



ストレージと仮想化の入門

NetApp virtualization solutions

NetApp
July 21, 2026

目次

ストレージと仮想化の入門	1
NetApp Tridentと Red Hat OpenShift の統合について学ぶ	1
NetApp Trident	1
Trident Protect	1
NetApp ONTAPのサポートされているハードウェアモデルとプロトコル	2
ONTAP ストレージ向け Trident ドライバー	2
Red Hat OpenShift クラスタに Trident をインストールしてストレージオブジェクトを作成する	4
ステップ1：Tridentをインストールする	5
ステップ2：環境のストレージバックエンドとStorageClass設定ファイルを準備する	9
ステップ3：VolumeSnapshotClass設定ファイルを作成する	29
ステップ4：構成ファイルをクラスタに適用する	29
ステップ5：デフォルトのTridentストレージおよびスナップショットクラスを設定する	31
ONTAPを使用した Red Hat OpenShift Virtualization の導入要件	32
前提条件	32
NetApp Trident CSI、ストレージプロファイル、OpenShift Virtualization のボリュームアクセスモード	33
Red Hat OpenShift Virtualization で ONTAP ストレージを使用して VM を作成する	38
VMの作成	39
ビデオデモンストレーション	42
Red Hat OpenShift Virtualization で、代替手段を使用して ONTAP ストレージを通じて VM を作成する ..	42
Red Hat OpenShift Virtualization を使用して YAML ファイルから VM を作成する	42
Red Hat OpenShift Virtualization を使用してスナップショット コピーから VM を作成する	44
OpenShift Virtualization で Trident を使用して VM をクローンする	48
Red Hat OpenShift クラスター内の 2 つのノード間で VM を移行する	52
VMライブマイグレーション	52
OpenShift Virtualization におけるストレージクラス間の VM のストレージ移行	54

ストレージと仮想化の入門

NetApp Tridentと Red Hat OpenShift の統合について学ぶ

ストレージはコンテナ化されたアプリケーションにおいて非常に重要であり、動的でしばしば一時的な環境においても、データが常に利用可能であり、効率的に管理されることを保証します。コンテナ向けの最新のストレージソリューションは、アプリケーションに合わせてシームレスに拡張できるように設計されており、アプリケーションのニーズに基づいてストレージリソースを動的に割り当てる柔軟性を提供し、成長と多様なワークロードの両方をサポートします。OpenShift Virtualization により、VM はコンテナとして実行されるため、OpenShift クラスター内の他のアプリケーションと同じストレージを使用します。

NetApp ONTAP は、NetApp のエンタープライズクラスのストレージソリューションの基盤となる、堅牢で拡張性に優れたデータ管理ソフトウェアです。オンプレミス、クラウド、ハイブリッド構成など、さまざまなストレージ環境とシームレスに統合されるユニファイド ストレージ アーキテクチャを提供します。ONTAP は、NFS、CIFS/SMB、iSCSI、FC、NVMe over TCP など、さまざまなプロトコルをサポートしており、さまざまなアプリケーションやワークロードとのシームレスな統合が可能です。これにより、SAN、NAS、オブジェクトストレージを単一のプラットフォームに統合することが可能になり、管理の簡素化とコスト削減を実現します。NetApp ONTAP は、オンプレミスのほか、クラウドにおけるマネージドサービスとしても利用可能です。

NetApp Trident

Tridentは、NetAppが開発したオープンソースのストレージオーケストレーターであり、コンテナ化された環境でのONTAPストレージの使用を容易にするために設計されています。これにより、コンテナ化されたアプリケーション向けの永続ストレージの動的なプロビジョニングと管理が可能になります。

Trident の主な機能

CSI準拠 最新のKubernetesストレージ機能および標準との互換性を確保します。宣言型構成 ユーザーがYAML ファイルでストレージ要件を定義できるようにし、その後Tridentによって自動的に管理されます。ドライバー TridentはONTAPがサポートするNFS、CIFS/SMB、iSCSI、FC、NVMe/TCPプロトコルのドライバーをサポートします。ハイブリッドおよびマルチクラウドのサポート TridentはオンプレミスのONTAPストレージおよびハイパースケーラーのマネージドストレージサービス（AWSのFSx for NetApp ONTAP、AzureのAzure NetApp Files、Google CloudのGoogle Cloud NetApp Volumes）のドライバーをサポートします。高度なデータ管理 ONTAPのスナップショットおよびクローニング機能を活用し、効率的なデータ保護とテスト環境および開発環境の迅速なプロビジョニングを実現します。

Tridentの詳細については、NetApp Trident "[ドキュメントはこちら](#)"を参照してください。

Trident Protect

Trident Protect は、コンテナ化された環境において強化されたデータ保護とセキュリティを提供する Trident の高度な機能です。重要なデータを潜在的な脅威やデータ損失のシナリオから確実に保護するように設計されています。すべての Trident ユーザーが無償で利用でき、Trident がインストールされた任意の OpenShift クラスターで簡単に有効化できます。

