト ID ベースの証明書の無効化」セクションを参照してください。

メモ: nbca-remove-backupservercert.yaml が適用されると、証明書は Kubernetes Operator のローカル証明書ストアから削除されます。ただし、まだ NetBackup データベースには存在し、有効なままです。したがって、証明書を無効にする必要があります。

プライマリサーバー証明書の更新

証明書が読み取り可能で、Kubernetes Operator に存在することを前提に、証明書を更新するシナリオを次に示します。

NetBackup Kubernetes Operator に存在する証明書が無効化されている場合は、更新操作を行って証明書を再発行できます。この問題を解決するには、サーバー証明書を更新するか、サーバー証明書を削除して新しい証明書を作成します。

メモ: 証明書の更新操作が失敗した場合は、最初に証明書を削除してから新しい証明書を作成する必要があります。

Kubernetes Operator でプライマリサーバー証明書を更新するには

- 1 更新操作を行って `backupservercert` オブジェクトを作成します。

`nbca-update-backupservercert.yaml` ファイルは次のようになります。

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupserver-nbca-update
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: backupserver.sample.domain.com
  certificateOperation: Update
  certificateType: NBCA
  nbcaAttributes:
    nbcaUpdateOptions:
      secretName: "Name of secret containing
token and fingerprint"
      force: true
```

- `nbca-update-backupservercert.yaml` ファイルのテキストをコピーします。
- テキストエディタを開き、`yaml` ファイルのテキストを貼り付けます。
- その後、そのテキストに `yaml` ファイル拡張子を付けて、Kubernetes クラスタにアクセスできるホームディレクトリに保存します。

- 2 `nbca-update-backupservercert.yaml` ファイルを作成するには、コマンド `kubectl create -f nbca-update-backupservercert.yaml` を実行します。
- 3 `backupservercert` オブジェクトが作成されたら、カスタムリソースの状態を確認します。

ECA 証明書操作の実行

外部認証局 (ECA) の作成、更新、削除の各操作を実行する前に、ECA モードでバックアップサーバーを構成する必要があります。

ECA モードがオンかどうかを確認するには、コマンド

`/usr/opensv/netbackup/bin/nbcertcmd -getSecConfig -caUsage` を実行します。

出力は次のようになります。

```
NBCA: ON
```

```
ECA: ON
```

ECA モードでバックアップサーバーを構成するには、『NetBackup™ セキュリティおよび暗号化ガイド』の「NetBackup での外部 CA のサポートについて」を参照してください。

ECA 証明書の指定は次のようになります。

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupservercert-sample-eca
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: primaryserver.sample.domain.com
  certificateOperation: Create | Update | Remove
  certificateType: ECA
  ecaAttributes:
    ecaCreateOptions:
      ecaSecretName: "Secret name consists of cert, key, passphrase,
cacert"
      copyCertsFromSecret: true | false
      isKeyEncrypted: true | false
    ecaUpdateOptions:
      ecaCrlCheck: DISABLE | LEAF | CHAIN
      ecaCrlRefreshHours: range[0,4380]
```

表 3-2 ECA 証明書操作

操作形式	オプションとコメント
作成 (Create)	■ secretName: 証明書、キー、パスフレーズ、cacert を含む Secret の名前。 ■ copyCertsFromSecret: 指定可能な値は true および false です。このオプションは、外部 CA がすべてのプライマリサーバーで共通しているため、追加されます。すべてのプライマリサーバーの Kubernetes Operator に同じ証明書を登録できます。したがって、証明書とキーを毎回コピーする必要はありません。証明書とキーのコピーは、このオプションを使用して制御できます。 ■ isKeyEncrypted: 秘密鍵が暗号化されている場合はこのフィールドを true に設定し、それ以外の場合は false に設定します。 証明書とキーの問題が原因で ECAHealthCheck が失敗した場合は、証明書を再度コピーする必要があります。
削除 (Remove)	なし
更新 (Update)	■ ecaCrlCheck: 外部証明書の失効の確認レベルを指定できます。指定可能な値は LEAF、CHAIN、DISABLE です。 ■ ecaCrlRefreshHours: 証明書失効リストをダウンロードする間隔 (時間単位) を指定します。指定可能な値の範囲は 0 から 4380 までです。

ECA が署名した証明書の作成

NetBackup は、Kubernetes Operator に登録された ECA の複数のプライマリサーバーでの使用をサポートしています。外部 CA がプライマリサーバーで共通している場合、通信中に証明書失効リストを動的にフェッチするには、証明書失効リスト配布ポイントを使用する必要があります。

ECA が署名した証明書を作成するには

- 1 証明書失効リスト配布ポイントを使用して、証明書失効リストをフェッチします。
- 2 ECA が署名した証明書チェーン、秘密鍵、(必要な場合は) パスフレーズをホームディレクトリに準備しておきます。
- 3 手順 2 で説明した各ファイルでサポートされているさまざまな形式 (DER、PEM など) を識別します。詳しくは、『NetBackup™ セキュリティおよび暗号化ガイド』の「外部 CA が署名した証明書の構成オプション」セクションを参照してください。
- 4 手順 3 で説明したファイルを使用して **Secret** を作成します。
 - 秘密鍵が暗号化されていない場合に **Secret** を作成するには、コマンド `kubectl create secret generic <Name of secret>` を実行します。

```
--from-file=cert_chain=<File path to ECA signed certificate
chain> --from-file=key=<File path to private key>
--from-file=cacert=<File path to External CA certificate> -n
<Namespace where kops is deployed>
```

- 秘密鍵が暗号化されている場合に **Secret** を作成するには、コマンド `kubectl create secret generic <Name of secret>` を実行します。

```
--from-file=cert_chain=<File path to ECA signed certificate
chain> --from-file=key=<File path to private key>
--from-file=cacert=<File path to External CA certificate>
--from-file=passphrase=<File path to passphrase
of encrypted private key> -n <Namespace where kops is deployed>
```

ディレクトリ構造は次のようになります。

```
|— cert_chain.pem
|— private
|   |__key.pem
|   |__passphrase.txt
|__trusted
    |__cacerts.pem
```

`cert_chain.pem` は ECA が署名した証明書チェーンです。

`private/key.pem` は秘密鍵です。

`private/passphrase.txt` は秘密鍵のパスフレーズです。

`trusted/cacerts.pem` は外部 CA 証明書です。

- 秘密鍵が暗号化されていない場合に名前 `eca-secret` の **Secret** を作成するには、次のコマンドを実行します。

```
kubectl create secret generic
eca-secret--from-file=cert_chain=cert_chain.pem
--from-file=key=private/key.pem
--from-file=cacert=trusted/cacerts.pem -n kops-ns
```

- 秘密鍵が暗号化されている場合に名前 `eca-secret` の **Secret** を作成するには、次のコマンドを実行します。

```
kubectl create secret generic eca-secret
--from-file=cert_chain=cert_chain.pem
--from-file=key=private/key.pem
--from-file=cacert=trusted/cacerts.pem
--from-file=passphrase=private/passphrase.txt
-n kops-ns
```

- 5 **Secret** が作成されたら、backupservercert オブジェクトのカスタムリソースを作成します。

eca-create-backupservercert.yaml ファイルは次のようになります。

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupservercert-eca-create
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: backupserver.sample.domain.com
  certificateOperation: Create
  certificateType: ECA
  ecaAttributes:
    ecaCreateOptions:
      ecaSecretName: eca-secret
      copyCertsFromSecret: true
      isKeyEncrypted: false
```

- eca-create-backupservercert.yaml ファイルのテキストをコピーします。
- テキストエディタを開き、yaml ファイルのテキストを貼り付けます。
- その後、そのテキストに **yaml** ファイル拡張子を付けて、**Kubernetes** クラスタにアクセスできるホームディレクトリに保存します。

- 6 証明書とキーを **Kubernetes Operator** にコピーするには、次の操作のいずれかを実行します。

- copyCertsFromSecret を **true** に設定します。
- **Kubernetes Operator** に存在する証明書とキーのコピーを回避するには、copyCertsFromSecret を **false** に設定します。

メモ: ECA はすべてのプライマリサーバーで共通しているため、**Kubernetes Operator** では、必要に応じてすべてのプライマリサーバーに登録できる 1 組の証明書とキーが必要です。以前にコピーした証明書とキーに問題がないかぎり、証明書とキーを毎回コピーする必要はありません。

メモ: 証明書やキーに関連した理由 (破損、期限切れ、または ECA の変更) が原因で ecaHealthCheck が失敗した場合は、エラーの原因を特定し、フラグを使用して有効な証明書のコピーを実行します。

- 7 秘密鍵が暗号化されている場合は `isKeyEncrypted` フラグを **true** に設定し、暗号化されていない場合は **false** に設定します。秘密鍵が暗号化されている場合は、パスフレーズが **Secret** で指定されていることを確認します。
- 8 手順 5 で `backupservercert.yaml` を作成した **Secret** 名を使用して `ecaSecretName` を設定します。
- 9 `eca-create-backupservercert.yaml` ファイルを作成するには、コマンド `kubectl create -f eca-create-backupservercert.yaml` を実行します。
- 10 `backupservercert` カスタムリソースが作成されたら、カスタムリソースの状態を確認します。
- 11 NetBackup Web UI で外部証明書の詳細を表示するには、『NetBackup™ Web UI 管理者ガイド』の「ドメイン内の NetBackup ホストの外部証明書情報の表示」セクションを参照してください。

ECA が署名した証明書の削除

ECA が署名した証明書をプライマリサーバーから削除できます。

ECA が署名した証明書を削除するには

- 1 操作を削除、証明書の種類を **ECA** に設定して `backupservercert` を作成します。
`eca-remove-backupservercert.yaml` ファイルは次のようになります。


```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupservercert-eca-remove
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: backupserver.sample.domain.com
  certificateOperation: Remove
  certificateType: ECA
```

 - `eca-remove-backupservercert.yaml` ファイルのテキストをコピーします。
 - テキストエディタを開き、**yaml** ファイルのテキストを貼り付けます。
 - その後、そのテキストに **yaml** ファイル拡張子を付けて、**Kubernetes** クラスタにアクセスできるホームディレクトリに保存します。
- 2 `eca-remove-backupservercert.yaml` ファイルを作成するには、コマンド `kubectl create -f eca-remove-backupservercert.yaml` を実行します。
- 3 オブジェクトが作成されたら、カスタムリソースの状態を確認する必要があります。失敗した場合は、必要な措置を講じることができます。

上記の手順を実行すると、指定したプライマリサーバーに関する外部証明書の詳細がローカル証明書ストアから削除されます。証明書は、システムからも NetBackup データベースからも削除されません。

ECA を無効にする場合は、『NetBackup™ セキュリティおよび暗号化ガイド』の「NetBackup ドメインでの外部 CA の無効化」セクションを参照してください。

バックアップサーバーの Kubernetes Operator に ECA を登録したものの、その後 NBCA のみをサポートするバックアップサーバーを再インストールした場合は、Kubernetes Operator から ECA の登録を削除する必要があります。これは、*nbcertcmd* の実行時に、バックアップサーバーの CA のサポートとの通信が比較されることがあり、不一致の場合にエラーが発生するためです。

ECA が署名した証明書の更新

ECA で設定可能な特定のオプションがあります。更新操作でこれらのオプションを設定できます。

ECA が署名した証明書を更新するには

- 1 操作形式を更新に設定して `backupservercert` オブジェクトを作成します。

`eca-update-backupservercert.yaml` ファイルは次のようになります。

```
apiVersion: netbackup.veritas.com/v1
kind: BackupServerCert
metadata:
  name: backupservercert-eca-update
  namespace: kops-ns
spec:
  clusterName: cluster.sample.com:port
  backupServer: backupserver.sample.domain.com
  certificateOperation: Update
  certificateType: ECA
  ecaAttributes:
    ecaUpdateOptions:
      ecaCrlCheck: DISABLE | LEAF | CHAIN
      ecaCrlRefreshHours: [0,4380]
```

- `eca-update-backupservercert.yaml` ファイルのテキストをコピーします。
- テキストエディタを開き、`yaml` ファイルのテキストを貼り付けます。
- その後、そのテキストに `yaml` ファイル拡張子を付けて、Kubernetes クラスタにアクセスできるホームディレクトリに保存します。

- 2 `eca-update-backupservercert.yaml` ファイルを作成するには、コマンド `kubectl create -f eca-update-backupservercert.yaml` を実行します。

- 3 ECA_CRL_CHECK オプションを使用すると、ホストの外部証明書の失効の確認レベルを指定できます。外部証明書の失効の確認を無効にすることもできます。確認に基づいて、ホストとの通信時に、証明書失効リスト (CRL) に対して証明書の失効状態が検証されます。詳しくは、『NetBackup™ セキュリティおよび暗号化ガイド』の「NetBackup サーバーとクライアントの ECA_CRL_CHECK」セクションを参照してください。
- 4 ECA_CRL_REFRESH_HOURS オプションは、ピアホスト証明書の証明書失効リスト配布ポイント (CDP) で指定した URL から CRL をダウンロードする間隔 (時間単位) を指定します。詳しくは、『NetBackup™ セキュリティおよび暗号化ガイド』の「NetBackup サーバーとクライアントの ECA_CRL_REFRESH_HOURS」セクションを参照してください。

証明書の種類の識別

NetBackup は、Kubernetes Operator に登録されている証明書の種類を識別するのに役立ちます。

証明書の種類を識別するには

- 1 Kubernetes Operator ポッドを一覧表示するには、コマンド `kubectl get pods -n <namespace of Kubernetes operator>` を実行します。
- 2 管理者権限を使用して Kubernetes Operator にログオンし、次のコマンドを実行します。

```
kubectl exec pod/nbu-controller-manager-7c99fb8474-hzrsl -n  
<namespace of Kubernetes operator> -c netbackupkops -it -- bash
```

- 3** Kubernetes に NBCA 証明書を使用しているバックアップサーバーを一覧表示するには、次のコマンドを実行します。

```
/nbcertcmdtool/nbcertcmdtool -atLibPath/nbcertcmdtool/  
-standalone -installDir "/usr/opensv" -listCertDetails -NBCA
```

出力は次のようになります。

```
Master Server : masterserver.sample.domain.com  
Host ID : b06738f0-a8c1-47bf-8d95-3b9a41b7bb0a  
Issued By : /CN=broker/OU=NBCANBKOps  
Serial Number : 0x508cdf4500000008  
Expiry Date : Dec 22 05:46:32 2022 GMT  
SHA-1 Fingerprint : 4D:7A:D9:B9:61:4E:93:29:B8:93:0B:E0:  
07:0A:28:16:46:F6:39:C6  
SHA-256 Fingerprint : C2:FA:AC:B5:21:6B:63:49:30:AC:4D:5E:  
61:09:9A:8C:C6:40:4A:44:B6:39:7E:2B:B3:36:DE:D8:F5:D1:3D:EF  
Key Strength : 2048  
Subject Key Identifier : AC:C4:EF:40:7D:8D:45:B4:F1:89:DA:FB:  
E7:FD:0F:FD:EC:61:12:C6  
Authority Key Identifier : 01:08:CA:40:15:81:75:7B:37:9F:51:78:  
  
B2:6A:89:A1:44:2D:82:2B
```

- 4 Kubernetes に ECA 証明書を使用しているバックアップサーバーを一覧表示するには、次のコマンドを実行します。

```
/nbcertcmdtool/nbcertcmdtool -atLibPath/nbcertcmdtool/  
-standalone -installDir"/usr/opensv" -listCertDetails -ECA
```

出力は次のようになります。

```
Subject Name : CN=ECA-KOPS,O=Veritas,OU=ECANBKOps  
Issued By : CN=ICA-2,O=Veritas,OU=ECANBKOps  
Serial Number : 0x56cf16040258d3654339b7f39817de89240d58  
Expiry Date : Dec 16 05:48:16 2022 GMT  
SHA-1 Fingerprint : 70:DE:46:72:57:56:4E:47:DB:82:8B:8D:A3:  
4B:BB:F9:8D:2C:B7:8E  
SHA-256 Fingerprint : E0:69:5F:79:A6:60:DB:7B:69:76:D3:A8:  
E6:E1:F2:0D:8C:6C:E6:4E:C4:5D:A4:77:17:5A:C2:42:89:74:15:7D  
Key Strength : 2048  
Subject Key Identifier : F0:E7:1F:8C:50:FD:4D:25:40:69:77:6C:  
2A:35:72:B6:1D:8E:E5:17  
Authority Key Identifier : D7:53:57:C7:A6:72:E3:CB:73:BD:48:51:  
  