Trident Protect の主な機能

自動バックアップ Trident Protect は、自動化されたポリシーベースのコンテナアプリケーションおよび VM のバックアップを実現し、手動操作なしで定期的にバックアップが行われるようにします。

スナップショット管理 ONTAPのスナップショット機能を活用して、コンテナアプリケーションおよびVMのポイントインタイムコピーを頻繁に作成し、信頼性の高いリカバリメカニズムを提供します。

データ暗号化 データが保存時と転送時の両方で暗号化されることを保証し、厳格なセキュリティ要件を満たし、機密情報を保護します。

コンプライアンスと監査 組織がコンプライアンス要件を満たし、データ保護慣行の定期的な監査を実施するのに役立つツールと機能を提供します。

ディザスタリカバリ ONTAPのレプリケーション機能と統合することで、堅牢なディザスタリカバリソリューションを提供し、壊滅的な事態が発生した場合でも事業継続性を確保します。

Trident Protect の詳細については、NetApp Trident Protect "[ドキュメントはこちら](#)"を参照してください。

NetApp ONTAPのサポートされているハードウェアモデルとプロトコル

NetApp ONTAP ソフトウェアは、さまざまなハードウェアモデル上で動作し、それぞれが異なる性能および容量要件に対応するように設計されています。Trident は、All Flash FAS (AFF)、All SAN Array (ASA)、および ASA R2 システムを含むさまざまな NetApp ONTAP システムをサポートする堅牢な機能を拡張します。このサポートにより、組織はコンテナ化された環境においてハイパフォーマンス ストレージ ソリューションの潜在能力を最大限に活用できるようになります。

オールフラッシュ **FAS (AFF)** システム AFF システムは優れたパフォーマンスと低レイテンシを実現し、高負荷アプリケーションに最適です。

All SAN Array (ASA) システム ASA システムは専用の SAN 環境向けに設計されており、堅牢なパフォーマンスと信頼性を提供します。

ASA R2 Systems ASA R2システムは、SAN環境向けに強化された機能と性能を提供します。

AFX NetApp AFX は、エンタープライズ AI ワークロード向けに設計された、目的に即した分散型オールフラッシュ ストレージ アーキテクチャです。ハイパフォーマンス NAS を提供し、コンピューティングと容量を独立してスケーリングできます。NetApp AFX ストレージシステムは、ストレージ レイヤの実装において、他の ONTAP ベースのシステム (ASA、AFF、FAS) とは異なります。AFX は高性能と拡張性を実現する分散ファイルシステムを使用しますが、他の ONTAP ベースのシステムは従来型のストレージ アーキテクチャを使用します。

1. AFF でサポートされているプロトコル：NFS、CIFS/SMB、iSCSI、FC、NVMe/TCP
2. ASA でサポートされているプロトコル：iSCSI、FC、NVMe/TCP
3. ASA R2 でサポートされているプロトコル：iSCSI、FC、NVMe/TCP
4. AFX でサポートされているプロトコル：Trident 25.10 以降、NFS プロトコルのみがサポートされており、SMB プロトコルはサポートされていません。

ONTAP ストレージ向け Trident ドライバー

Tridentには以下のCSIドライバが含まれており、それぞれバックエンドストレージにボリュームをプロビジョ

ニングできます。

対応ハードウェアモデルの**NAS**プロトコルについて

1. ontap-nas: 1 つの PVC が 1 つの PV にマッピングされ、これは専用の FlexVolume です
2. ontap-nas-economy: プロビジョニングされた各 PV は qtree であり、FlexVolume ごとに設定可能な qtree 数（デフォルトは 200、50 ～ 300 の間で設定可能）を持ちます。1 つの PVC は、共有 FlexVolume 上の qtree である 1 つの PV にマッピングされます。



仮想マシンのすべてのディスクをqtreeとして、単一のボリューム内に配置できます。ただし、スナップショット、クローン、レプリケーション機能はTridentではサポートされていないことにご注意ください。これらの機能がストレージ管理者によって管理される場合、PVの粒度には対応していません。