2F:CB:98:A3:0B:8B:BA:5C  
Master Server : masterserver.sample.domain.com  
Host ID : b85ba9bf-02a8-439e-b787-ed52589c37d1
```

Kubernetes 資産の管理

この章では以下の項目について説明しています。

- [Kubernetes クラスタの追加](#)
- [設定を行う](#)
- [資産への保護の追加](#)
- [マルウェアのスキャン](#)

Kubernetes クラスタの追加

NetBackup に Kubernetes クラスタを追加する前に、クラスタに Kubernetes Operator をインストールして構成する必要があります。そうしないと、クラスタの検証が失敗し、さらにクラスタの追加操作が失敗します。

Kubernetes Operator を構成すると、NetBackup に Kubernetes クラスタを追加し、クラスタ内のすべての資産を自動的に検出できます。

クラスタを追加するには

- 1 左側の[作業負荷 (Workloads)]で、[Kubernetes]をクリックします。
- 2 [Kubernetes クラスタ (Kubernetes clusters)]タブをクリックし、[追加 (Add)]をクリックします。
- 3 [Kubernetes クラスタの追加 (Add Kubernetes cluster)]ページで、次を入力します。
 - [クラスタ名 (Cluster name)]: クラスタの名前を入力します。この名前は DNS の解決可能な値または IP アドレスである必要があります。例:
cluster.sample.domain.com。
 - [ポート (Port)]: Kubernetes API サーバーのポート番号を入力します。

- [コントローラの名前空間 (Controller namespace)]: Kubernetes クラスタ内で NetBackup Kubernetes Operator が配備されている名前空間を入力します。
例: kops-ns。
- 4 [次へ (Next)]をクリックします。[クレデンシャルの管理 (Manage credentials)]ページで、クラスタにクレデンシャルを追加できます。
- 既存のクレデンシャルを使用するには、[既存のクレデンシャルから選択してください (Select from existing credentials)]を選択し、[次へ (Next)]をクリックします次のページで、必要なクレデンシャルを選択し、[次へ (Next)]をクリックします。
 - 新しいクレデンシャルを作成するには、[クレデンシャルの追加 (Add credential)]をクリックし、[次へ (Next)]をクリックします。[クレデンシャルの管理 (Manage credentials)]ページで、次を入力します。
 - [クレデンシャル名 (Credential name)]: クレデンシャルの名前を入力します。
 - [タグ (Tag)]: クレデンシャルに関連付けるタグを入力します。
 - [説明 (Description)]: クレデンシャルの説明を入力します。
 - NetBackup に Kubernetes クラスタを追加するには、認証局 (CA) 証明書とトークンが必要です。Kubernetes クラスタの認証と認可には、CA 証明書とバックアップサービスアカウントのトークンが必要です。CA 証明書とトークンを取得するには、Kubernetes クラスタでコマンド `kubectl get secret <[namespace-name]-backup-server-secret> -n <namespace name> -o yaml` を実行します。
 - [トークン (Token)]: 認証トークンの値を Base64 エンコード形式で入力します。
 - [CA 証明書 (CA certificate)]: CA 証明書ファイルの内容を指定します。
- 5 [次へ (Next)]をクリックします。

クレデンシャルが検証され、検証に成功すると、クラスタが追加されます。クラスタが追加されると、自動検出が実行され、クラスタ内の利用可能な資産が検出されます。

メモ: NetBackup Kubernetes バージョン 10.1 で、クラスタの編集操作が失敗し、エラーメッセージが表示されます。この問題を解決するには、最初にクラスタを削除し、クラスタを再び追加することをお勧めします。

設定を行う

Kubernetes の設定では、Kubernetes の配備のさまざまな側面を構成できます。

Kuberentes リソース形式のリソース制限の変更

リソース制限の設定について

この設定によって、Kubernetes クラスタで同時に実行できるバックアップの数を制御できます。Kubernetes では、スナップショットジョブを実行する場合のデフォルト値は 1、スナップショットからのバックアップジョブを実行する場合のデフォルト値は 4 とそれぞれ異なります。

例:

20 の資産を保護し、制限を 5 に設定している場合に、スナップショットのみのバックアップジョブを実行すると、5 つの資産のみ同時にバックアップを実行でき、残りの 15 の資産はキューに入ります。最初の 5 つの資産のうち 1 つのバックアップが完了すると、キューの資産にバックアップの順番が回ります。

スナップショットジョブを実行する場合、リソース制限のデフォルト値は 1 です。これは、クラスタごとに 1 つのバックアップジョブのみが進行中になり、残りの資産はキューに投入された状態になることを示します。

システムとネットワークリソースの使用を最適化するため、この設定をお勧めします。この設定は、選択しているプライマリサーバーのすべての Kubernetes バックアップに適用されます。

リソース制限を設定するには

- 1 左側で[作業負荷 (Workloads)]、[Kubernetes]の順にクリックします。
- 2 右上で[Kubernetes 設定 (Kubernetes settings)]、[リソース制限 (Resource limits)]の順にクリックします。
- 3 リソース制限を設定するには、次のいずれかを実行します。
 - [Kubernetes クラスタあたりのバックアップジョブ (Backup jobs per Kubernetes cluster)]の横にある[編集 (Edit)]をクリックします。デフォルトでは、制限は 1 です。
これは、クラスタごとに並行して処理される名前空間の量を定義します。また、スナップショットからのバックアップジョブにスナップショットを使用してバックアップを作成するための最初の操作にも適用されます。
デフォルトでは、クラスタあたりのバックアップジョブのリソース制限は 1 です。
 - [Kubernetes クラスタあたりのスナップショットからのバックアップジョブ (Backup from Snapshot Jobs per Kubernetes Cluster)]の横にある[編集 (Edit)]をクリックします。
これは、スナップショットの作成後にクラスタごとに並行してバックアップされる名前空間の量を定義します。スナップショットからのバックアップでは、それぞれがクラスタの NetBackup 名前空間でデータムーバー ポッドを開始し、データを処理します。

デフォルトでは、クラスタあたりのスナップショットからのバックアップジョブのリソース制限は 4 です。

- 4 [Kubernetes クラスタの編集 (Edit Kubernetes cluster)]ダイアログで、次の操作を行います。
- [グローバル (Global)]フィールドに値を入力し、すべてのクラスタのグローバル制限を設定します。この制限は、クラスタで同時に実行されるバックアップジョブとスナップショットからのバックアップジョブの数を示します。
 - そのクラスタのグローバル制限を上書きする個別の制限をクラスタに追加できます。クラスタに個々の制限を設定するには、[追加 (Add)]をクリックします。
 - リストから利用可能なクラスタを選択し、選択したクラスタの制限値を入力できます。配備されている利用可能な各クラスタに制限を追加できます。
 - [保存 (Save)]をクリックして、変更を保存します。

メモ: NetBackup 10.0 リリースでは、datamover ポッドは Kubernetes のリソース制限の設定を超過します。

p.132 の「[datamover ポッドが Kubernetes のリソース制限を超過](#)」を参照してください。

自動検出の間隔の構成

自動検出により、クラスタ内で NetBackup によって保護される資産数が記録されます。この設定を使用すると、NetBackup が自動検出を実行して、クラスタ内の新しい資産を特定する間隔を設定できます。クラスタから排除または削除された資産の数を収集します。

指定できる値は、5 分から 1 年の間です。デフォルト値は 30 分です。

自動検出の間隔を設定するには

- 1 左側で[作業負荷 (Workloads)]、[Kubernetes]の順にクリックします。
- 2 右上で[Kubernetes 設定 (Kubernetes settings)]、[自動検出 (Autodiscovery)]の順にクリックします。
- 3 [間隔 (Frequency)]の近くにある[編集 (Edit)]をクリックします。
- 4 NetBackup が自動検出を実行した後の時間数を入力します。[保存 (Save)]をクリックします。

完全検出と増分検出の実行

Kubernetes クラスタが追加されると、自動検出サイクルがトリガされ、Kubernetes クラスタで利用可能なすべての資産が検出されます。その日最初の自動検出は完全検出で、以降の自動検出は増分検出です。

検出を実行するには

- 1 左側で[作業負荷 (Workloads)]、[Kubernetes]の順にクリックします。
- 2 **Kubernetes** クラスタのリストで、クラスタ名を見つけます。次に、[処理 (Actions)]、[今すぐ検出 (Discover now)]の順にクリックします。

この場合、増分検出では前回の検出実行以降にクラスタで変更された **NetBackup** 資産のみをフェッチします。したがって、最初の検出は完全検出で、それ以降のすべての検出は増分検出になります。

権限の構成

管理権限を使用して、ユーザーロールに異なるアクセス権を割り当てることができます。詳しくは、『**NetBackup Web UI 管理者ガイド**』の「役割ベースのアクセス制御の管理」の章を参照してください。

資産のクリーンアップ

資産のクリーンアップとは、コストを最適化し、セキュリティを向上し、効率的な操作を維持するために、未使用または古いリソースを特定して削除するプロセスです。

資産のクリーンアップは、特定の基準に基づいて自動的に実行できます。

資産のクリーンアップの前の一般的な条件を次に示します。

- 資産にバックアップイメージがない場合。
- 資産がバックアップポリシーまたは保護計画にサブスクライブされていない場合。
- 資産が構成可能な経過時間より古い場合。

デフォルトで、**NetBackup** は、30 日が経過した資産をクリーンアップします。

資産の自動クリーンアップの経過時間を設定するには

1. [Kubernetes]作業負荷をクリックし、次に、ページの右上にある[Kubernetes 設定 (Kubernetes settings)]をクリックします。
2. [資産のクリーンアップ (Assets cleanup)]をクリックします。[資産のクリーンアップ (Asset cleanup)]ページが表示されます。
3. [編集 (Edit)]をクリックし、経過時間を日数で指定します。
4. [保存 (Save)]をクリックします。

資産への保護の追加

[名前空間 (Namespaces)]タブ ([作業負荷 (Workloads)]、[Kubernetes])を使用して、**Kubernetes** クラスタ内の資産の監視、保護状態の確認、保護されていない資産へ

の保護の追加を簡単に行えます。また、[今すぐバックアップ (Backup now)]機能を使用して資産のクイックバックアップを作成できます。この機能は、スケジュール設定されたバックアップに影響を与えることなく、選択した資産のワンタイムバックアップを作成します。

[名前空間 (Namespaces)]タブに、NetBackup によって保護できる検出済みおよびインポート済みの Kubernetes 資産がすべて表示されます。このタブには、次の情報が表示されます。

- [名前空間 (Namespaces)]: 資産の表示名。
- [クラスター (Cluster)]: 資産が属するクラスター。
- [保護計画名 (Protected by)]: 資産に適用された保護計画の名前。
- [最後に成功したバックアップ (Last successful backup)]: 資産のバックアップが最後に成功した日時。

[名前空間 (Namespaces)]タブで次の操作を実行できます。

保護されていない資産に保護を追加するには

- 1 左側で[作業負荷 (Workloads)]、[Kubernetes]の順にクリックします。
- 2 資産の行でオプションを選択します。右上の[保護の追加 (Add protection)]をクリックします。または、資産の行の[処理 (Actions)]メニューをクリックして、[保護の追加 (Add protection)]をクリックします。
- 3 リストから保護計画を選択し、[次へ (Next)]をクリックします。次のページで、[保護 (Protect)]をクリックします。

資産をすばやくバックアップするには

- 1 資産の行でオプションを選択し、右上の[今すぐバックアップ (Backup now)]をクリックします。または、資産の行の[処理 (Actions)]メニューをクリックして、[今すぐバックアップ (Backup now)]をクリックします。
- 2 次のページで、
 - すでに保護されている資産をバックアップする場合は、資産がすでにサブスクライブされている計画のリストから保護計画を選択し、[バックアップの開始 (Start backup)]をクリックします。
 - 保護されていない資産をバックアップする場合は、その資産で利用可能な計画から保護計画を選択し、[バックアップの開始 (Start backup)]をクリックします。

マルウェアのスキャン

NetBackup バージョン 10.4 以降では、Kubernetes の作業負荷を介して Kubernetes 資産でマルウェアをスキャンするためのサポートが提供されます。

マルウェアスキャンをトリガするには、スキャンホストを構成する必要があります。スキャンホストの構成について詳しくは、『NetBackup セキュリティおよび暗号化ガイド』の「スキャンホストの構成」の章を参照してください。

作業負荷の種類ごとの資産

このセクションでは、Kubernetes VM 資産でマルウェアをスキャンする手順について説明します。

メモ: YARA スキャンでは、Kubernetes のみがサポートされています。

サポート対象の資産でマルウェアをスキャンするには、次の手順を実行します。

- 1 左側の[作業負荷 (Workloads)]で、サポートされている作業負荷を選択します。
- 2 バックアップが完了したリソースを選択します。
例: Kubernetes
- 3 [処理 (Actions)]、[マルウェアのスキャン (Scan for malware)]を選択します。
- 4 [マルウェアスキャン (Malware scan)]ページで、次の操作を行います。
 - [開始日時 (Start date/time)]と[終了日時 (End date/time)]を選択して、スキャンの日付範囲を選択します。
 - [スキャナホストプール (Scanner host pool)]を選択します。
 - [現在の感染状態 (Current infection status)]リストから、次のいずれかを選択します。
 - 未スキャン (Not scanned)
 - 感染なし (Not infected)
 - マルウェアスキャンで検出された感染 (Infection detected by malware scan)
 - ファイルハッシュ検索で検出された感染 (Infection detected by file hash search)
 - すべて (All)
- 5 [マルウェアのスキャン (Scan for malware)]をクリックします。

メモ: マルウェアスキャナホストは、一度に 3 つのイメージのスキャンを開始できます。

- 6 スキャンが開始されると、[マルウェアの検出 (Malware detection)]に[スキャンの状態 (Scan status)]が表示され、次のフィールドが表示されます。
 - 未スキャン (Not scanned)

- 感染なし (Not infected)
- 感染 (Infected)
- 失敗 (Failed)

メモ: 検証で失敗したバックアップイメージは無視されます。

- 処理中 (In progress)
- 保留中 (Pending)

Kubernetes インテリジェントグループの管理

この章では以下の項目について説明しています。

- [インテリジェントグループについて](#)
- [インテリジェントグループの作成](#)
- [インテリジェントグループの削除](#)
- [インテリジェントグループの編集](#)

インテリジェントグループについて

問い合わせと呼ばれるフィルタのセットに基づいて、インテリジェント資産グループを定義して、資産のダイナミックグループを作成および保護できます。**NetBackup** は、問い合わせに基づいて **Kubernetes** 名前空間を選択し、それらをグループに追加します。インテリジェントグループでは、資産の環境内の変更が自動的に反映されるため、環境内で資産を追加または削除しても、グループ内の資産のリストを手動で修正する必要がないことに注意してください。

インテリジェントグループに保護計画を適用すると、問い合わせ条件を満たすすべての資産が自動的に保護されます。

メモ: インテリジェントグループの作成、更新、削除は、管理が必要な資産に対する必要な RBAC 権限が役割に付与されている場合にのみ実行できます。**NetBackup** のセキュリティ管理者は、資産タイプ (クラスタ、名前空間、VMGroup) に対するアクセス権を付与できます。『**NetBackup Web UI 管理者ガイド**』を参照してください。

インテリジェントグループの作成

NetBackup Web UI で Kubernetes 作業負荷のインテリジェントグループを作成するには、次の手順を使用します。

メモ: Kubernetes での仮想化サポートにより、特定のリソースの種類に基づいて名前空間をフィルタ処理するインテリジェントグループを作成できます。仮想マシン、永続ボリューム、永続ボリューム要求は、フィルタ処理で利用できるリソースの種類です。

インテリジェントグループを作成するには

- 1 左側の[作業負荷 (Workloads)]で、[Kubernetes]をクリックします。
- 2 [インテリジェントグループ (Intelligent groups)]タブ、[+ 追加 (+ Add)]の順にクリックします。
- 3 グループの名前と説明を入力します。
- 4 [クラスタ (Cluster)]セクションで、[クラスタの追加 (Add clusters)]をクリックします。
- 5 [クラスタの追加 (Add clusters)]ウィンドウで、一覧から 1 つ以上のクラスタを選択し、[選択 (Select)]をクリックします。選択したクラスタがインテリジェントグループに追加されます。

メモ: インテリジェントグループは、複数のクラスタをまたいで作成できます。グループにクラスタを追加するために必要な権限を持っていることを確認します。グループを表示して管理するには、選択したクラスタとグループに対する表示と管理の権限がグループ管理者に付与されている必要があります。

- 6 [資産の選択 (Select assets)]セクションで、次のいずれかを実行します。
 - [すべての資産を含める (Include all assets)]を選択します。
このオプションでは、デフォルトの問い合わせを使用して、保護計画の実行時にすべての資産をバックアップ対象として選択します。
 - 特定の条件を満たす資産のみを選択するには、独自の問い合わせを作成するために[条件の追加 (Add condition)]をクリックします。
 - 資産のラベル条件を追加するには、[ラベルの条件の追加 (Add label condition)]をクリックして追加します。

- 7 条件を追加するには、ドロップダウンを使用してキーワードと演算子を選択し、値を入力します。

問い合わせの効果を変更するには、[+ 条件 (+ Condition)]をクリックし、[AND]または[OR]をクリックして、キーワード、演算子、条件の値を選択します。

メモ: ラベル条件を追加するには、[ラベル条件の追加 (Add label condition)]をクリックしてラベルキーと値を入力します。

メモ: 条件にラベル値を含めず、ラベルキーのみを含めることもできます。値は、ラベル条件に追加するオプションのパラメータであるためです。

メモ: サブクエリーを追加するには、[サブクエリーの追加 (Add sub-query)]をクリックします。複数のレベルのサブクエリーを追加できます。

- 8 問い合わせをテストするには、[プレビュー (Preview)]をクリックします。

問い合わせベースの選択処理は動的です。Kubernetes クラスタの変更は、保護計画の実行時に問い合わせが選択する資産に影響する可能性があります。その結果、保護計画が後で実行された時に問い合わせが選択する資産が、プレビューに現在表示されているものと同一でなくなる可能性があります。

メモ: [インテリジェントグループ (Intelligent groups)]で問い合わせを使用する場合、問い合わせ条件に英語以外の文字が含まれていると、NetBackup Web UI に、問い合わせに一致する正確な資産のリストが表示されないことがあります。

任意の属性に `not equals` フィルタ条件を使用すると、属性に値が存在しない (null) 資産を含む資産が戻されます。

メモ: [プレビュー (Preview)]をクリックするかグループを保存した場合、グループの資産を選択するときに、問い合わせオプションでは大文字と小文字が区別されます。

- 9 グループを保護計画に追加せずに保存するには、[追加 (Add)]をクリックします。
- 10 グループを保護計画に追加して保存するには、[追加と保護 (Add and protect)]をクリックします。
- 11 保護計画にグループをサブスクライブするには、[保護の追加 (Add protection)]をクリックします。

グループを選択して保護計画を適用し、[保護する (Protect)]をクリックします。

選択した資産グループが保護計画に正常にサブスクライブされます。

資産にラベル条件を追加する際の制限事項

条件とラベルの組み合わせがある場合は、最初に名前空間条件を定義してから、ラベル条件を定義する必要があります。

メモ: 条件については、名前空間の値のみが許可されます。

インテリジェントグループの削除

インテリジェントグループを削除するには

- 1 左側の[作業負荷 (Workloads)]で、[Kubernetes]をクリックします。
- 2 [インテリジェントグループ (Intelligent groups)]タブでグループを見つけます。
- 3 グループが保護されていない場合は、グループを選択して[削除 (Delete)]をクリックします。
- 4 グループが保護されている場合は、グループを選択し、[保護の削除 (Remove protection)]をクリックしてすべての保護計画を削除します。
- 5 次に、[インテリジェントグループ (Intelligent groups)]タブでこのグループを選択し、[削除 (Delete)]をクリックします。

インテリジェントグループの編集

インテリジェントグループの名前と説明の詳細を編集できます。スケジュールバックアップの時間帯や他のオプションなど、保護計画の特定の設定を編集できます。

インテリジェントグループを編集するには

- 1 左側の[作業負荷 (Workloads)]で、[Kubernetes]をクリックします。
- 2 [インテリジェントグループ (Intelligent groups)]タブで、保護を編集するグループをクリックします。
- 3 次のいずれかを実行します。
 - [名前と説明を編集する (Edit name and description)]をクリックして、選択したグループの名前と説明を編集した後、[保存 (Save)]をクリックします。
 - [資産 (Assets)]タブで、[編集 (Edit)]をクリックしてクラスタを追加または削除します。選択した資産の問い合わせ条件を更新し、[保存 (Save)]をクリックします。
グループのクラスタリストを編集したり、グループのクラスタを追加または削除したりできます。選択した資産グループの問い合わせ条件を変更することもできます。
 - [権限 (permissions)]タブで、[追加 (Add)]をクリックして利用可能な役割の権限を更新し、[保存 (Save)]をクリックします。

Kubernetes ポリシーの管理

この章では以下の項目について説明しています。

- [ポリシーの作成](#)

ポリシーの作成

次の手順を使用して、NetBackup Web UI でポリシー形式 **Kubernetes** を使用してバックアップポリシーを作成します。

ポリシーを作成するには

- 1 左側で[保護 (Protection)]、[ポリシー (Policies)]の順に選択します。
- 2 [追加 (Add)]をクリックします。
- 3 [属性 (Attributes)]タブで、次の操作を実行します。
 - [ポリシー名 (Policy name)]フィールドにポリシー名を入力します。
 - [ポリシー形式 (Policy type)]として **Kubernetes** を選択します。
 - 使用する[ポリシーストレージ (Policy storage)]を選択します。
 - その他のポリシー属性を選択または構成します。
- 4 [スケジュール (Schedules)]タブで、必要なすべてのスケジュールを構成します。
たとえば、完全および増分スケジュールを構成します。
- 5 [**Kubernetes**]タブで、次のいずれかを実行します。
 - [インテリジェントグループ (Intelligent group)]を選択して、保護対象とする新規または既存のインテリジェントグループを追加します。
 - [名前空間 (Namespace)]を選択して、保護する名前空間をリストから追加します。

- 6 [リソースの種類とラベルの選択 (Resource kind and label selection)] タブで、次のいずれかを実行します。
 - [すべてのリソースの種類をバックアップに含めます (Include all resource kinds in the backup)] オプションを選択して、バックアップ内のすべてのリソースの種類を含めます。
 - [次のリソースの種類をバックアップから除外します (Exclude the following resource kinds from the backup)] オプションを選択して、リソースの種類を手動で入力するか、バックアップジョブから除外するリソースの種類を選択します。
 - [ラベルの選択 (Label selection)] で、追加するラベルクエリーについて[追加 (Add +)] をクリックします。
- 7 [作成 (Create)] をクリックします。

Kubernetes 資産の保護

この章では以下の項目について説明しています。

- [インテリジェントグループの保護](#)
- [インテリジェントグループからの保護の削除](#)
- [バックアップスケジュールの構成](#)
- [バックアップオプションの構成](#)
- [バックアップの構成](#)
- [A.I.R. \(自動イメージレプリケーション\) と複製の構成](#)
- [ストレージユニットの構成](#)
- [ボリュームモードのサポート](#)
- [アプリケーションの一貫したバックアップの構成](#)

インテリジェントグループの保護

Kubernetes 作業負荷用に Kubernetes 固有の保護計画を作成できます。その後、保護計画にインテリジェントグループをサブスクライブできます。

次の手順を使用して、インテリジェントグループを保護計画にサブスクライブします。

メモ: 自分に割り当てられている RBAC の役割によって、管理するインテリジェントグループと、使用する保護計画にアクセスできるようにする必要があります。

インテリジェントグループを保護するには

- 1 左側で[Kubernetes]をクリックします。
- 2 [インテリジェントグループ (Intelligent groups)]タブで、グループにチェックマークを付けて[保護の追加 (Add protection)]をクリックします。
- 3 保護計画を選択し、[次へ (Next)]をクリックします。
- 4 グループを選択し、[保護する (Protect)]をクリックして保護計画にサブスクライブします。

即時保護のための[今すぐバックアップ (Backup now)]オプション

スケジュール設定された保護計画とは別に、[今すぐバックアップ (Backup now)]オプションを使用してグループをすぐにバックアップし、計画外の状況に対して保護することもできます。

インテリジェントグループからの保護の削除

インテリジェントグループのサブスクライブを、保護計画から解除できます。インテリジェントグループのサブスクライブが保護計画から解除されると、それ以降バックアップは実行されません。

インテリジェントグループから保護を削除するには

- 1 左側の[作業負荷 (Workloads)]で、[Kubernetes]をクリックします。
- 2 [インテリジェントグループ (Intelligent groups)]タブで、保護を削除するグループをクリックします。
- 3 [保護の削除 (Remove protection)]、[はい (Yes)]の順にクリックします。

バックアップスケジュールの構成

Kubernetes 作業負荷の保護計画を作成する際、[バックアップスケジュールの追加 (Add backup schedule)]ダイアログの[属性 (Attributes)]タブでバックアップスケジュールを追加できます。

保護計画の作成方法について詳しくは、『NetBackup Web UI 管理者ガイド』の「保護計画の管理」セクションを参照してください。

Kubernetes バックアップジョブのバックアップスケジュールを追加するには

- 1 左側で[保護 (Protection)]、[保護計画 (Protection plans)]、[追加 (Add)]の順にクリックします。
- 2 [基本プロパティ (Basic properties)]で、[名前 (Name)]と[説明 (Description)]を入力し、[作業負荷 (Workload)]ドロップダウンリストから[Kubernetes]を選択します。
- 3 [次へ (Next)]をクリックします。[スケジュール (Schedules)]で、[スケジュールの追加 (Add schedule)]をクリックします。

[バックアップスケジュールの追加 (Add backup schedule)]タブで、バックアップとスナップショットを保持するためのオプションを構成できます。
- 4 [反復 (Recurrence)]ドロップダウンから、バックアップの頻度を指定します。
- 5 [スナップショットとバックアップコピー (Snapshot and backup copy)]オプションで、以下のいずれかを行います。
 - 保護計画のスナップショットからのバックアップを構成するには、[スナップショットからバックアップを作成 (Create backup from snapshot)]オプションを選択します。[バックアップの保持期間 (Keep backup for)]ドロップダウンを使用して、スナップショットからのバックアップの保持期間を指定します。

メモ: Kubernetes 作業負荷でサポートされるのは、完全バックアップのスケジュールのみです。バックアップ期間は、時間、日、週、月、年単位で設定できます。

デフォルトでは、4 週間がバックアップの保持期間です。

メモ: バックアップコピーのレプリケーションと複製のオプションを有効にするには、[スナップショットからバックアップを作成 (Create backup from snapshot)]オプションを選択する必要があります。

- [スナップショットからバックアップを作成 (Create backup from snapshot)]オプションを選択しなかった場合、デフォルトでは、バックアップジョブを実行するために[スナップショットのみのストレージ (Snapshot only storage)]バックアップが構成されます。
- バックアップのレプリカコピーを作成するには、[スナップショットからバックアップのレプリカコピー (自動イメージレプリケーション)を作成 (Create a replica copy (Auto Image Replication) of the backup from snapshot)]オプションを選択します。
- バックアップの複製コピーを作成するには、[スナップショットからバックアップの複製コピーを作成 (Create a duplicate copy of the backup from snapshot)]オプションを選択します。

- 6 『NetBackup Web UI 管理者ガイド』の「保護計画の管理」セクションにある説明に従って、[開始時間帯 (Start window)] タブでスケジュールの作成を続行します。
- 7 『NetBackup Web UI 管理者ガイド』の「保護計画の管理」セクションにある説明に従って、スナップショットからのバックアップ用に[ストレージオプション (Storage options)]の設定を続行します。

バックアップオプションの構成

保護計画のバックアップオプションを構成できます。

保護計画の作成方法について詳しくは、『NetBackup Web UI 管理者ガイド』の「保護計画の管理」セクションを参照してください。

保護計画の構成時にバックアップオプションを構成するには

- 1 [バックアップオプション (Backup options)] ページの [リソースの種類の選択 (Resource kind selection)] セクションで、次を実行します。
 - デフォルトでは[すべてのリソースの種類をバックアップに含めます。 (Include all resource kinds in the backup)] オプションが選択されており、バックアップジョブのすべてのリソースの種類が含まれています。
 - [次のリソースの種類をバックアップから除外します。 (Exclude the following resource kinds from the backup)] オプションを選択すると、リソースの種類がバックアップジョブから除外されます。[選択 (Select)] をクリックして、静的リストからリソースの種類を選択します。選択したリソースの種類がテキストフィールドに表示されるか、カスタムリソース定義 (CRD) を正しい形式 (type.group) で手動で入力できます。選択したリソースの種類を除外リストから削除できます。カスタムリソースの種類の定義が静的リストにない場合は、カスタムリソース定義 (CRD) を手動で入力できます。たとえば、demo.nbu.com のように入力します。

メモ: リソースの種類の除外リストは、リソースをマッピングするという点で、バックアップ用に選択したラベルより優先されます。

- 2 [ラベルの選択 (Label selection)] セクションで[追加 (Add)] をクリックして、バックアップ用に関連付けられたリソースをマッピングするラベルを追加し、ラベルの接頭辞とキーを入力し、演算子を選択します。含まれているラベルに関連付けられているすべてのリソースが、バックアップジョブに対してマッピングされます。

ラベルに追加できる 4 つの演算子を次に示します。

- 値と等しいラベルキーを入力します。
- すでに存在するラベルキーを、値なしで入力します。
- 一連の値に含まれているラベルキーを入力します。

- 一連の値に含まれていないラベルキーを入力します。

演算子の一連の値に含まれている、または含まれていない複数の値をカンマ区切りで追加できます。

メモ: 条件が正常に適用されるようにするには、選択したラベルがバックアップ時に存在する必要があります。

メモ: ラベルの選択では、複数のラベル条件間で矛盾しないリソースの種類の選択のみを除外する必要があります。

[確認 (Review)] ページには、リソースの種類の除外リストと、リストに含めるために選択したラベル、および選択したストレージユニットが表示されます。

メモ: Kubernetes 作業負荷用に作成された保護計画は編集または削除できます。

Kubernetes 作業負荷用に作成された保護計画はカスタマイズできません。

バックアップの構成

NetBackup では、スナップショットのみとスナップショットからのバックアップという 2 種類のバックアップジョブを Kubernetes 作業負荷で実行できます。Kubernetes Operator のバックアップジョブを構成する手順に従ってください。

Kubernetes 作業負荷でバックアップを実行するには

- 1 左側で[保護 (Protection)]、[保護計画 (Protection plans)]、[追加 (Add)]の順にクリックします。
- 2 [基本プロパティ (Basic properties)]で、[名前 (Name)]と[説明 (Description)]を入力し、[作業負荷 (Workload)]ドロップダウンリストから[Kubernetes]を選択します。
- 3 [次へ (Next)]をクリックします。[スケジュール (Schedules)]で、[スケジュールの追加 (Add schedule)]をクリックします。
[バックアップスケジュールの追加 (Add backup schedule)]タブで、バックアップとスナップショットを保持するためのオプションを構成できます。
- 4 [反復 (Recurrence)]ドロップダウンから、バックアップの頻度を指定します。
- 5 [スナップショットとバックアップコピー (Snapshot and backup copy)]オプションで、以下のいずれかを行います。
 - 保護計画のスナップショットからのバックアップを構成するには、[スナップショットからバックアップを作成 (Create backup from snapshot)]オプションを選択し

ます。[バックアップの保持期間 (Keep backup for)]ドロップダウンを使用して、スナップショットからのバックアップの保持期間を指定します。

メモ: Kubernetes 作業負荷でサポートされるのは、完全バックアップのスケジュールのみです。バックアップ期間は、時間、日、週、月、年単位で設定できます。デフォルトでは、4 週間がバックアップの保持期間です。

メモ: バックアップコピーのレプリケーションと複製のオプションを有効にするには、[スナップショットからバックアップを作成 (Create backup from snapshot)]オプションを選択する必要があります。

- [スナップショットからバックアップを作成 (Create backup from snapshot)]オプションを選択しなかった場合、デフォルトでは、バックアップジョブを実行するために[スナップショットのみのストレージ (Snapshot only storage)]バックアップが構成されます。
 - バックアップのレプリカコピーを作成するには、[スナップショットからバックアップのレプリカコピー (自動イメージレプリケーション) を作成 (Create a replica copy (Auto Image Replication) of the backup from snapshot)]オプションを選択します。
 - バックアップの複製コピーを作成するには、[スナップショットからバックアップの複製コピーを作成 (Create a duplicate copy of the backup from snapshot)]オプションを選択します。
- 6 『NetBackup Web UI 管理者ガイド』の「保護計画の管理」セクションにある説明に従って、[開始時間帯 (Start window)]タブでスケジュールの作成を続行します。
- 7 『NetBackup Web UI 管理者ガイド』の「保護計画の管理」セクションにある説明に従って、スナップショットからのバックアップ用に[ストレージオプション (Storage options)]の設定を続行します。
- [スナップショットからのバックアップ (Backup from Snapshot)]オプションにストレージを選択する場合、選択したストレージユニットには NetBackup バージョン 10.0 以降のメディアサーバーが必要です。
 - ストレージを管理するメディアサーバーには、選択した Kubernetes クラスタへのアクセス権が必要です。
 - メディアサーバーは API サーバーに接続する必要があります。メディアサーバーからのアウトバウンド接続のために、API サーバーに対応するポートを開く必要があります。datamover ポッドはメディアサーバーに接続する必要があります。

A.I.R. (自動イメージレプリケーション) と複製の構成

1 つの NetBackup ドメインで生成されたバックアップは、1 つ以上のターゲット NetBackup ドメインのストレージにレプリケートできます。この処理は A.I.R. (自動イメージレプリケーション) と呼ばれます。

NetBackup Kubernetes は、ある NetBackup ドメインのメディアサーバー重複排除プール (MSDP) から、別のドメインのメディアサーバー重複排除プール (MSDP) への自動イメージレプリケーションをサポートします。NetBackup は、A.I.R. 操作を管理するソースドメインとターゲットドメインで SLP (ストレージライフサイクルポリシー) を使用します。

自動イメージレプリケーション (A.I.R.) は、すべてのスケジュール形式 (差分増分、累積増分、自動のスケジュールを含む) をサポートします。

自動イメージレプリケーションの構成について詳しくは、『NetBackup 管理者ガイド Vol. 1』の「NetBackup のレプリケーションについて」の章を参照してください。

メモ: Kubernetes A.I.R. 構成には、バージョン 10.0.1 以降の NetBackup プライマリサーバーとメディアサーバーが必要です。

Kubernetes バックアップの A.I.R. (自動イメージレプリケーション) と複製を構成するには

- 1 2 台の NetBackup プライマリサーバー間で、自動イメージレプリケーションを構成します。
 - ドメイン間の操作のために、2 台のプライマリサーバー間の信頼関係を確立します。
 - ソースプライマリサーバーにログオンし、左側で[ホスト (Hosts)]、[ホストプロパティ (Host properties)]の順に選択して、ソースプライマリサーバーとターゲットプライマリサーバー間の接続を構築します。
 - ソースプライマリサーバーを選択します。必要に応じて、[接続 (Connect)]をクリックします。次に、[プライマリサーバーの編集 (Edit primary server)]をクリックします。
 - [サーバー (Servers)]をクリックします。[信頼できるプライマリサーバー (Trusted primary servers)]タブで、[追加 (Add)]をクリックしてソースサーバーを追加します。
 - [認証局の検証 (Validate Certificate Authority)]をクリックし、[次へ (Next)]をクリックして認証局の検証に進みます。
 - 信頼できるプライマリサーバーを作成するには、次のオプションから選択します。
 - [信頼できるプライマリサーバーの認証トークンの指定 (Specify authentication token of the trusted primary server)]を選択して既

存のトークンを追加するか、ソースプライマリサーバーに新しいトークンを作成します。

- [信頼できるプライマリサーバーのクレデンシャルの指定 (**Specify credentials of the trusted primary server**)]を選択して、ソースプライマリサーバーのユーザークレデンシャルを追加します。

- [信頼を作成 (**Create trust**)]をクリックします。

ホストプロパティのデータベースが正常に更新されます。

- [保存 (**Save**)]をクリックします。

2 ソースプライマリサーバーでメディアサーバー重複排除プール (MSDP) ストレージを構成し、MSDP ディスクプールにレプリケーションターゲットを追加します。

- 左側で[ストレージ (**Storage**)]、[ディスクストレージ (**Disk storage**)]の順に選択します。
- MSDP ストレージとディスクプールを追加します。
- [ディスクプール (**Disk pools**)]タブ、[追加 (**Add**)]の順に選択します。
 - 信頼できるプライマリサーバーとターゲットストレージサーバーを選択します。
 - [ユーザー名 (**Username**)]フィールドと[パスワード (**Password**)]フィールドに、レプリケーションターゲットサーバーのユーザークレデンシャルを追加します。
- [追加 (**Add**)]をクリックします。

3 ターゲットプライマリサーバーでインポート操作を使用して SLP を作成します。

- 左側で[ストレージ (**Storage**)]、[ストレージライフサイクルポリシー (**Storage lifecycle policy**)]の順に選択します。次に[追加 (**Add**)]をクリックします。
- [ストレージライフサイクルポリシー名 (**Storage lifecycle policy name**)]フィールドにポリシー名を入力し、[追加 (**Add**)]をクリックします。
- [操作 (**Operation**)]リストから、[インポート (**Import**)]を選択します。
- [宛先ストレージ (**Destination storage**)]リストで、MSDP ストレージユニットを選択します。
- [作成 (**Create**)]をクリックします。

4 [スナップショットからバックアップを作成 (**Create backup from snapshot**)]オプションを指定して Kubernetes 保護計画を作成し、レプリケーションコピーオプションを有効にします。

左側で[保護 (**Protection**)]、[保護計画 (**Protection plans**)]の順に選択します。[スケジュール (**Schedules**)]で、[スケジュールの追加 (**Add schedule**)]をクリックします。

- 5 [スナップショットとバックアップコピーのオプション (Snapshot and backup copy options)]セクションで[スナップショットからバックアップを作成 (Create backup from snapshot)]オプションを選択して、レプリケーションおよび複製コピーのオプションを有効にします。
- 6 [スナップショットからバックアップのレプリカコピー (自動イメージレプリケーション) を作成 (Create a replica copy (Auto Image Replication) of the backup from snapshot)]オプションを選択し、レプリカコピーを保持する期間を設定します。

メモ: 自動イメージレプリケーションは、信頼できる NetBackup プライマリサーバーでのみ作成できます。

- 7 [スナップショットからバックアップの複製コピーを作成 (Create a duplicate copy of the backup from snapshot)]オプションを選択し、複製コピーを保持する期間を設定します。
- 8 [追加 (Add)]をクリックします。
- 9 『NetBackup Web UI 管理者ガイド』の「保護計画の管理」セクションにある説明に従って、[開始時間帯 (Start window)]タブでスケジュールの作成を続行します。
- 10 [次へ (Next)]をクリックします。
- 11 [ストレージオプション (Storage options)]タブで、スナップショットからのバックアップ、レプリケート、または複製を行うストレージユニットを選択します。

メモ: スナップショットと複製からのバックアップでは、単純なストレージユニットを追加できます。ただし、レプリケーションの場合は、インポートストレージライフサイクルポリシー (SLP) を使用して、信頼できるストレージユニットを追加する必要があります。

- 12 選択したバックアップオプションの右側にある[編集 (Edit)]をクリックして、バックアップ用に選択したストレージユニットを変更します。
 - レプリカコピーオプションの場合は、レプリケーションコピー用のプライマリサーバーを選択します。次に[次へ (Next)]をクリックします。
 - 信頼できるサーバーで定義されているインポートストレージライフサイクルポリシーを選択し、[選択したレプリケーションターゲットを使用 (Use selected replication target)]をクリックします。
- 13 ウィザードの手順を続行します。

ストレージユニットの構成

保護計画では、すべての形式のストレージユニットをバックアップ用に構成できます。

メモ: バックアップジョブでは、ストレージライフサイクルポリシー (SLP) でサポートされるすべてのストレージ形式がサポートされます。

ストレージユニットをバックアップ用に構成するには

- 1 左側で[ストレージ (Storage)]、[ストレージユニット (Storage units)]の順に選択します。
- 2 [ストレージユニット (Storage units)]タブをクリックし、[追加 (Add)]をクリックして、ストレージユニットの構成を追加します。
- 3 リストからストレージの形式を選択します。
- 4 [カテゴリ (Category)]を選択し、[開始 (Start)]をクリックします。
- 5 [名前 (Name)]フィールドにストレージユニット名を入力します。
- 6 [最大並列実行ジョブ数 (Maximum concurrent jobs)]フィールドで、バックアップジョブの最大数を選択します。
- 7 [最大フラグメントサイズ (Maximum fragment size)]フィールドで、ストレージユニットのフラグメントサイズの最大数を選択し、[次へ (Next)]をクリックします。
- 8 [ディスクプール (Disk pool)]で、ストレージユニットで使用するディスクプールを選択し、[次へ (Next)]をクリックします。
- 9 [オンデマンドのみ (On demand only)]オプションはストレージユニットがオンデマンドで排他的に利用可能かどうかを指定します。このストレージユニットを使用するためにポリシーまたはスケジュールを明示的に構成する必要があります。
- 10 [メディアサーバー (Media servers)]タブで、使用するメディアサーバーを選択し、[次へ (Next)]をクリックします。NetBackup がメディアサーバーを自動で選択するか、ラジオボタンを使用してメディアサーバーを手動で選択できます。
 - すべてのメディアサーバーが NetBackup バージョン 10.0 以降である必要があります
 - ストレージを管理するすべてのメディアサーバーには、選択した Kubernetes クラスターへのアクセス権が必要です。
 - メディアサーバーは API サーバーに接続する必要があります。メディアサーバーからのアウトバウンド接続のために、API サーバーに対応するポートを開く必要があります。datamover ポッドはメディアサーバーに接続する必要があります。

- 11 ストレージユニットの設定を確認し、[保存 (Save)]をクリックします。
- 12 スケジュールバックアップまたは今すぐバックアップのジョブの詳細を確認するには、[アクティビティモニター (Activity monitor)]タブで[ジョブ ID (Job ID)]をクリックして、バックアップジョブの詳細を表示します。ファイルモードの場合、[ジョブの詳細 (Job Details)]セクションですべてのイメージのバックアップ済みファイルの合計数を確認できます。

ボリュームモードのサポート

NetBackup Kubernetes は、次の機能を使用してサポートします。

- 次の機能をサポートする CSI (Container Storage Interface) プロバイダにおけるモードファイルシステムまたはブロック (あるいはその両方) の PVC (永続ボリューム要求) のバックアップとリストア:
 - PVC スナップショット機能。
 - NFS (Network File System) または他の非ブロックストレージに基づく PVC ボリュームプロビジョニング。
 - ブロックストレージに基づく PVC ボリュームプロビジョニング。

メモ: ボリュームが混在する (VolumeMode: ファイルシステムとブロック) 名前空間のバックアップとリストアは、NetBackup 10.3 以降でサポートされるようになりました。

アプリケーションの一貫したバックアップの構成

データベースなどのアプリケーションを実行しているいくつかのポッドには、アプリケーションの一貫したバックアップを取得するために追加の手順が必要です。

アプリケーションの一貫したバックアップでは、アプリケーションメタデータ、メモリ内の状態、永続ストレージに存在する永続データを把握するためのメカニズムが必要です。リストア中に正常な状態を実現するために、これらのすべての Kubernetes リソースにわたって一貫したアプリケーションバックアップを取得することは、リカバリ処理の合理化に役立ちます。クラッシュ整合バックアップのみが必要な場合、これらの手順は必要ありません。

アプリケーションには、アプリケーションの整合性スナップショットを作成するために入出力 (I/O) 操作を一時停止する手順があり、これはベンダーによって文書化されています。この手順はアプリケーションによって異なるため、それぞれの実行方法の性質が重要になります。これらの手順の内容はお客様の責任です。

NetBackup を使用して Kubernetes 作業負荷を保護する場合、アプリケーションの整合性スナップショットを作成するには、バックアップフックを利用するアプリケーションポッドの注釈を適用します。Kubernetes の注釈は、任意の Kubernetes リソースに適用できる単なるメタデータです。Kubernetes 内のフックはユーザー定義の処理で、任意のコマン

ドまたは複数のコマンドを指定できます。**Kubernetes** インフラ内で、静止状態を必要とするアプリケーションポッドにこれらの注釈とフックを適用します。

バックアップフックは、前処理 (スナップショットの前) と後処理 (スナップショットの後) の両方に使用されます。データ保護のコンテキストでは、通常、**netbackup-pre-backup** フックが静止プロシージャまたはコマンドを呼び出し、**netbackup-post-backup** フックが静止解除のプロシージャまたはコマンドを呼び出すことを意味します。フックの各セットで、コマンドとその適用先のコンテナを指定します。コマンドはコンテナのシェル内では実行されないことに注意してください。したがって、指定の例では、ディレクトリを含む完全なコマンド文字列が使用されます。

アプリケーションの一貫したバックアップを必要とするアプリケーションを識別し、**Kubernetes** データ保護の構成の一部として一連のバックアップフックを含む注釈を適用します。

ポッドに注釈を追加するには、**Kubernetes UI** (ユーザーインターフェース) を使用します。または、特定のポッドまたはラベルに対して、**Kubernetes** クラスターコンソールで `kubectl` の注釈機能を使用します。注釈を適用する方法はディストリビューションによって異なる場合があるため、次の例では、ほとんどのディストリビューションでの広範な可用性に基づいて、`kubectl` コマンドに重点を置いて説明します。

さらに、配備リソースやレプリカセットリソースなどの基本 **Kubernetes** オブジェクトに注釈を追加して、新しく配備されたポッドに注釈を確実に含めることができます。**Kubernetes** 管理者は注釈を動的に更新できます。

ラベルは、ポッドやサービスなどの **Kubernetes** オブジェクトに関連付けられるキーと値のペアです。ラベルは、ユーザーにとって重要で関連性のあるオブジェクトの属性として使用されます。ラベルは作成時にオブジェクトにアタッチでき、その後いつでも追加および変更できます。**Kubernetes** では、これらのラベルを使用して、選択したサブセットに対してオブジェクトを問い合わせ、一括操作を実行するための統合サポートが提供されます。各オブジェクトには、キーと値のラベルのセットを定義できます。各キーは特定のオブジェクトに対して一意である必要があります。

ラベルメタデータのフォーマットと構文の例を次に示します。