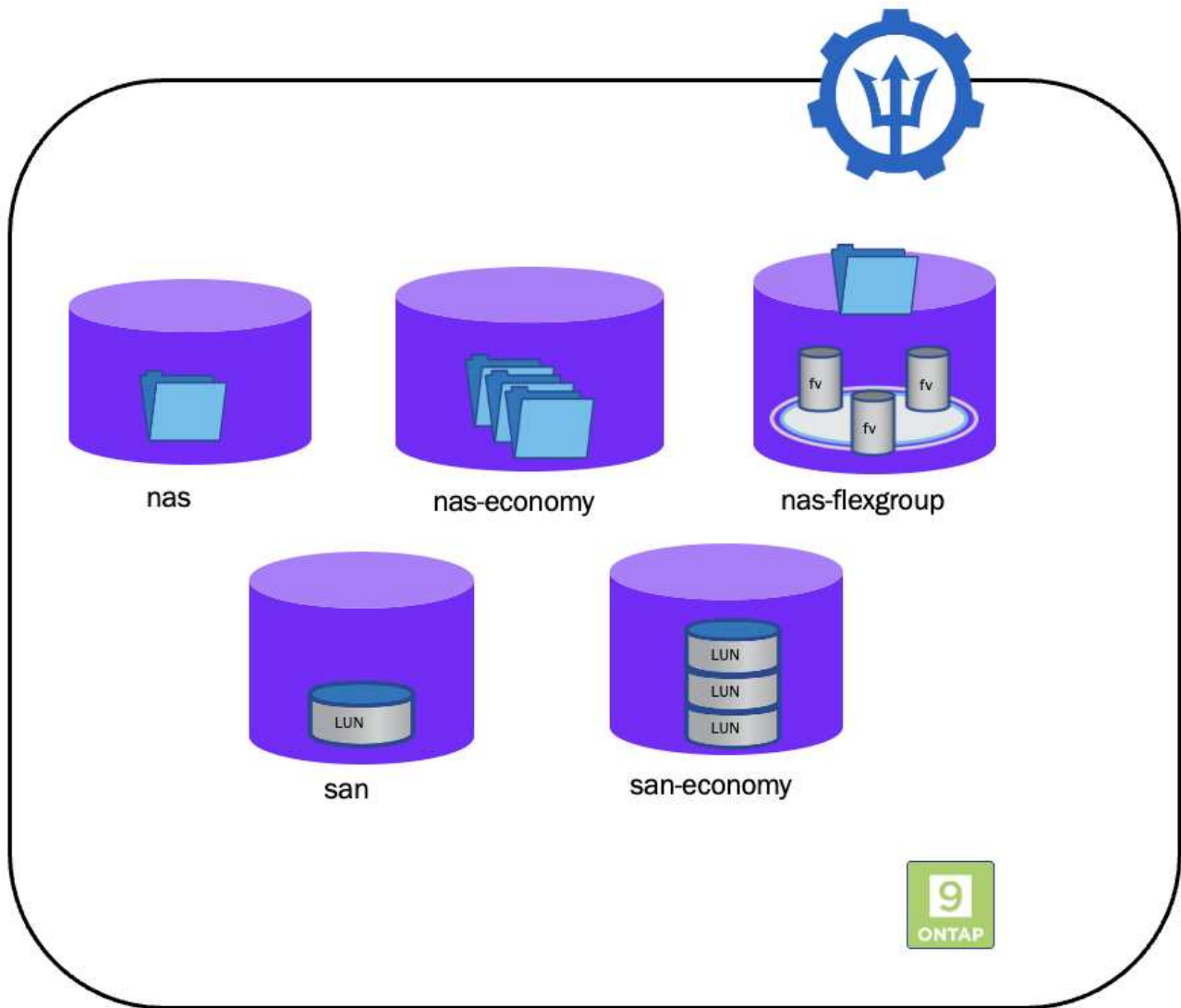
1. ontap-nas-flexgroup: プロビジョニングされた各 PV は完全な ONTAP FlexGroup であり、SVM に割り当てられたすべてのアグリゲートが使用されます。

サポートされているハードウェアモデルにおける**SAN**プロトコルの場合

1. ontap-san: 1 つの PVC が 1 つの PV にマッピングされ、これは専用の FlexVolume 上の LUN です。
2. ontap-san-economy: 1 つの PVC が、共有 flexvolume 上の LUN である 1 つの PV にマッピングされます。共有 flexvolume には複数の LUN を含めることができます（デフォルトは 100、flexvol ボリュームあたり 50～200 の LUN/PV に設定可能）。



仮想マシンのすべてのディスクをLUNとして、単一のボリューム内に配置できます。ただし、このドライバーはスナップショットとクローンはサポートしていますが、レプリケーションはサポートしていません。ストレージ管理者がレプリケーションを管理する場合、PV単位の粒度はありません。



Trident ドライバを通じて公開されている機能と、ストレージ管理者が適用する必要がある機能の完全なリストについては、Trident ドキュメントの ["ONTAP バックエンドドライバ"](#) セクションを参照してください。

Red Hat OpenShift クラスタに Trident をインストールしてストレージオブジェクトを作成する

Red Hat認定TridentオペレーターでTridentをインストールし、ONTAPおよびAmazon FSx for NetApp ONTAP用のストレージオブジェクトを作成して、コンテナおよびVMの動的ボリュームプロビジョニングを有効にします。必要に応じて、ブロックアクセス用のワーカーノードを準備します。