```
"metadata": {"labels": {"key1": "value1", "key2": "value2"}}
```

ポッド名を具体的に指定するか、ポッドの目的のグループに該当するラベルを指定します。複数の注釈引数を使用する場合は、次に示す **JSON** アレイのように正しい **JSON** 形式を指定します: `["item1","item2","itemn"]# kubectl annotate pod [ {pod_name} | -l {label=value} ] -n {the-pods-namespace_name} [annotation syntax - see following]`

目的の結果を得るために一部のアプリケーションで複数のコマンドが必要な場合は、複数のコマンドを結合するために、このメソッドを `&&` と組み合わせることができます。指定したコマンドは **Cohesity** によって提供されないため、ユーザーは手動でアプリケーションポッドをカスタマイズする必要があります。`{values}` を環境で使用されている実際の名前に置き換えます。

メモ: すべての `kubectl` コマンドは 1 行で定義する必要があります。次の例をコピーまたは貼り付けるときは注意してください。

NetBackup 10.2 にアップグレードした後、これらの新しい `netbackup-pre` と `netbackup-post` のバックアップフックに対する注釈を更新します。これには、「`netbackup`」の接頭辞が含まれています。

```
netbackup-pre.hook.back.velero.io/command
netbackup-pre.hook.backup.velero.io/container
netbackup-post.hook.back.velero.io/command
netbackup-post.hook.backup.velero.io/container
```

ポッド名を使用した MongoDB の例

MongoDB 4.2.23 データベースをロックおよびロック解除するコマンドを次に示します。

```
# mongo --eval "db. fsyncLock ()"
# mongo --eval "db.fsyncUnlock ()"
```

これは、MongoDB のバックアップ前と後の両方のフックを設定する次の単一のコマンドに変換されます。特殊文字と、JSON 形式の一部として使用される角カッコ (`()`)、一重引用符、二重引用符、およびカンマ (`,`) をエスケープするための特殊な構文に注意してください。

```
# kubectl annotate pod {mongodb-pod-name} -n {mongodb namespace}
netbackup-pre.hook.back.velero.io/command='["/bin/bash", "-c", "mongo
--eval `db.fsyncLock()`"]'
netbackup-pre.hook.backup.velero.io/container={mongodb-pod-name}
netbackup-post.hook.backup.velero.io/command='["/bin/bash", "-c", "mongo
--eval `db.fsyncUnlock()`"]'
netbackup-post.hook.backup.velero.io/container={mongodb-pod-name}
```

ラベルを使用した MySQL の例

次に、MySQL データベースを静止および静止解除するコマンドを示します。

```
# mysql -uroot -ppassword -e "flush tables with read lock"
# mysql -uroot -ppassword -e "unlock tables"
```

これは、MySQL のバックアップ前と後の両方のフックを設定する次の単一のコマンドに変換されます。この例では、ポッド名の代わりにラベルを使用しており、このラベルは一度に複数のポッドに注釈を付けることができます。特殊文字と、JSON 形式の一部として使用される角カッコ (`()`)、一重引用符、二重引用符、およびカンマ (`,`) をエスケープするための特殊な構文に注意してください。

```
# kubectl annotate pod -l label=value -n {mysql namespace}
netbackup-pre.hook.backup.velero.io/command='["/bin/bash", "-c",
"mysql -uroot -ppassword -e ¥"flush tables with read lock¥"'
netbackup-pre.hook.backup.velero.io/container={mysql container name}
netbackup-post.hook.backup.velero.io/command='["/bin/bash", "-c",
"mysql -uroot -ppassword -e ¥"unlock tables¥"'
netbackup-post.hook.backup.velero.io/container={mysql container name}
```

ラベルを使用した Postgres の例

次に、PostgreSQL データベースを静止および静止解除するコマンドを示します。

```
# Psql -U postgres -c "SELECT pg_start_backup('tagvalue');"
# psql -U postgres -c ¥"SELECT pg_stop_backup();" "
```

これは、Postgres のバックアップ前と後の両方のフックを設定する次の単一のコマンドに変換されます。この例では、ポッド名の代わりにラベルを使用しており、ラベルは一度に複数の一致するポッドに注釈を付けることができます。任意の Kubernetes オブジェクトにラベルを適用できます。この場合は、特定のコンテナを変更し、特定のポッドのみを選択するための別の方法としてラベルを使用しています。特殊文字と、JSON 形式の一部として使用される角カッコ ({}), 一重引用符、二重引用符、およびカンマ (,) をエスケープするための特殊な構文に注意してください。

```
# kubectl annotate pod -l app=app-postgresql -n {postgres namespace}
netbackup-pre.hook.backup.velero.io/command='["/bin/bash", "-c",
"psql -U postgres -c ¥"SELECT
pg_start_backup(quote_literal($EPOCHSECONDS));¥"'
netbackup-pre.hook.backup.velero.io/container={postgres container
name} netbackup-post.hook.backup.velero.io/command='["/bin/bash",
"-c", "psql -U postgres -c ¥"SELECT pg_stop_backup();¥"'
netbackup-post.hook.backup.velero.io/container={postgres container
name}
```

コンテナフックなしの NGINX アプリケーションの例

次に、NGINX アプリケーションを静止および静止解除するコマンドを示します。

```
# /sbin/fsfreeze --freeze /var/log/nginx
# /sbin/fsfreeze --unfreeze /var/log/nginx
```

これは、NGINX のバックアップ前と後の両方のフックを設定する次の単一のコマンドに変換されます。この例では、コンテナフックを省略します。これにより、デフォルトでポッド名と一致する最初のコンテナが変更されます。特殊文字と、JSON 形式の一部として使用される角カッコ ({}), 一重引用符、二重引用符、およびカンマ (,) をエスケープするための特殊な構文に注意してください。

```
# kubectl annotate pod {nginx-pod-name} -n {nginx namespace}
netbackup-pre.hook.backup.velero.io/command='["/sbin/fsfreeze",
"--freeze", "/var/log/nginx"]'
netbackup-post.hook.backup.velero.io/command='["/sbin/fsfreeze",
"--unfreeze", "/var/log/nginx"]'
```

Cassandra の例

次に、**Cassandra** データベースを静止および静止解除するコマンドを示します。

```
# nodetool flush
# nodetool verify
```

これは、**Cassandra** のバックアップ前と後の両方のフックを設定する次の単一のコマンドに変換されます。特殊文字と、JSON 形式の一部として使用される角カッコ ([])、一重引用符 (')、二重引用符 ("")、およびカンマ (,) をエスケープするための特殊な構文に注意してください。

```
# kubectl annotate pod {cassandra-pod} -n {Cassandra namespace}
netbackup-pre.hook.backup.velero.io/command='["/bin/bash", "-c",
"nodetool flush"]'
netbackup-pre.hook.backup.velero.io/container={cassandra-pod}
netbackup-post.hook.backup.velero.io/command='["/bin/bash", "-c",
"nodetool verify"]'
netbackup-post.hook.backup.velero.io/container={cassandra-pod}
```

メモ: ここに示す例では、初期ガイドのみが提供されています。各作業負荷の特定の要件には、バックアップ、作業負荷、**Kubernetes** 管理者間のコラボレーションが含まれている必要があります。

現時点では、**Kubernetes** はエラー時のフックをサポートしていません。ユーザー指定のコマンドが失敗した場合、バックアップスナップショットは続行されません。

終了状態を返すコマンドのデフォルトのタイムアウト値は **30** 秒です。ただし、この値は、ポッドへの注釈として次のフックで変更できます。

```
netbackup-pre.hook.backup.velero.io/timeout=#in-seconds#
netbackup-post.hook.backup.velero.io/timeout=#in-seconds#
```

イメージグループの管理

この章では以下の項目について説明しています。

- [イメージグループについて](#)

イメージグループについて

各 **Kubernetes** リカバリポイントに対して、イメージグループが作成されます。イメージグループには、名前空間内の対象永続ボリューム要求の数に応じて、複数のイメージが含まれる場合があります。

メタデータに対して個別のイメージが作成され、永続ボリューム要求ごとに 1 つのイメージが作成されます。

リカバリポイントの詳細 **API** は、イメージグループのすべてのバックアップ ID、リソース名、コピー完了状態についての詳細を取得するために使用します。

Kubernetes 作業負荷に対するスナップショットからのバックアップ操作をサポートするために、単一の名前空間に対して複数のバックアップイメージが作成され、スナップショットからのバックアップが実行されます。

Kubernetes のバックアップ操作では、すべての永続ボリュームに対して個別のバックアップイメージが作成されます。特定の操作 (リストア、削除、インポートなど) を正常に実行するには、作成されたすべてのイメージをグループ化する必要があります。

イメージの期限切れ

期限切れのイメージが占有するストレージ領域を再利用するには、それらのイメージを削除する必要があります。

イメージの有効期限に関連した重要なポイントを次に示します。

複数のイメージで構成されるリカバリポイントの場合:

- イメージグループ内の 1 つのイメージを期限切れにしても、残りのイメージの有効期限が自動的に切れることはありません。イメージグループ内のすべてのイメージを明示的に期限切れにする必要があります。
- いくつかのイメージを期限切れにした場合、リカバリポイントは不完全になります。リカバリポイントが不完全な場合、リストア操作はサポートされません。
- いずれかのイメージの有効期限を変更した場合は、残りのイメージの有効期限を変更する必要があります。そうしないと、リカバリポイントに対応するイメージの有効期限にずれが生じ、ある時点でリカバリポイントが不完全になります。

イメージのインポート

Kubernetes のリカバリポイントは、複数のイメージで構成されている場合があります。リストア操作を実行するには、リカバリポイントに対応するすべてのイメージをインポートする必要があります。そうしないと、リカバリポイントは不完全とマークされ、リストアは実行されません。

詳しくは、『NetBackup™ 管理者ガイド Vol. 1』の「バックアップイメージのインポートについて」セクションを参照してください。

イメージのコピー

イメージコピーは、次の 2 種類のバックアップ操作で作成できます。

1. スナップショットはデフォルトのコピーで、コピー番号 1 としてマークされます。
2. スナップショットからのバックアップはコピー番号 2 としてマークされます。

任意の今すぐバックアップ操作またはスケジュールバックアップがトリガされるたびに、スナップショットが作成されます。ただし、[スナップショットからのバックアップ (Backup from snapshot)] は、保護計画の作成時に [スナップショットからのバックアップ (Backup from snapshot)] オプションが選択されているかどうかによって左右されるため、任意です。

イメージグループは、メタデータの資産のイメージや永続ボリューム要求 (PVC) で構成されます。各コピーには、名前空間用に 1 つのイメージ、名前空間に存在する各 PVC 用に 1 つのイメージが含まれます。

リカバリポイントの詳細 API を使用して、イメージのコピー完了状態を識別します。この API では、それぞれのコピーに存在するすべてのバックアップ ID とリソース名も詳細に示されます。不完全なイメージコピーから資産をリストアしようするとエラーが発生するため、イメージコピーが完全な状態か不完全な状態かはリストア機能に役立ちます。

不完全なイメージコピー

不完全なイメージの条件を次に示します。

1. スナップショットジョブまたはスナップショットからのバックアップジョブが進行中の場合、対応するコピーが不完全なコピーとして表示されます。
2. PVC のバックアップ処理が失敗すると、コピーは不完全としてマークされます。

3. コピーの子イメージ (複数の子を含む) が期限切れになると、コピーは不完全としてマークされます。

NetBackup でのランチャ管理クラスタの保護

この章では以下の項目について説明しています。

- 自動構成を使用した NetBackup へのランチャ管理 RKE クラスタの追加
- NetBackup でのランチャ管理 RKE クラスタの手動での追加

自動構成を使用した NetBackup へのランチャ管理 RKE クラスタの追加

次の手順に従って、自動構成を使用して NetBackup にランチャ管理 RKE クラスタを追加します。

自動構成を使用して NetBackup にランチャ管理 RKE クラスタを追加するには

メモ: グローバルランチャ管理サーバー証明書を抽出します。この CA 証明書は、管理サーバーの作成時に、ランチャによってデフォルトで生成される証明書、または別の/外部の CA (認証機関) を使用して構成される証明書である場合があります。

- 1 CA 証明書を抽出します。ランチャ管理サーバーの UI に移動して、左側のパネルの[グローバル設定 (Global Settings)]の[CA 証明書 (CA Certificate)]の下で、[CA 証明書の表示 (Show CA Certificate)]ボタンをクリックします。一時ファイルに CA 証明書の完全な値を抽出します。

メモ: 開始行と終了行を含む完全な値を抽出してください。

- 2 CA 証明書の値が、Kubernetes Operator の Helm インストールの前に作成されるシークレットに追加されます。

- 3 トークンを抽出するには、次のようにします。ランチャ管理サーバーの UI で左側のパネルを開き、[Explore Cluster] セクションで保護するクラスタに移動し、右上隅にある [Download KubeConfig] アイコンをクリックします。
- 4 アイコンを使用してクラスタの KubeConfig をダウンロードします。トークンフィールドがファイル内に存在します。
- 5 ダウンロードした Kubeconfig ファイルから、二重引用符を除いてトークンの値を抽出します。
- 6 この構成プロセスでは、シークレットの命名パターン (<kops-namespace>-nb-config-deploy-secret) が使用されます。
シークレットには、手順 1 と手順 3 で抽出した値があります。
- 7 次の形式で yaml ファイル nb-config-deploy-secret. yaml を作成し、すべてのフィールドに値を入力します。

```
apiVersion: v1
kind: secret
metadata:
  name: <kops-namespce>-nb-config-deploy-secret
  namespace: <kops-namespace>

type: Opaque
stringData:
  #All the 3 fields are mandatory here to add a Rancher managed
  RKF2 cluster in NetBackup
  apikey:
A_YoUkgYQwPLUkmyj9Q6A1-6RX8RNY-PtYX0SukbqCwIK-osPz8qVm9zCL9phje

  k8stoken:
kubecfg-user-mvvgcm8sq8:nrscvn8hj46t24r2tjrx2kn8tzo2bg4kj8waxpw36k8ktrchp826

  k8scacert: |
-----BEGIN CERTIFICATE-----
MIIDDDCCAfSgAwIBAgIBATANBgkqhkiG9w0BAQwIgYDVQQDDBtpbmdy
ZXNzLW9wZXJhdG9yQDE2ODc1MzY4NjgWHhcNMjMwNjIzMTYxNDI3WhcNMjUwNjIy
XtXqbaBGrXIuCCo90mxv4g==
-----END CERTIFICATE-----
```

- 8 コマンド `kubectl apply -f nb-config-deploy-secret. yaml` を実行します。
- 9 Helm Chart の `values.yaml` ファイルの残りの入力については、『Kubernetes クイックスタートガイド』の自動構成に関するセクションを参照し、設定の完了に必要なすべての値を入力します。

- 10 必要なすべての簡易インストールの入力が `values.yaml` ファイルに追加されたら、NetBackup Kubernetes Operator グラフで Helm のインストールコマンドを実行します。自動化された構成ポッド `<kops-namespace>-netbackup-config-deploy` が起動するはずです。
- 11 `<kops-namespace>-netbackup-config-deploy` ログを確認して、更新されたシークレット値が `config-deploy` ポッドによって選択されたかどうかを確認します。
- 12 `config-deploy` ポッドがそのタスクを実行すると、クラスタが NetBackup に正常に追加され、検出要求が進行中になるか、正常に完了します。NetBackup Web UI から別のクレデンシャルの検証と手動検出を実行して、プロセスが正常に動作していることを確認します。

NetBackup でのランチャ管理 RKE クラスタの手動での追加

次の手順に従って、NetBackup でランチャ管理 RKE クラスタを手動で追加します。

NetBackup 用の Kubernetes クレデンシャルの作成

NetBackup Web UI で[クレデンシャルの管理 (Credential Management)]、[指定したクレデンシャル (Named Credentials)]、[追加 (Add)]、[クレデンシャルを追加 (Add credentials)]の順に移動し、クレデンシャルストアを NetBackup として選択し、[カテゴリ (Category)]フィールドで Kubernetes を選択し、以前に Global Rancher Management プラットフォーム UI から抽出したトークンと CA 証明書を入力し、このクレデンシャルを保存します。

NetBackup でランチャ管理 RKE クラスタを手動で追加するには

- 1 外部 CA 証明書: 外部アクセス用の証明書の構成に使用されている CA (認証機関) が異なる場合、NetBackup がクラスタと正常に通信するために外部 CA 証明書が必要です。
 - ランチャ管理サーバーの UI に移動して、左側のパネルの[Global Settings]の[cacerts]の下で、[showcacerts]ボタンをクリックします。
一時ファイルにこの CA 証明書の完全な値を抽出します。
 - たとえば、`<cacert-value-file>`
- 2 サービスアカウントの CA 証明書:

メモ: クラスタ内で利用可能なサービスアカウントの CA 証明書と比較して、Kubernetes API サーバーの外部アクセス用に構成されている CA (認証機関) が異なるため、次の手順を実行する必要があります。そのため、これらの 2 つの CA 証明書を組み合わせる必要があります。

サービスアカウントの CA 証明書を取得するには、Linux クラスタホストで次のコマンドを実行します。

- 次のコマンドを使用して、Kubernetes Operator の名前空間で利用可能なサービスアカウントシークレット名を取得します。

```
kubectl describe serviceaccount <kopsnamespace>-backup-server
-n <kopsnamespace> | grep Tokens | cut -d ":" -f 2
```

- 次のコマンドを使用して、このサービスアカウントのシークレットから **base 64** デコードされた形式で CA 証明書を取得します。

```
kubectl get secret <output-from-previous-command> -n
<kopsnamespace> -o jsonpath='{. data.ca.crt}' | base64 -d
```

このコマンドの出力全体を、手順 1 で作成した一時ファイルに追加する必要があります。

- 3 手順 2 の後に生成された出力を <cacert-value-file> ファイルの最後に追加します。必要な外部 CA 証明書と内部 CA 証明書の値が抽出され、ファイル <cacert-value-file> で利用可能になります。CA 証明書の値は **base 64** でデコードされた形式ですが、NetBackup でクレデンシヤルを作成するときに再度エンコードする必要があります。

- 4 トークン: ランチャ管理サーバーの UI で左側のパネルを開き、[Explore Cluster] セクションで保護するクラスタに移動し、右上隅にある [Kubeconfig] アイコンをクリックします。

- ([Download KubeConfig]を使用して) ダウンロードした Kubeconfig ファイルから、一時ファイル <token-value-file> に二重引用符を除いたトークンの値を抽出します。
- これらのトークンと cacert の両方のフィールドは、Kubernetes の NetBackup クレデンシヤルに追加するために、**base64** エンコード形式であることが必要です。
- 次の **base64** コマンドを使用して、これらの抽出された両方の値の **base64** エンコードバージョンを取得する方法を示します。

#Linux VM を使用して、この手順の値をエンコードします #注意: フラグ -w0 には 0 の記号ではなく数字のゼロを使用します。

#CA 証明書:

```
Cat <cacert-value-file>| base64 -w0
```

この出力を、NetBackup のクレデンシヤル作成ページの [CA 証明書 (CA certificate)] フィールドに貼り付けます。

#トークン:

この出力を、NetBackup Web UI のクレデンシヤル作成ページの [トークン (Token)] フィールドに貼り付けます。

- これらの値を、NetBackup Web UI の[クレデンシヤルの管理 (Credential management)]、[指定したクレデンシヤル (Named Credentials)]、[追加 (Add)]で使用して、NetBackup に有効なランチャクレデンシヤルを追加します。
- クレデンシヤルが作成されたら、次のクラスタ情報の出力に示されている名前を使用して、NetBackup に Kubernetes クラスタを追加します。

クラスタ情報の出力を取得するには、次のコマンドを実行します。

- 1 クラスタ情報の出力は、次の例の形式である必要があります。[root@master-0~]

```
# kubectl cluster-info
```
- 2 Kubernetes コントロールプレーンは、
<https://<rancher-hostname>/k8s/clusters/c-m-zjrrft56> で実行されます
- 3 CoreDNS は、<https://<rancher-hostname>/k8s/clusters/c-m-zjrrft56/api/v1/> で
 実行されます

```
namespaces/kube-system/services/rke2-coredns-rke2-coredns:udp-53/proxy
```
- 4 出力から、API サーバーのエンドポイント全体 (<https://> を含む) を抽出します。形式
 は、<https://<rancher-hostname>/k8s/clusters/c-m-zjrrft56> のパターンになるはず
 です
- 5 NetBackup Web UI の[作業負荷 (Workloads)]、[Kubernetes]、[Kubernetes
 クラスタ (Kubernetes clusters)]、[追加 (Add)]で、ランチャクラスタ名全体を追加
 します。
- 6 [Kubernetes クラスタの追加 (Add Kubernetes cluster)]ページで、URL またはエ
 ンドポイントに関連付けられているオプションを選択して、(<https://>) を含むエンドポ
 イントに基づいてクラスタを追加できるようにします。

メモ: エンドポイントベースのアプローチを使用して追加したクラスタ名は編集できま
 せん。そのようなクラスタ名は、削除および再追加だけを行います。

- 7 NetBackup Web UI (エンドポイントまたは URL) の入力フィールドに、上記で抽出
 したクラスタ情報の出力を入力します。
- 8 先に進み、手順 1 から 4 で準備したクレデンシヤルを選択または作成します。
- 9 クレデンシヤルが検証されると、クラスタが正常に追加されます。自動検証と検出が
 トリガされます。
- 10 自動検出が正常に実行されたら、ユーザーは手動のクレデンシヤル検証と検出を試
 み、すべてが正常に動作していることを確認します。

- 11 NetBackup でランチャ管理クラスタを追加します。
- 12 BFS (スナップショットからのバックアップ) 機能を設定するバックアップサーバー証明書シークレットとデータムーバー `configmap` を作成します。

次に、推奨されるセットアップガイドに従って、残りの構成手順に進みます。

Kubernetes 資産のリカバリ

この章では以下の項目について説明しています。

- [リカバリポイントの検索と検証](#)
- [スナップショットからのリストア](#)
- [バックアップコピーからのリストア](#)
- [スナップショットまたはバックアップからの仮想マシンのリストア](#)

リカバリポイントの検索と検証

NetBackup バージョン 10.0 以降では、スナップショットからのリストア操作とバックアップコピーからのリストア操作による **Kubernetes** 資産のリカバリをサポートしています。

メモ: リカバリ後、新しく作成された名前空間、永続ボリューム、その他のリソースには、新しいシステム生成 UID が割り当てられます。

NetBackup は、**Kubernetes** 作業負荷のバックアップコピーの完了状態または未完了状態を通じてバックアップイメージの検証を実行するのに役立ちます。NetBackup では、未完了のバックアップコピーからリストア操作を実行できません。

Kubernetes 名前空間に対応するリカバリポイントは、複数のイメージで構成されます。一部のイメージのコピーが利用できない可能性があるため、リカバリポイントは未完了である可能性があります。このようなリカバリポイントは未完了としてマークされます。

リカバリポイントの検証を実行するには

- 1 左側の[作業負荷 (Workloads)]で、[Kubernetes]をクリックします。
- 2 [名前空間 (Namespaces)]タブで、リカバリする資産の名前空間をクリックします。

- 3 [リカバリポイント (Recovery points)] タブをクリックします。
- 4 [リカバリポイント (Recovery points)] タブには、すべてのリカバリポイントがバックアップの日時およびコピーとともに表示されます。

リカバリポイントの横にあるコピー数のボタンをクリックすると、場所、デフォルトのコピー、コピーの種類、完了状態が表示されます。

完了状態は、リストア操作を実行するために選択したリカバリポイントを検証するのに役立ちます。

未完了のバックアップコピー、進行中のバックアップ、イメージの期限切れ、ハードウェア障害、またはネットワーク通信の問題には、複数の理由が考えられます。

スナップショットからのリストア

NetBackup は、単一のリストアジョブを使用して、リカバリポイントでのすべてのバックアップイメージをリストアできる、スナップショットからのリストア機能を備えています。アクティビティモニターにスナップショットジョブからのリストアを表示できます。

スナップショットからリストアするには

- 1 左側で[作業負荷 (Workloads)]、[Kubernetes]の順に選択します。
- 2 [名前空間 (Namespace)] タブで、リカバリする資産の名前空間をクリックします。
- 3 [リカバリポイント (Recovery points)] タブをクリックします。
[リカバリポイント (Recovery points)] タブには、すべてのリカバリポイントがバックアップの日時およびコピーとともに表示されます。フィルタを設定して、表示されたリカバリポイントをフィルタ処理できます。[日付 (Date)] 列の日付をクリックすると、リカバリポイントの詳細が表示されます。[リカバリポイントの詳細 (Recovery points details)] ダイアログには、構成マップ、Secret、永続ボリューム、ポッドなど、バックアップされたリソースが表示されます。これらのリソースについて詳しくは、<https://kubernetes.io/docs/reference/kubernetes-api/> を参照してください。
- 4 リストアするリカバリポイントを見つけます。
- 5 [コピー (Copies)] 列で、[# コピー (# copies)] ボタンをクリックします。たとえば、コピーが 2 つある場合、ボタンには[2 コピー (2 copies)] と表示されます。
- 6 コピーのリストで、スナップショットコピーを見つけます。[処理 (Actions)]、[名前空間のリストア (Restore namespace)] の順にクリックします。

メモ: [マルウェアに感染したリカバリポイントの選択を許可 (Allow the selection of recovery points that are malware-affected)] オプションを選択して、感染したすべてのコピーの[リストア (Restore)] オプションを有効にします。

- 7 [リカバリターゲット (Recovery target)] ページに、ターゲットクラスタが自動入力されます。

メモ: 代替クラスタのリストアは、スナップショットコピーではサポートされません。

- 8 [宛先名前空間を指定 (Specify destination namespace)] で、次のいずれかのリストアオプションを選択します。
- バックアップされた元の名前空間をリストアに使用の場合は[元の名前空間を使用 (Use original namespace)]。デフォルトでは、このオプションが選択されています。
 - リストアに代替名前空間を使用の場合は[代替名前空間を使用 (Use alternate namespace)]。その後、[次へ (Next)] をクリックします。
- 9 [リカバリするリソース形式の選択 (Select resource types to recover)] で、次のいずれかのリストアするリソース形式を選択します。
- すべてのリソース形式をリカバリする場合は[すべてのリソース形式 (All resource types)]。デフォルトでは、このオプションが選択されています。
 - 選択したリソース形式のみをリカバリする場合は[選択されたリソース形式のリカバリ (Recover selected resource types)]。

メモ: [リカバリするリソース形式の選択 (Select resource types to recover)] オプションは、上級ユーザー向けです。リストアするリソースの選択に注意しないと、リストア後に完全に機能する名前空間が得られない場合があります。

- 10 [リカバリする永続ボリューム要求の選択 (Select Persistent volume claims to recover)] で、次のいずれかのリカバリする永続ボリューム要求を選択します。
- すべての永続ボリューム要求をリカバリする場合は[すべての永続ボリューム要求 (All Persistent volume claims)]。デフォルトでは、このオプションが選択されています。
 - 選択した永続ボリューム要求をリカバリする場合は[選択された永続ボリューム要求のリカバリ (Recover selected Persistent volume claims)]。

メモ: [選択されたリソース形式のリカバリ (Recover selected resource types)] でオプションを選択せずに空の永続ボリューム要求を含めるオプションが選択されている場合、永続ボリューム要求はリストアされません。

[選択された永続ボリューム要求のリカバリ (Recover selected persistent volume claims)] でオプションを選択しない場合、[リカバリオプション (Recovery options)] セクションで永続ボリューム要求は空になり、永続ボリューム要求はリストアされません。

メモ: [永続ボリュームのみリストア (Restore only persistent volume)] を使用すると、選択した永続ボリューム要求で、永続ボリュームのみをリストアするよう切り替えることができます。この設定により、対応する永続ボリューム要求が作成されることはありません。

- 11 [エラー戦略 (Failure strategy)] セクションをクリックして、リカバリのためのエラー戦略オプションを表示します。
- 12 [リカバリのためのエラー戦略を選択 (Select failure strategy to recover)] で、次のリカバリのためのエラー戦略のいずれかを選択します。

メモ: メタデータまたは PVC のリストア中にエラーが発生すると、選択したエラー戦略に従ってリストアジョブが実行されます。

- [即座に終了 (Fail Fast)] を選択すると、エラーが発生した場合にリストアを終了します。
- [先に進む (Proceed Ahead)] を選択すると、次の PVC のリストアを続行します。親イメージ (最初のイメージ) のリストアが失敗した場合、リストアジョブは終了します。
- [再試行 (Retry)] では、メタデータまたは PVC リストアの再試行回数を指定します。再試行後もリストアが失敗した場合、リストアジョブは終了します。

メモ: 選択したエラー戦略がアクティビティモニターに表示されます。

- 13 [次へ (Next)] をクリックします。
- 14 [リカバリオプション (Recovery options)] ページで、[リカバリの開始 (Start recovery)] をクリックしてリカバリのエントリを送信します。
- 15 [アクティビティモニター (Activity monitor)] で、[ジョブ ID (Job ID)] をクリックし、リストアジョブの詳細を表示します。

メモ: NetBackup Kubernetes のリストアでは、単一ジョブを使用してすべての永続ボリューム要求と 1 つの名前空間をリストアします。[アクティビティモニター (Activity monitor)] でログを表示して、永続ボリューム、永続ボリューム要求、またはメタデータのリストアを追跡できます。

バックアップコピーからのリストア

選択した名前空間に複数の PVC が存在する場合、バックアップからの NetBackup リストアも並行して行われます。リストアを開始すると、ジョブは親子階層を作成します (名前空間にリストア対象の PVC が少なくとも 1 つ含まれている場合)。親ジョブは、オーケストレータとして機能し、子ジョブの状態を監視します。最初の子ジョブはメタデータをリストアし、その後 PVC が並行してリストアされます。

メモ: メタデータのリストアが失敗した場合、これ以上のジョブはリストア操作のために送信されません。メタデータが正常にリストアされると、PVC はバッチで並列してリストアされます。

スナップショットからのリストアで説明したのと同じ手順に従い、コピー形式として [バックアップ (Backup)] を選択します。代替ターゲットクラスタにもリストアできます。

バックアップコピーからリストアするには

- 1 左側で [作業負荷 (Workloads)]、[Kubernetes] の順に選択します。
- 2 [名前空間 (Namespace)] タブで、リカバリする資産の名前空間をクリックします。[リカバリポイント (Recovery points)] タブをクリックします。
- 3 [リカバリポイント (Recovery points)] タブには、すべてのリカバリポイントがバックアップの日時およびコピーとともに表示されます。フィルタを設定して、表示されたリカバリポイントをフィルタ処理できます。[日付 (Date)] 列の日付をクリックすると、リカバリポイントの詳細が表示されます。[リカバリポイントの詳細 (Recovery points details)] ダイアログには、ConfigMap、Secret、永続ボリューム、ポッドなど、バックアップされたリソースが表示されます。これらのリソースについて詳しくは、<https://kubernetes.io/docs/reference> を参照してください。
- 4 リストアするリカバリポイントを見つけます。
- 5 [コピー (Copies)] 列で、[# コピー (# copies)] ボタンをクリックします。たとえば、コピーが 2 つある場合、ボタンには [2 コピー (2 copies)] と表示されます。

- 6 コピーのリストで、[バックアップ (Backup)] コピーを見つけます。[処理 (Actions)]、[名前空間のリストア (Restore namespace)] の順にクリックします。

メモ: [マルウェアに感染したリカバリポイントの選択を許可 (Allow the selection of recovery points that are malware-affected)] オプションを選択して、感染したすべてのコピーの [リストア (Restore)] オプションを有効にします。

- 7 [リカバリターゲット (Recovery target)] ページで、資産を同じソースクラスタにリカバリすることが自動的に入力されます。[次へ (Next)] をクリックします。
- 8 [宛先名前空間を指定 (Specify destination namespace)] で、以下のオプションから選択します。
- [元の名前空間を使用 (Use original namespace)] を選択して、元の名前空間を使用します。
 - [代替名前空間を使用 (Use alternate namespace)] を選択して、代替名前空間を入力します。
[次へ (Next)] をクリックします。
- 9 [リカバリするリソース形式の選択 (Select resource types to recover)] で、次のリソース形式から選択します。
- すべてのリソース形式をリカバリする場合は、[すべてのリソース形式 (All resource types)] を選択します。
 - 選択したリソース形式のみをリカバリする場合は、[選択されたリソース形式のリカバリ (Recover selected resource types)] を選択します。
- 10 [リカバリする永続ボリューム要求の選択 (Select Persistent volume claims to recover)] で、次のオプションから選択します。
- すべての永続ボリューム要求をリカバリする場合は、[すべての永続ボリューム要求 (All Persistent volume claims)] を選択します。
 - 選択した永続ボリューム要求をリカバリする場合は、[選択された永続ボリューム要求のリカバリ (Recover selected Persistent volume claims)] を選択します。

メモ: [選択されたリソース形式のリカバリ (Recover selected resource types)] でオプションを選択せずに空の永続ボリューム要求を含めるオプションが選択されている場合、永続ボリューム要求はリストアされません。

[選択された永続ボリューム要求のリカバリ (Recover selected persistent volume claims)] でオプションを選択しない場合、[リカバリオプション (Recovery options)] セクションで永続ボリューム要求は空になり、永続ボリューム要求はリストアされません。

メモ: [永続ボリュームのみリストア (Restore only persistent volume)]を使用すると、選択した永続ボリューム要求で、永続ボリュームのみをリストアするよう切り替えることができます。この設定により、対応する永続ボリューム要求が作成されることはありません。

- 11 [エラー戦略 (Failure strategy)]セクションをクリックして、リカバリのためのエラー戦略オプションを表示します。
- 12 [リカバリのためのエラー戦略を選択 (Select failure strategy to recover)]で、次のリカバリのためのエラー戦略のいずれかを選択します。

メモ: メタデータまたは PVC のリストア中にエラーが発生すると、選択したエラー戦略に従ってリストアジョブが実行されます。

- [即座に終了 (Fail Fast)]を選択すると、エラーが発生した場合にリストアを終了します。
 - このリストアエラー戦略は、最初のエラーが発生したときにリストアジョブを終了するのに役立ちます。
 - 現在のバッチの残りのすべての実行中のリストアジョブは完了でき、それ以降のバッチはリストアのために送信されません。
- [先に進む (Proceed Ahead)]を選択すると、次の PVC のリストアを続行します。親イメージ (最初のイメージ) のリストアが失敗した場合、リストアジョブは終了します。
 - この戦略は、進行中のバッチで PVC リストアのいずれかが失敗した場合に、残りの PVC のリストアを進めるのに役立ちます。
 - メタデータリストアが失敗すると、最終的なジョブは失敗としてマークされ、リストア用の PVC は送信されません。
 - この場合、部分的な成功とマーク付けされた最終的なジョブ状態と、失敗状態の PVC のリストが親ジョブの [アクティビティモニター (Activity Monitor)] タブに表示されます。
- [再試行 (Retry)]では、メタデータまたは PVC リストアの再試行回数を指定します。再試行後もリストアが失敗した場合、リストアジョブは終了します。
 - このエラー戦略は、リストアジョブの開始時に構成可能な、失敗した PVC またはメタデータのリストアジョブを再試行するのに役立ちます。
 - 再試行の最大数にかかわらずリストアジョブが失敗した場合、失敗とマーク付けされたジョブは失敗し、それ以降のバッチはリストアのために送信されません。

メモ: 選択したエラー戦略がアクティビティモニターに表示されます。

Windows では、[即座に終了 (Fail Fast)] オプションを使用したリストアジョブは失敗する場合があります。p.142 の「エラー「2890 K8s のデータのリストア操作が失敗しました (2890 K8s restore data operation failed)」が表示され、即座に終了リストア戦略が失敗しました。」を参照してください。

- [次へ (Next)] をクリックします。
- 13** [リカバリの開始 (Start recovery)] をクリックしてリカバリのエントリを送信します。
- 14** [アクティビティモニター (Activity monitor)] で、[ジョブ ID (Job ID)] をクリックし、リストアジョブの詳細を表示します。
- 15** [ジョブの詳細 (Job Details)] ページで、[詳細 (Details)] タブをクリックします。リストアジョブのシーケンス(リストア前、データの移動、リストア後のジョブ)が表示されます。

メモ: 親ジョブを取り消して、リストア操作を取り消すことができます。親ジョブによって、実行中のすべての子リストアジョブが終了します。

構成の変更

並列 PVC リストアのバッチサイズは `bp.conf` で構成できます。目的のバッチサイズを設定するために `bp.conf` ファイルにキー

`KUBERNETES_RESTORE_FROM_BACKUP_COPY_PARALLEL_RESTORE_BATCH_SIZE` を追加できます。これはオプションのキーであり、定義されていない場合は値 `5` になります。

バッチサイズに割り当てることができる最小値は `1` であるのに対し、最大値は `100` です。

`bpsetconfig` コマンドを `NetBackup` プライマリサーバーで使用して、バッチサイズを更新できます。

スナップショットまたはバックアップからの仮想マシンのリストア

`NetBackup` には、スナップショット機能と特定の名前空間のバックアップから仮想マシンをリストアする機能が搭載されています。実行されたバックアップとスナップショットは、[リカバリポイント (Recovery points)] タブに表示されます。

メモ: この機能は、この 11.1 リリースで作成されたバックアップに適用されます。古いリリースで作成されたバックアップの場合は、[名前空間のリストア (Restore namespace)] オプションを選択します。次の手順の手順 4 を参照してください。

警告: KubeVirt に対応した OpenShift クラスタからバニラ Kubernetes クラスタ (KubeVirt なし) に仮想マシンをリストアすると、PVC のみがリストアされます。ターゲットクラスタに KubeVirt が存在しないため、VM オブジェクトは表示されないか、機能しません。

KubeVirt に対応したソースから VM の作業負荷をリストアする前に、ターゲットの Kubernetes クラスタに KubeVirt がインストールされ構成されていることを確認してください。KubeVirt がいない場合、PVC のみがリストアされ、VM 定義は無視されるか、配備に失敗します。

仮想マシンをリストアするには

メモ: プライマリサーバーと NetBackup Kubernetes Operator が NetBackup 11.1 以降を実行している場合でも、NetBackup 11.1 より以前のバージョンのメディアサーバーを使用して作成されたバックアップでは、単一の VM のリストアはサポートされません。このようなバージョンが混在する環境では、単一 VM リストアの試行が状態 2892 で失敗する場合があります。名前空間全体のリストアがサポートされており、想定どおりに機能します。

- 1 左側で[作業負荷 (Workloads)]、[Kubernetes]の順にクリックします。
- 2 [名前空間 (Namespace)]タブで、リカバリする資産の名前空間をクリックします。
- 3 [リカバリポイント (Recovery points)]タブをクリックします。
[リカバリポイント (Recovery points)]タブには、すべてのリカバリポイントがバックアップの日時およびコピーとともに表示されます。フィルタを設定して、表示されたリカバリポイントをフィルタ処理できます。[日付 (Date)]列の日付をクリックすると、リカバリポイントの詳細が表示されます。[リカバリポイントの詳細 (Recovery points details)]ダイアログには、構成マップ、Secret、永続ボリューム、ポッドなど、バックアップされたリソースが表示されます。
- 4 [コピー (Copies)]列で、[# コピー (# copies)]ボタンをクリックします。たとえば、コピーが 2 つある場合、ボタンには[2 コピー (2 copies)]と表示されます。
- 5 仮想マシンをリストアするリカバリポイントを見つけます。ケバブメニューをクリックし、[仮想マシンのリストア (Restore virtual machine)]を選択します。
- 6 [仮想マシンのリストア (Restore virtual machine)]ウィザードが表示されます。
- 7 [リカバリターゲット (Recovery target)]の最初のタブに、ターゲットクラスタが自動入力されます。[次へ (Next)]をクリックします。
- 8 [リカバリオプション (Recovery options)]の手順が表示されます。[すべての仮想マシン (All virtual machines)]または[特定の仮想マシン (Specific virtual machine)]のいずれかのオプションを選択します。

- 9 VM の起動時に新しい MAC アドレスを生成できるように、古い割り当て済みの MAC アドレスを削除する場合は、[リストア中に仮想マシンの MAC アドレスを消去 (Clear virtual machine MAC address during restore)] オプションを選択します。
- 10 [リカバリのためのエラー戦略 (Failure strategy to recover)] から少なくとも 1 つのオプションを選択します。選択できるオプションは次のとおりです。

メモ: メタデータまたは PVC のリストア中にエラーが発生すると、選択したエラー戦略に従ってリストアジョブが実行されます。ただし、選択した仮想マシンの依存リソースは自動的にリカバリされます。

- [即座に終了 (Fail fast)] を選択すると、エラーが発生した場合にリストアを終了します。(デフォルト)
- [先に進む (Proceed ahead)] を選択すると、次の PVC のリストアを続行します。親イメージ (最初のイメージ) のリストアが失敗した場合、リストアジョブは終了します。
- [再試行 (Retry)] を選択すると、メタデータまたは PVC リストアの再試行回数を指定できます。再試行後もリストアが失敗した場合、リストアジョブは終了します。最大 5 回の再試行を選択できます。デフォルトは 3 回です。

メモ: 選択したエラー戦略がアクティビティモニターに表示されます。

- 11 [次へ (Next)] をクリックします。
- 12 NetBackup に選択または指定した値の概略情報が表示されます。[リカバリの開始 (Start recovery)] をクリックします。
- 13 [リカバリの開始 (Start recovery)] をクリックすると、NetBackup に [リストアアクティビティ (Restore activity)] タブが表示され、開始されたジョブアクティビティが表示されます。

メモ: [アクティビティモニター (Activity monitor)] でリカバリアクティビティを表示することもできます。

増分バックアップとリストアについて

この章では以下の項目について説明しています。

- [Kubernetes](#) の増分バックアップとリストアのサポート

Kubernetes の増分バックアップとリストアのサポート

NetBackup Kubernetes バージョン 10.4 以降では、差分、累積、自動のスケジュール向けのバックアップサポートが提供されます。

増分バックアップでは、NetBackup のバックアップ処理時間が大幅に短縮されます。この方式で、NetBackup は最後の完全バックアップ以降に変更されたデータだけをバックアップします。

増分バックアップのサポート

増分バックアップは、ファイルシステムタイプの永続ボリュームのみをサポートします。ブロックタイプの永続ボリュームバックアップは、スケジュール形式に関係なく常に完全バックアップです。

メモ: スナップショットコピーは、ストレージクラスの制限により常に完全バックアップです。スナップショットコピー以外に、スナップショットからのバックアップ、複製コピーでは増分がサポートされません。

リストアジョブ

完全なリカバリポイントからリストアすると、指定した時点へのリストアが実行されます。そのリカバリポイントがリストアされるまでのすべてのデータ。

[完了 (Complete)]フィールドに[いいえ (No)]と表示されている場合は、そのリカバリポイントから復元できません。

イメージチェーンの検証

イメージチェーンの検証操作はリカバリポイントコピーに対して実行され、検証は各バックアップコピーのリカバリポイントの[Complete]フィールドに反映されます。

リカバリポイントのすべての関連イメージが存在する場合、[Complete]フィールドは[Yes]に設定されます。

メモ: 増分バックアップチェーンが不完全な場合、またはイメージグループにイメージが欠けている場合、[Complete]フィールドは[No]([Complete]=[No])としてマーク付けされます。

手動インポートからのリストア

手動でインポートした増分イメージは、有効なリカバリポイント([Complete]=[Yes])からリストアできます。

手動インポートのトラブルシューティング

手動インポートの後にリカバリポイントが[未完了 (Incomplete)]とマークされている場合は、手動インポートの操作中にいくつかのイメージが失われたことが原因で、イメージチェーンが破損している可能性があります。

手動インポート操作でイメージチェーンを再作成するには

- 1 ファイル /usr/openv/netbackup/logs/bpdbm/root{dateformat}.log を開き、以前のバックアップ関係についての行を見つけます。関係をリストアするには、どのイメージが手動インポート操作から抜け落ちているかを把握します。
- 2 抜け落ちたイメージをインポートし、次のコマンドを実行して新しいイメージチェーンを作成します。

```
`bpimage -update -id <backupid> -previous_backupid <previous  
backup id>`
```

ctime フラグと mtime フラグ

NetBackup クライアントの USE_CTIME_FOR_INCREMENTALS オプション:

- USE_CTIME_FOR_INCREMENTALS エントリは、ファイルが変更されているかどうかを NetBackup が判断する方法を変更します。増分バックアップの実行中、クライアントソフトウェアで変更時刻および i ノード変更時刻の両方 (mtime および ctime) を使用して、ファイルが変更されたかどうかを判断します。

NetBackup クライアントの DO_NOT_RESET_FILE_ACCESS_TIME オプション:

- DO_NOT_RESET_FILE_ACCESS_TIME エントリは、ファイルのバックアップが行われたとき、ファイルのアクセス時刻 (Atime) にバックアップ時刻が表示されるように指定します。デフォルトでは、NetBackup はアクセス時刻を保持します。NetBackup はバックアップの以前の値をリセットします。

- データムーバーのプロパティを設定するには、Kubernetes クラスターの NetBackupKOps 名前空間に作成された、NetBackup プライマリサーバー固有の ConfigMap でフラグを設定または更新する必要があります。

- 例:

```
apiVersion: v1
data:
  datamover.properties: |
    image=reg.domain.com/datamover/image:latest
    VERBOSE=5
    VXMS_VERBOSE=5
    USE_CTIME_FOR_INCREMENTALS=YES
    DO_NOT_RESET_FILE_ACCESS_TIME=YES
    version: "1"
kind: ConfigMap
metadata:
  name: backupserver.sample.domain.com
  namespace: <NetBackupKOps-Namespace>
```

保護計画

NetBackup Kubernetes 作業負荷では、次のスケジュールがサポートされます。

- 自動
- 完全
- 差分増分
- 累積増分

異なるスケジュールが含まれる保護計画は、次のように構成できます。

異なるスケジュール形式でバックアップするには

- 1 保護計画で、バックアップ形式 (完全、差分増分、累積増分、自動) を選択します。
- 2 スナップショットのスケジュールの構成で、繰り返しとスナップショット保持の値を指定します。

[スナップショットからのバックアップ (Backup from snapshot)]の保持期間の値を指定します。

- 3 [開始時間帯 (Start window)]タブでスケジュール時刻を設定し、[追加 (Add)]をクリックします。

- 4 [スケジュールと保持 (Schedules and retention)] セクションで、[スケジュールの追加 (Add schedule)] をクリックして、同じ保護計画 (差分増分、累積増分) に複数のスケジュールを追加します。
- 5 ストレージを選択し、残りの手順に従ってバックアップジョブを実行します。

自動スケジュール

- 自動形式のスケジュールの場合、スナップショットの繰り返しに基づいて、スケジュールは保護計画の作成後に解決されます。
- 繰り返しが 1 週間未満の場合は、1 つの差分スケジュールと 1 つの完全スケジュールが作成されます。

推奨事項

- 増分スケジュールの保護計画の[保持 (Retention)]の値に関する推奨事項に従います。
 - 任意のコピー (スナップショット、スナップショットからのバックアップ、複製) のリカバリポイントからリストアを実行するには、コピーの保持期間をより長い期間にすることを勧めます。
 - たとえば、バックアップコピーからリストアする場合、[スナップショットからのバックアップ (Backup from Snapshot)] の保持期間は[スナップショットコピー (Snapshot copy)] より長くする必要があります。そうしないと、バックアップコピーは期限切れになり、リカバリポイントは[Complete]=[No]としてマーク付けされます。
 - このような場合、NetBackup Web UI に次のように警告が表示されます。
 - バックアップの保持期間は、スナップショットの保持期間より長く設定することをお勧めします。
 - 複製の保持期間は、バックアップの保持期間より長く設定することをお勧めします。
- 完全バックアップスケジュールは、常に累積バックアップスケジュールとともに追加します。そうしないと、すべての累積バックアップは完全バックアップとして実行されます。
- デフォルトでは、[スナップショットからのバックアップ (Backup from Snapshot)] オプションは常に増分バックアップ形式に対して有効になっています。

アクセラレータベースのバックアップの有効化

この章では以下の項目について説明しています。

- [Kubernetes](#) 作業負荷に対する [NetBackup](#) アクセラレータのサポートについて
- プライマリサーバーにあるトラックログのディスク容量の制御
- ストレージクラスの動作がアクセラレータに与える影響
- アクセラレータ強制再スキャンについて
- アクセラレータバックアップのエラーに関する警告と考えられる理由

Kubernetes 作業負荷に対する NetBackup アクセラレータのサポートについて

NetBackup アクセラレータは、Kubernetes クラスタのバックアップにかかるバックアップ時間を減らします。

Kubernetes バックアップの場合、アクセラレータ機能はアクセラレータをサポートするストレージ形式を選択すると有効になります。たとえば、アクセラレータが有効なバックアップをサポートする MSDP、OpenStorage、CloudStorage、MSDP-C (Azure および AWS)、Kubernetes クラスタなどです。

メモ: アクセラレータが有効なバックアップは、ファイルモードの PVC でのみサポートされます。

特定の Kubernetes クラスタに対するアクセラレータサポートの有効化

NetBackup Kubernetes Operator の `values.yaml` には、
`acceleratorTracklogPvcStorageClass: None` というエントリがあります。

アクセラレータを有効にするには、アクセラレータのバックアップのトラックログを生成するため、有効なストレージクラス名を指定します。ストレージクラスは、Kubernetes クラスタ内のワーカーノードで使用できるファイルモードボリュームを作成するのに役立ちます。

メモ: `acceleratorTracklogPvcStorageClass` が `None` に設定され、アクセラレータが有効なストレージが選択されている場合、アクセラレータのバックアップジョブは実行されません。NetBackup 10.4 リリースにアップグレードした後の `acceleratorTracklogPvcStorageClass` のデフォルト値は `None` です。

詳しくは、『NetBackup for Kubernetes 管理者ガイド』の「アクセラレータストレージクラスの検証」セクションを参照してください。

アクセラレータのトラックログストレージクラスに関するリソースのスロットルとストレージの要件

- Kubernetes クラスタあたりのスナップショットからのバックアップジョブ数のデフォルト値は 4 です。
- アクセラレータを伴う 4 つのスナップショットからのバックアップジョブを実行して、4 つの PVC を同時にバックアップすると、相当のストレージが消費されます。
- この計算によると、各 PVC には、トラックログを作成するためのある程度の容量が必要です。トラックログのサイズ (バイト単位) = $2 * ((\text{PVC のファイル数} * 200) + ((\text{PVC の KiB}/128\text{KiB の合計使用ディスク容量}) * 20))$
- 4 つのスナップショットからのバックアップジョブを同時に実行するために必要なストレージ = 4 つの PVC のトラックログサイズの合計。
つまり、Kubernetes クラスタあたりのスナップショットからのバックアップジョブの数が変わると、ストレージ要件も変わります。
- バックアップジョブを実行する前に、十分なストレージが利用可能であることを確認してください。ストレージの問題を避けるため、エラスティックストレージを使用できます。

バックアップストリーム

NetBackup アクセラレータは、バックアップストリームを次のように作成します。

- 名前空間に前回のバックアップがない場合、NetBackup は完全バックアップを実行します。
- 次回のバックアップジョブでは、NetBackup は、前回のバックアップ以降変更されたデータを識別します。変更されたブロックとヘッダー情報のみが、完全バックアップを作成するためにバックアップに含まれます。

- バックアップが完了すると、データムーバーの `bpbkar` によってトラックログが更新されます。データムーバー内のトラックログのパス - `usr/opensv/netbackup/track/<primary server>/<storage server>/<k8s cluster name>_<namespace uuid>_<pvc uuid>/<policy>/<backup selection>`
- このトラックログは、インライン形式でプライマリサーバーの次の場所に転送されます。
`/usr/opensv/netbackup/db/track/<primary server>/<storage server>/<k8s cluster name>_<namespace uuid>_<pvc uuid>/<policy>/<backup selection>`
- 次のアクセラレータのバックアップジョブが開始されると、変更されたファイルを識別するためにトラックログがプライマリサーバーからフェッチされます。その後、新しい内容で更新された後、プライマリサーバーに戻されます。

プライマリサーバーにあるトラックログのディスク容量の制御

アクセラレータによる有効なバックアップを続行するため、NetBackup は、トラックログの処理によってディスクの空きがなくなる状況を想定しています。空き容量が少なくなると、プライマリサーバー上のアクセラレータトラックログが問題になる可能性があります。

デフォルトでは、システムの空き容量が 5 GB または 5% を下回ると、NetBackup はアクセラレータバックアップを回避します。

NetBackup には、アクセラレータバックアップを開始するため、ホストの空きディスク容量を制御する 2 つの構成設定があります。

- `ACCELERATOR_TRACKLOG_FREE_SPACE_MB`
- `ACCELERATOR_TRACKLOG_FREE_SPACE_PERCENT`

各設定のデフォルト値はそれぞれ 5,120 MB と 5% です。十分な領域がない場合にバックアップをすばやく失敗させるには、プライマリサーバーの `bp.conf` ファイルでこれらの値を設定します。

ストレージクラスの動作がアクセラレータに与える影響

ストレージクラスの動作がアクセラレータに与える影響は次のとおりです。

- **Kubernetes** アクセラレータが有効なバックアップの場合、NetBackup は変更されたデータ量に基づいて最適化を示します。
ただし、後続のアクセラレータジョブの完了にかかる時間は、完全バックアップジョブを完了する場合とほぼ同様です。
- この状況は、データや、ファイルのメタデータが変更されたかどうかとは関係なく、ファイルの `INODE` や `CTIME` が変更されたストレージクラスの動作が原因で発生します。

- これは、ストレージクラスの内部実装が原因です。詳しくは、カスタマポータルサイトの Red Hat ナレッジベースの記事 (<https://access.redhat.com/solutions/7036388>) を参照してください。

アクセラレータ強制再スキャンについて

NetBackup アクセラレータ強制再スキャン機能のサポートは、破損したバックアップイメージの問題を防ぐのに役立ちます。アクセラレータ強制再スキャンを使用すると、選択したバックアップターゲットのすべてのデータがバックアップされます。

アクセラレータ強制再スキャンジョブを実行するには、`ForcedRescan` コマンドを手動で実行します。アクセラレータ強制再スキャンを使用すると、選択したバックアップターゲットのすべてのデータがバックアップされます。

このバックアップは、ポリシーの最初のアクセラレータバックアップに似ています。バックアップの所要時間は、アクセラレータを使わない場合のフルバックアップの所要時間とほぼ同様です。強制再スキャンによって安全性が強化され、次のアクセラレータバックアップの基準が確立されます。この機能は、チェックサム検証の失敗などの潜在的な損害から保護します。

強制再スキャンを使用する場合の推奨事項:

- 手動で強制再スキャンを実行してバックアップを開始するには、コマンドプロンプトまたは Linux 端末で次のコマンドを実行します。

```
bpbbackup -i -p <policy_name> -s ForcedRescan
```

- 例:

```
bpbbackup -i -p msdp_10mins_FRS+5d990ab5-f791-474f-885a-ae0c30f31c98  
-s ForcedRescan
```

次の API を使用して、強制再スキャンのスケジュールを開始できます。

```
POST admin/manual-backup
```

アクセラレータバックアップのエラーに関する警告と考えられる理由

警告	メッセージ	推奨処置
Kubernetes アクセラレータバックアップ機能は、この NetBackup メディアサーバーバージョンではサポートされていないため、アクセラレータなしのバックアップを実行します。	メディアサーバーのバージョンが 10.4 未満です。(Media server version is less than 10.4.)	メディアサーバーを 10.4 以降にアップグレードします。
トラックログ PVC の古いバージョンのクライアントまたはストレージクラスがクライアントで構成されていないため、クライアントはアクセラレータをサポートしません。ただし、クライアントはアクセラレータを使用しないバックアップを実行します。	NetBackup Kubernetes Operator のバージョンが 10.4 未満である、またはトラックログ PVC のストレージクラスがクライアントで構成されていません。(NetBackup Kubernetes operators version is less than 10.4 or Storage class for track log PVC is not configured on the client.)	NetBackup Kubernetes Operator とデータムーバーを 10.4 以降にアップグレードします。 トラックログ PVC のストレージクラスが Kubernetes Operator で正しく構成されていることを確認します。

Kubernetes での FIPS モードの有効化

この章では以下の項目について説明しています。

- [Kubernetes での FIPS \(連邦情報処理標準\) モードの有効化](#)

Kubernetes での FIPS (連邦情報処理標準) モードの有効化

NetBackup Kubernetes 10.3 リリースでは、RedHat ベースの NetBackup 配備に FIPS サポートを提供します。NetBackup、Kubernetes Operator、データムーバーに関連するすべての Kubernetes 作業負荷コンポーネントは、FIPS モードで実行する必要があります。FIPS サポートを実現するには、これらのすべてのコンポーネントで満たす必要がある特定の要件があります。

システム要件

NetBackup Kubernetes 作業負荷の FIPS サポートのシステム要件を次に示します。

名前	パラメータ
NetBackup プライマリおよびメディア	■ プライマリとメディアの両方が、FIPS が有効になっている RHEL-8 システムをベースにした NetBackup 10.2.1 に配備されている必要があります。 ■ RHEL OS は RHEL8 より新しいバージョンである必要があります。 ■ 次のコマンドを使用して、RedHat マシンのバージョンを確認できます。 <code>cat /etc/Redhat-release</code> ■ 次のコマンドを使用して、基盤となるシステムで FIPS が有効になっているかどうかを確認できます。 <code>FIPS-MODE-SETUP--CHECK</code> ■ 詳しくは、次のコマンドのマニュアルページのエントリを確認してください。 <code>fips-mode-setup</code>
Kubernetes クラスタ	■ Kubernetes クラスタは、FIPS 対応モードで配備する必要があります。 ■ FIPS モードで Kubernetes クラスタを配備するプロセスはベンダーによって異なります。 ■ たとえば、FIPS 対応の Openshift を配備します

構成パラメータ

NetBackup Kubernetes 作業負荷の FIPS モードの構成パラメータを次に示します。

構成	パラメータ
NetBackup プライマリおよびメディア	NetBackup プロセスを FIPS モードで実行できるようにします。 ■ 次のキーで <code><Netbackup-Installation-Path>/netbackup/bp.conf</code> を更新します。 <code>NB_FIPS_MODE = ENABLE</code> ■ FIPS モードの NetBackup について詳しくは、『NetBackup™ セキュリティおよび暗号化ガイド』の「NetBackup ドメインでの FIPS モードの構成」セクションを参照してください。

構成

NetBackup Kubernetes Operator

パラメータ

FIPS モードを有効にするには、次のいずれかを実行します。

- Helm Chart から `values.yaml` ファイルでパラメータ `fipsMode` の値を `ENABLE` に更新します。
- バックアップオペレータでパラメータ `NB_FIPS_MODE` の値を `ENABLE` に更新します。

メモ: NetBackup Kubernetes 作業負荷が実行されているすべてのシステムが FIPS 準拠であることを確認します。

FIPS のトラブルシューティング

AIR (自動イメージレプリケーション) 操作への影響:

- FIPS 対応環境の AIR では、追加の構成を行う必要があります。
- サポートサイトの <KB-Article> を更新してください。
- CLI (コマンドラインインターフェース) で次のコマンドを実行します。

```
/usr/opensv/java/jre/bin/keytool/keytool -storetype BCFKS
-providerpath
/usr/opensv/wmc/webserver/lib/ccj-3.0.1.jar -providerclass
com.safelogic.cryptocomply.jcajce.provider.CryptoComplyFipsProvider
-importcert -trustcacerts -file <target CA certificate file (pem
encoded)> -keystore
NB_INSTALL_DIR/var/global/wsl/credentials/cacerts.bcfks -storepass
<password from the /usr/opensv/var/global/jkskey file>
-alias <alias name of the trusted certificate entry to be added>
```

Openshift Virtualization のサポートについて

この章では以下の項目について説明しています。

- [Openshift Virtualization のサポート](#)
- [アプリケーションの一貫性がある仮想マシンのバックアップ](#)
- [仮想化のトラブルシューティング](#)

Openshift Virtualization のサポート

NetBackup では、Kubernetes クラスタで実行されている 1 台以上の仮想マシンで、名前空間に対するバックアップとリストアがサポートされます。

OpenShift クラスタでは、仮想化は `openshift-cnv` オペレータ (KubeVirt の RedHat バージョン) を使用してサポートされます。

Kubernetes での仮想化サポートにより、KubeVirt を使用して Kubernetes 環境に仮想マシンを配備できます。KubeVirt は Kubernetes 用の拡張機能で、コンテナと併せて仮想マシンを管理できます。KubeVirt により、仮想マシンを Kubernetes ポッドとして実行でき、コンテナと仮想マシン向けの一貫性のある管理インターフェースが実現します。

仮想マシンとその関連リソースがある名前空間を保護するための Kubernetes 保護計画を作成できます。

OpenShift 環境では、仮想マシンを含む名前空間のバックアップとリストアを正常に実行するために、必要なすべてのコンポーネントを正しくインストールして構成することが不可欠です。

- OpenShift Virtualization Operator (`openshift-cnv`) をインストールします。
- (オプション) Container Data Importer (CDI) をインストールします (ネットワークソースからディスクイメージを使用して仮想マシンを作成する場合)。

インテリジェントグループ

Kubernetes での仮想化サポートにより、特定のリソースの種類に基づいて名前空間をフィルタ処理するインテリジェントグループを作成できます。フィルタ処理で利用できるリソースの種類を次に示します。

- 仮想マシン
- 永続ボリューム
- 永続ボリューム要求

アプリケーションの一貫性がある仮想マシンのバックアップ

virt-launcher ポッドが仮想マシン (VM) の量産を担当している場合は、virt-launcher ポッドに NetBackup のプリフックとポストフックを注釈付けする必要があります。これがないと、VM を作成できないためです。

仮想マシンをフリーズおよびアンフリーズするコマンド:

```
■ /usr/bin/virt-freezer --freeze --name <vm-name> --namespace  
  <namespace>  
  
■ /usr/bin/virt-freezer --unfreeze --name <vm-name> --namespace  
  <namespace>  
  
# kubectl annotate pod -l vm.kubevirt.io/name=<vm-name>  
-n <vm-namespace>  
netbackup-pre.hook.backup.velero.io/command='["/usr/bin/virt-freezer",  
"--freeze", "--name", "<vm-name>", "--namespace", "<vm-namespace>"]'  
  
netbackup-pre.hook.backup.velero.io/container=compute  
  
netbackup-post.hook.backup.velero.io/command='  
["/usr/bin/virt-freezer", "--unfreeze", "--name",  
"<vm-name>", "--namespace", "<vm-namespace>"]'  
  
netbackup-post.hook.backup.velero.io/container=compute
```

Velero リストアフックをサポートする NetBackup

NetBackup で仮想マシンのリストア操作を実行した場合、KubeVirt ベースの仮想マシンについては、Velero のリストア前およびリストア後のフックが実行されません。この制限が生じるのは、KubeVirt により仮想マシンのランチャーポッドが動的に生成されるため、それらの作成プロセスが Velero のリストアワークフローから分離されているからです。

その結果、**Velero** はそれらの動的に作成されたポッドにフックを関連付けたり適用したりできません。

KubeVirt 仮想マシン内でリストア後の処理を実行するには、仮想マシンのリストア後に、`cloudInitNoCloud` メカニズムを使用して、ゲストオペレーティングシステム内にスクリプトを直接取り込み、実行します。

警告: **NetBackup** が **Velero** リストアフックをサポートしている場合でも、**NetBackup** で **KubeVirt** ベースの VM に使用する場合は制限事項があります。

メモ: アプリの一貫性を実現するには、仮想マシンに **qemu-guest-agent** をインストールして、**kubevirt** 固有の **pre-exec** ルールと **post-exec** ルールを実装する必要があります。

NetBackup のプリフックとポストフックの構成について詳しくは、<https://www.veritas.com/> を参照してください。

リストア後のフックの例:

```
InitContainer Restore Hooks annotations:
init.hook.restore.velero.io/container-image
init.hook.restore.velero.io/container-name
init.hook.restore.velero.io/command
```

```
InitContainer Restore Hooks As Pod Annotation Example:
$ kubectl annotate pod -n <POD_NAMESPACE> <POD_NAME> ¥
    init.hook.restore.velero.io/container-name=restore-hook ¥
    init.hook.restore.velero.io/container-image=alpine:latest ¥
    init.hook.restore.velero.io/command='["/bin/ash", "-c", "date"]'
```

```
Exec Restore Hooks annotations:
post.hook.restore.velero.io/container
post.hook.restore.velero.io/command
post.hook.restore.velero.io/on-error
post.hook.restore.velero.io/exec-timeout
post.hook.restore.velero.io/wait-timeout
```

```
Exec Restore Hooks As Pod Annotation Example:
$ kubectl annotate pod -n <POD_NAMESPACE> <POD_NAME> ¥
    post.hook.restore.velero.io/container=postgres ¥
    post.hook.restore.velero.io/command='["/bin/bash", "-c", "psql
```

```
< /backup/backup.sql"]' ¥  
  post.hook.restore.velero.io/wait-timeout=5m ¥  
  post.hook.restore.velero.io/exec-timeout=45s ¥  
  post.hook.restore.velero.io/on-error=Continue
```

仮想化のトラブルシューティング

- 同じクラスタで、静的 **MAC** (メディアアクセス制御アドレス) を定義した **VM** のリストアが失敗します。
ユーザーが、代替の名前空間を使用して同じクラスタで仮想マシンのリストア操作を行うと、既存の **VM MAC** がソース **VM** に割り当てられる可能性があります。ユーザーは、**MAC** がどの **VM** にも割り当てられていないことを確認する必要があります。
- 名前空間のクロスクラスタリストア中に、ファイルシステムに提供されたストレージクラスがソース **PVC** と異なる場合、リストア操作が失敗する場合があります。
これを防ぐため、ユーザーは、ソースクラスタとターゲットクラスタに同じストレージクラスがあることを確認する必要があります。

Kubernetes の問題のトラブルシューティング

この章では以下の項目について説明しています。

- プライマリサーバーのアップグレード時のエラー: **NBCheck** が失敗する
- 古いイメージのリストア時のエラー: 操作が失敗する
- 永続ボリュームのリカバリ **API** でのエラー
- リストア中のエラー: ジョブの最終状態で一部が失敗していると表示される
- 同じ名前空間でのリストア時のエラー
- **datamover** ポッドが **Kubernetes** のリソース制限を超過
- リストア時のエラー: 高負荷のクラスタでジョブが失敗する
- 特定のクラスタ用に作成されたカスタムの **Kubernetes** の役割でジョブを表示できない
- **OperatorHub** からインストールされたアプリケーションのリストア時に、選択されていない空の **PVC** が **Openshift** によって作成される
- **Kubernetes** ノードで **PID** の制限を超えると **NetBackup Kubernetes Operator** が応答しなくなる
- **NetBackup Kubernetes 10.1** におけるクラスタの編集時のエラー
- サイズの大きい **PVC** のバックアップまたはリストアが失敗する
- 名前空間ファイルモードの **PVC** を別のファイルシステムにリストアすると部分的に失敗する
- バックアップコピーからのリストアがイメージの不整合エラーで失敗する

- **NetBackup** プライマリサーバー、メディアサーバー、**Kubernetes** サーバー間の接続性チェック
- ログに利用可能な領域がない場合のアクセラレータバックアップ中のエラー
- ログ **PVC** の作成エラーによるアクセラレータバックアップ中のエラー
- 無効なアクセラレータストレージクラスによるアクセラレータバックアップ中のエラー
- ログポッドの起動中に発生したエラー
- ログ **PVC** 操作のためのデータムーバーインスタンスの設定に失敗する
- **configmap** からログのストレージクラスを読み取る際のエラー
- 異なるクラスに単一の **VM** をリストアするときにリストア操作が失敗する
- **backupservercert** の作成時に **Kubernetes** 作業負荷の自動配備が失敗する
- **Kubernetes** 作業負荷の自動配備がタイムアウトエラーで失敗する
- エラー「**2890 K8s** のデータのリストア操作が失敗しました (**2890 K8s restore data operation failed**)」が表示され、即座に終了リストア戦略が失敗しました。

プライマリサーバーのアップグレード時のエラー: **NBCheck** が失敗する

NetBackup プライマリサーバーのバージョン **9.1** から **10.0** へのアップグレードが失敗し、重要でない **NBCheck** エラーが発生します。

エラーメッセージ: テストにより、**{{ポリシーの数}}** 個の有効な **Kubernetes** ポリシーが見つかりました。**NetBackup** インスタンスに有効な **Kubernetes** ポリシーがある場合、このテストは失敗します。(The test found **{{no. of policies}}** active **Kubernetes** policy. This test fails if the **NetBackup** instance has any active **Kubernetes** policies.)

推奨処置: **NetBackup** をバージョン **10.0** にアップグレードする前に、プライマリサーバーで有効な **Kubernetes** ポリシーをすべて無効にします。

詳しくは、<https://support.cohesity.com/s/> を参照してください。

古いイメージのリストア時のエラー: 操作が失敗する

NetBackup 9.1 バージョンを使用して作成された古いイメージでは、**Kubernetes** のリストア操作が失敗します。

エラーメッセージ: 10.0 より前のバージョンの NetBackup のバックアップイメージでは、リストア操作はサポートされません。(Restore operation is not supported on the backup images of NetBackup older than 10.0 version.)

推奨処置: Velero コマンドを使用して古いイメージをリストアします。Velero は、安全にバックアップとリストアを行い、ディザスタリカバリを実行し、Kubernetes クラスタリソースと永続ボリュームを移行するためのオープンソースツールです。そのため、Velero から古いイメージをリストアするには、インストールがクラスタでの前提条件です。

NetBackup 管理者の Web UI からバックアップ名またはバックアップ ID を取得し、それを Velero コマンドで使用してリストアします。

詳しくは、<https://support.cohesity.com/s/> を参照してください。

永続ボリュームのリカバリ API でのエラー

NetBackup Kubernetes Operator バージョン 10.0 では、永続ボリュームのリカバリ API が削除され、サポートされません。NetBackup の古いバージョンでは、永続ボリュームのリストアにこの API が使用されていました。そのため、NetBackup 10.0 バージョンにアップグレードした場合、永続ボリュームのリカバリ API を使用してリストアすると、リストア操作は失敗します。

エラーメッセージ: NetBackup の Kubernetes リカバリ処理の再設計に伴い、Kubernetes 永続ボリュームのリカバリ API は使用できなくなり、製品から削除されました。(Kubernetes persistent volume recovery API is no longer in use and has been removed from the product due to redesign at NetBackup Kubernetes recovery process.)

推奨処置: NetBackup Kubernetes Operator バージョン 10.0 では、選択したリソースをバックアップからリカバリするように NetBackup がアップグレードされています。したがって、永続ボリュームまたは永続ボリューム要求をリカバリする場合は、NetBackup から永続ボリュームを選択し、宛先名前空間にリカバリできます。

詳しくは、<https://support.cohesity.com/s/> を参照してください。

リストア中のエラー: ジョブの最終状態で一部が失敗していると表示される

リストアジョブの最終状態で一部が失敗しており、リソース RoleBinding に固有の警告がいくつか表示されます。

表示される警告は、API グループ `groupauthorization.openshift.io` と `rbac.authorization.kubernetes.io` のリソース RoleBinding に固有です。

RoleBinding は、コントローラを使用して自動管理され、新しい名前空間を作成するときに作成されるためです。

推奨処置: 関連する **RoleBinding** リソースをリストアから除外するか、作成された警告を無視できます。

同じ名前空間でのリストア時のエラー

選択した **PVC** が名前空間にすでに存在する場合、元の名前空間に **PVC** をリストアすると失敗することがあります。

推奨処置:

- 代替名前空間のリストアを使用できます。
- [リカバリオプション (Recovery option)]で、リストア操作の実行中に既存の **PVC** と重複していない **PVC** を選択できます。

datamover ポッドが Kubernetes のリソース制限を超過

NetBackup は、2 つのリソース制限プロパティを使用して、**Kubernetes** 作業負荷における実行中のバックアップジョブの合計数を制御します。**NetBackup** バージョン 10.0 では、**datamover** ポッドが **Kubernetes** クラスタごとに設定されたリソース制限[バックアップ]と[スナップショットからのバックアップ]を超過します。

リソース制限の問題の例

例 1

Activity monitor				
Jobs		Daemons	Processes	Background tasks
 Search...				
Job ID ↓	Type	Client or display name		Job state
☐ 3021	Backup From Snapshot	nginx-logs;34.68.168.50		Queued
▼ ☐ 3020	Backup From Snapshot	nginx-rfb;34.68.168.50		Active
☐ 3022	Backup From Snapshot	nginx-rfb;34.68.168.50		Active
▼ ☐ 3018	Backup	kacustervm		Done
☐ 3019	Snapshot	nginx-rfb;34.68.168.50		Done

Kubernetes クラスタあたりのスナップショットからのバックアップジョブのリソース制限は 1 に設定されています。

ジョブ ID 3022 は子ジョブで、クラスタからストレージユニットへのデータ移動が行われます。

datamover ポッドをクリーンアップし、親ジョブ ID 3020 のライフサイクルを完了するプロセスを進めているため、ジョブ ID 3020 がまだ進行中であることに注意してください。

Activity monitor

Jobs	Daemons	Processes	Background tasks
Search...			
Job ID ↓	Type	Client or display name	Job state
☐  3021	Backup From Snapshot	nginx-logs;34.68.168.50	Active
▼ ☐  3020	Backup From Snapshot	nginx-rfb;34.68.168.50	Active
☐  3022	Backup From Snapshot	nginx-rfb;34.68.168.50	Done
▼ ☐  3018	Backup	kaclustervm	Done
☐  3019	Snapshot	nginx-rfb;34.68.168.50	Done

スナップショットからのバックアップジョブが複数トリガされるビジー状態の環境では、リソース制限値を低く設定すると、バックアップジョブはほとんどの時間、キューに投入された状態になる可能性があります。

3022 などのデータ移動ジョブは並行して実行されますが、クリーンアップアクティビティは順次処理されます。これは、**datamover** リソースのクリーンアップにかかる時間と組み合わせたときに、**PVC** または名前空間データのバックアップにかかる時間に近づくと、ジョブの完了がさらに遅延することになります。

データ移動とリソースのクリーンアップの合計時間がバックアップジョブと同じ場合、その後、永続ボリュームまたは名前空間データのバックアップジョブによって、ジョブの完了が遅れる可能性があります。

推奨処置: システムのリソースとパフォーマンスを確認し、それに応じてリソース制限値を設定します。この測定は、すべてのバックアップジョブで最高のパフォーマンスを実現するのに役立ちます。

リストア時のエラー: 高負荷のクラスタでジョブが失敗する

高負荷の **Kubernetes** クラスタではリストアジョブが失敗します。

エラーメッセージ: **ERR - VxMS** を初期化できません。ソケットにデータを書き込めません。peer により接続がリセットされました。(ERR - Unable to initialize VxMS, cannot write data to socket, Connection reset by peer.)

クライアント **cluster.sample.domain.com** のエラー **bpbrm (pid=712755): ERR - VxMS** を初期化できません。(Error bpbrm (pid=712755) from client cluster.sample.domain.com: ERR - Unable to initialize VxMS.)

エラー **bptm (pid=712795)** ソケットにデータを書き込めません。peer により接続がリセットされました。(Error bptm (pid=712795) cannot write data to socket, Connection reset by peer.)

推奨処置: リストア操作中にこの問題が発生した場合は、リストア操作を負荷の小さいクラスタで実行するか、クラスタがアイドル状態のときに実行する必要があります。

特定のクラスタ用に作成されたカスタムの **Kubernetes** の役割でジョブを表示できない

特定の **Kubernetes** クラスタで **Kubernetes** 作業負荷用にカスタムの **RBAC** の役割が作成されたら、システム管理者は、**Kubernetes** ジョブを表示する権限を明示的に付与する必要があります。そうしないと、**Kubernetes** 固有のジョブはすべて、表示されません。

システム管理者が **Kubernetes** ジョブを表示する権限を付与しない場合、ユーザーが表示できるジョブは次のとおりです。

- 階層表示のリストアジョブのみ。
- 一覧表示のスナップショットジョブとリストアジョブのみ。

作成されたカスタムベースの **Kubernetes** の役割で特定の **Kubernetes** クラスタのジョブを表示できない場合、次の手順を実行して表示権限を付与します。

表示権限を付与するには

- 1 左側で[作業負荷 (Workload)]で[Kubernetes]をクリックします。
- 2 右側で[Kubernetes 設定 (Kubernetes setting)]、[権限を管理 (Manage permissions)]の順にクリックします。
- 3 対応する役割の横にある縦型の省略記号をクリックし、[編集 (Edit)]を選択します。
- 4 [権限の編集 (Edit permissions)]で、役割の[編集 (Edit)]権限と[ジョブの表示 (View jobs)]権限を選択し、[保存 (Save)]をクリックします。

Kubernetes のカスタム役割のユーザーは、階層表示と一覧表示の両方で、バックアップジョブ、スナップショットジョブ、リストアジョブ、スナップショットからのバックアップジョブを表示できるようになります。

想定:

- 設定がアップグレードされると、ユーザーは次のものを表示できます。
 - 階層表示にある、既存のジョブからのリストアジョブのみ。
 - 一覧表示にある、既存のジョブからのスナップショットジョブとリストアジョブのみ。
- 選択した Kubernetes クラスタに対する権限を指定して Kubernetes のカスタム役割が作成されると、ユーザーはスナップショットジョブのみで操作をキャンセルおよび再開できます。

OperatorHub からインストールされたアプリケーションのリストア時に、選択されていない空の PVC が Openshift によって作成される

アプリケーションが OperatorHub カタログソースを介してインストールされる Openshift 環境では、ユーザーが、そのようなアプリケーション名前空間のバックアップから選択した PVC のリストアを実行しようとした場合、代わりにすべての PVC が作成されます。

この問題は、Openshift 環境では、リストア先の名前空間に必要なサイズで、選択されていない PVC がプロビジョニングされるために発生します。

メモ: このようなアプリケーションでは、ユーザーがリストア対象として PVC を選択しなくても、配備の構成に従って PVC は自動プロビジョニングされます。

Kubernetes ノードで PID の制限を超えると NetBackup Kubernetes Operator が応答しなくなる

Linux システムでは、ゾンビプロセスを消去するために PID 1 として実行される `initd` またはシステムプロセスが存在します。そのような `initd` プロセスを持たないコンテナでは、ゾンビプロセスが生成され続けます。

一定の期間が経過すると、このようなゾンビプロセスが蓄積され、Kubernetes ノードで設定されている PID の上限に達します。

NetBackup Kubernetes Operator では、`nbcertcmdtool` が証明書関連の操作を実行するために子プロセスを生成します。操作が完了すると、プロセスは孤立し、消去されません。最終的に PID の上限に達し、NetBackup Kubernetes Operator は応答しなくなります。

Error message: login pod/nbukops-controller-manager-67f5498bbb-gn9zw -c netbackupkops -n nbukops ERRO[0005] exec failed: container_linux.go:380: starting container process caused: read init-p: connection reset by peer a command that is terminated with exit code 1.

推奨処置:

- PID の上限の超過の問題を解決するには、`Initd` スクリプトを使用します。`Initd` スクリプトは、コントローラポッドの親プロセスまたはエントリポイントスクリプトとして機能します。
親プロセスは、子プロセスの完了後にゾンビプロセスを自分自身に接続し、永続的なゾンビプロセスを終了させます。また、コンテナを正常にシャットダウンするためにも役立ちます。`Initd` スクリプトは、NBUKOPs ビルドバージョン 10.0.1 で利用可能です。
- 既存の `nbcertcmdtool` ゾンビプロセスを削除するには、次の手順を実行します。

1. NetBackup オペレータポッドを記述し、コントローラポッドが実行されている Kubernetes ノードを見つけます。次のコマンドを実行します。

```
kubectl describe -c netbackupkops <NB k8s operator pod name> -n <namespace>
```

2. 次のコマンドを実行し、Kubernetes ノードにログオンします。

```
kubectl debug node/nodename
```

3. 次のコマンドを実行し、`nbcertcmdtool` ゾンビプロセスを終了します。

```
ps -ef | grep "[nbcertcmdtool]" | awk '{print $3}' | xargs kill -9
```

メモ: これらの手順により、そのワーカーノードのすべてのゾンビプロセスが終了します。ただし、この問題の解決は一時的です。永続的な解決策としては、`initd` スクリプトを使用して新しい `KOps` ビルドを配備する必要があります。

NetBackup Kubernetes 10.1 におけるクラスタの編集時のエラー

Kubernetes クラスタの編集操作には問題があり、NetBackup Kubernetes 10.1 バージョンでは動作しません。

対処方法: クラスタを編集するには、まず保護計画から Kubernetes クラスタを削除してから、クラスタを再び追加する必要があります。

サイズの大きい PVC のバックアップまたはリストアが失敗する

大きいサイズの PVC で、構成されたポーリングタイムアウト時間内に PVC がバインドされないと、スナップショットからのバックアップと、スナップショットまたはバックアップからのリストアが失敗します。この問題は、大容量のボリュームスナップショットのハイドレーションにデフォルトの 15 分のタイムアウトより長い時間がかかるために発生します。

スナップショットからのバックアップ

サイズが大きい PVC (例: 1.5 TB) でスナップショットからのバックアップがエラーコード 34 で失敗する

エラーメッセージ:

Error nbcs (pid=250908) failed to setup the data mover instance for tracklog pvc operation.

Error nbcs (pid=250908) unable to initialize the tracklog data mover instance, data mover pod status: Pending reason:Failed message>Error: context deadline exceeded.

スナップショットからのリストアまたはバックアップからのリストア

サイズが大きい PVC (例: 100 GB) でスナップショットからのリストアがエラーコード 5 で失敗する

エラーメッセージ:

Error nbcs (pid=29228) timeout occurred while waiting for the persistent volume claim pvc-sample status to be in the bound phase

推奨処置:

バックアップオペレータ configmap でポーリングタイムアウトを大きくします。

- ConfigMap 名: <kops-name>-backup-operator-configuration
- 更新するキー: pollingTimeoutInMinutes

名前空間ファイルモードの PVC を別のファイルシステムにリストアすると部分的に失敗する

名前空間ファイルモードの PVC を別のファイルシステムにリストアすると、名前空間ボリュームが部分的に成功します。この場合、ソースファイルシステム以外のファイルシステムにファイルシステムオブジェクト(ファイルまたはディレクトリ)をリストアすると、互換性のないメタデータのリストアに失敗します。その結果、この操作は部分的に成功したリストアとして表示されます。

Error message: 7:38:57 AM - Error bpbrm (pid=30171) client restore EXIT STATUS 1: the requested operation was partially successful.

対処方法: 宛先ファイルシステムを確認し、ファイルが配置されていることを確認します。ファイルがリストアされる時、実際にはデータに問題はありません。この部分的なエラーは、メタデータのリストアに問題があり、オペレータがそのことを認識する必要があるという助言として報告されます。

バックアップコピーからのリストアがイメージの不整合エラーで失敗する

古いバージョンのメディアサーバーをストレージに使用すると、バックアップコピーからのリストアがイメージの不整合エラーで失敗します。たとえば、ストレージに 10.1.1 より古いバージョンのメディアサーバー、バックアップコピーからのリストアに NetBackup バージョン 10.1.1 が使用されている場合などが該当します。

Error message: Sep 22, 2022 3:12:55 PM - Info tar (pid=1459) done. status: 229: events out of sequence - image inconsistency Sep 22, 2022 3:12:55 PM - Error bpbrm (pid=16523) client restore EXIT STATUS 229: events out of sequence - image inconsistency

対処方法: すべての Kubernetes ワークフローで、Kubernetes ファイルシステムベースのバックアップには、常にプライマリ、メディア、NetBackup Kubernetes Operator バージョン 10.1.1 を使用する必要があります。

NetBackup プライマリサーバー、メディアサーバー、Kubernetes サーバー間の接続性チェック

NetBackup プライマリホストと他のホスト間の接続を確認するには、次のコマンドを参照できます。

- NetBackup プライマリサーバーとの通信を容易にするために必要なポートを開いたら、次のコマンドを実行できます。
- NetBackup プライマリサーバー/メディアサーバーと Kubernetes クラスタ間の接続を確認するには、Kubernetes Operator のポッドから次のコマンドを実行します。

```
curl -v telnet://<netbackup-server-host>:<port-no>
```

- NetBackup プライマリサーバーと Kubernetes クラスタ間の接続を確認するには、NetBackup プライマリホストから次のコマンドを実行します。

```
curl -v telnet://<kubernetes-api-server-host>:<port-no>
```

メモ: これらの両方のコマンドに対する応答に、接続が正常に確立されたことが示されている必要があります。

トラックログに利用可能な領域がない場合のアクセラレータバックアップ中のエラー

トラックログを保存するための十分な領域がない場合、スナップショットからのバックアップジョブは、「ソケットの書き込みに失敗しました (socket write failed) (24)」というエラーで失敗します。

エラーメッセージ: 状態 24 で重複があります (ソケットの書き込みに失敗) (Duplicate existed with status 24 (socket write failed))。

推奨処置: バックアップジョブを正常に実行するには、トラックログが格納されるパスに十分なストレージが必要です。

トラックログ PVC の作成エラーによるアクセラレータバックアップ中のエラー

トラックログ PVC の作成でエラーが発生した場合、スナップショットからのバックアップジョブは失敗します。トラックログ PVC の作成は、次のような複数の理由により失敗する場合があります。

- 無効なストレージクラスが指定されている。
- PVC 作成のための十分な空き容量がない。

推奨処置:

- 必要なサイズの PVC を作成するのに十分な空き容量があるかどうかを確認します。
- Kubernetes クラスタでストレージクラスが正しく構成されているかどうかを確認します。

無効なアクセラレータストレージクラスによるアクセラレータバックアップ中のエラー

アクセラレータバックアップジョブに無効なストレージクラスが指定されている場合、トラックログ PVC の作成は失敗します。PVC 状態がバインドフェーズになるのを待機している間にエラーが発生します。

エラーメッセージ: StorageClass.storage.k8s.io cstor-storage-class-x2 が見つかりません (StorageClass.storage.k8s.io cstor-storage-class-x2 not found)

Jan 11, 2024 2:12:54 AM - Error nbcs (pid=92639) StorageClass.storage.k8s.io cstor-storage-class-x2 が見つかりません (Jan 11, 2024 2:12:54 AM - Error nbcs (pid=92639) StorageClass.storage.k8s.io cstor-storage-class-x2 not found)

推奨処置: バックアップオペレータ configmap で有効なストレージクラスを指定して、バックアップジョブを再実行します。

トラックログポッドの起動中に発生したエラー

エラーメッセージ:トラックログのデータムーバーインスタンスを初期化できません。(Unable to initialize the track log data mover instance.)

推奨処置: describe コマンドを使用して、Kubernetes クラスタのポッド作成ログで詳細なエラーを確認します。

トラックログ PVC 操作のためのデータムーバーインスタンスの設定に失敗する

エラーメッセージ:トラックログデータムーバーポッドの状態とイベントのフェッチに失敗しました。(Failed to fetch track log data mover pod status and events.)

推奨処置: describe コマンドを使用して、Kubernetes クラスタのポッド作成ログで詳細なエラーを確認します。

configmap からトラックログのストレージクラスを読み取る際のエラー

エラーメッセージ: トラックログのストレージクラスの取得に失敗しました。(Failed to get storage class for track log.)

推奨処置: NetBackup Kubernetes Operator が正しく構成されているかどうかを確認します。

異なるクラスタに単一の VM をリストアするときにリストア操作が失敗する

エラーメッセージ: K8s のデータのリストア操作が失敗しました (2890)。

原因: このエラーは、ソースと異なるストレージプロビジョナが存在する代替クラスタに仮想マシン PVC をリストアするときに発生します。ターゲットとソースのデータサイズが同じでも、ストレージのプロビジョニングが異なるため、ファイルシステムのオーバーヘッドが発生します。このような場合、メモリが十分に割り当てられていないため、ストアが失敗することがあります。

メモ: ただし、上記のエラーは、同じストレージプロビジョナ、CephFs、および一致する PVC サイズの場合の例外です。

解決方法:

- 同じストレージプロビジョナを使用する

backupservercert の作成時に Kubernetes 作業負荷の自動配備が失敗する

values.yaml ファイルのプライマリサーバー名に大文字が含まれていると、backupservercert の作成時に Kubernetes 作業負荷の自動配備が失敗します。

backupservercert の作成プロセス中、NetBackup を配備するときに Kubernetes Secret (secret-PRIMARY) を作成しようとします。Kubernetes では、リソース名に RFC 1123 サブドメイン形式ルールが適用されます。

secret-PRIMARY には大文字 (PRIMARY) が含まれ、この規則に違反するため、拒否されます。

Kubernetes 向けに NetBackup を配備する場合は、values.yaml ファイルのプライマリサーバー名がすべて小文字で書かれていることを確認してください。

Kubernetes 作業負荷の自動配備がタイムアウトエラーで失敗する

NetBackup バックアップサーバーの証明書の構成にデフォルトのポーリングタイムアウト (5 分間) 以上時間がかかると、Kubernetes 作業負荷の自動配備が失敗することがあります。

NetBackup Kubernetes Operator の「最大オープンファイル数」の制限が大きすぎると、バックアップサーバーの証明書の設定時にタイムアウトする場合があります。Kubernetes では、コンテナはノード上のコンテナランタイム (Docker や containerd など) からこの制限を取得します。一般的に、制限が大きいと、証明書の構成速度が低下し、デフォルトのタイムアウトを超える場合があります。

この問題を解決するには、コンテナランタイムの構成を使用して、最大オープンファイル数の値を適切なレベルまで減らします。Kubernetes はこの制限を直接管理していないため、各クラスターノードのランタイム設定を更新する必要があります。再構成後、NetBackup Kubernetes の自動配備を再実行します。

詳しくは、[記事](#)を参照してください。

エラー「2890 K8s のデータのリストア操作が失敗しました (2890 K8s restore data operation failed)」が表示され、即座に終了リストア戦略が失敗しました。

問題:

即座に終了戦略を選択している場合に、50 を超える PVC がリストアジョブに含まれていると、次のエラーが発生してジョブは失敗し、VM はリストアされません。

```
2890 K8s restore data operation failed
```

メモ: このエラーは Windows でのみ発生します。

原因:

この問題は、通常、多数の PVC を同時に処理する際にサーバーへの負荷が大きくなるのが原因で発生します。即座に終了戦略では、すべての PVC を並行してリストアします。このため、システムリソースに大きな負荷がかかり、エラーが発生することがあります。

解決方法:

- ジョブの詳細で、[再試行 (Retry)] オプションを選択します。再試行時には、システムが異なる方法で作業負荷を処理するため、リソースの競合が軽減され、リストア操作は成功します。

エラー「**2890 K8s** のデータのリストア操作が失敗しました (**2890 K8s restore data operation failed**)」が表示され、即座に終了リストア戦略が失敗しました。

- 可能な場合は、単一のリストアジョブの **PVC** の数を減らすか、大規模なリストアには即座に終了戦略を使用しないようにします。