開始する前に

- OpenShift Virtualizationをインストールする前に、このページの手順を完了してください。OpenShift VirtualizationでVMテンプレート用のゴールデンイメージを作成するには、デフォルトのTridentベースのStorageClassとVolumeSnapshotClassが必要です。
- OpenShift VirtualizationをインストールしてからTridentを設定する前に、異なるストレージクラスで作成

されたゴールデンイメージをすべて削除してください。Tridentをデフォルトとして設定すると、OpenShift VirtualizationはTridentストレージを使用してゴールデンイメージを再作成します。

```
oc delete dv,VolumeSnapshot -n openshift-virtualization-os-images  
--selector=cdi.kubevirt.io/dataImportCron
```

ステップ1：Tridentをインストールする

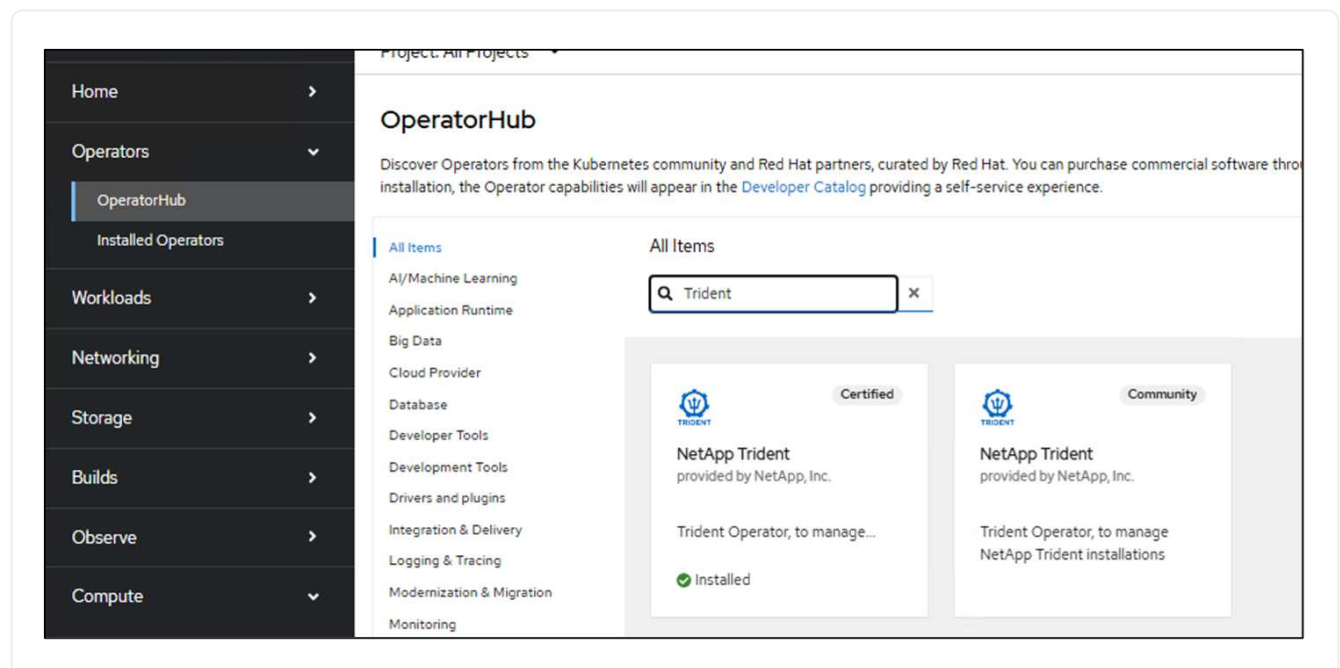
Red Hat認定Tridentオペレーターは、NetAppによってOpenShiftのオンプレミス、パブリッククラウド、およびROSAなどのマネージドサービスでサポートされています。Trident 25.02以降、オペレーターは、Amazon FSx for NetApp ONTAPを使用し、OpenShift Virtualization VMワークロードを実行する予定の場合に、iSCSI用のワーカーノードを準備することもできます。

その他のインストールオプションについては、"[Tridentのドキュメント](#)"を参照してください。

手順

1. **OperatorHub** で、**Certified NetApp Trident** を選択します。

例を表示



2. 「インストール」 ページで、最新バージョンを保持し、「インストール」を選択します。

例を表示

OperatorHub > Operator Installation

Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel * ⓘ

stable

Version *

25.2.1

25.2.1

25.2.0

Operator will be available in all namespaces.

☐ A specific namespace on the cluster
This mode is not supported by this Operator

Installed Namespace *

PR openshift-operators

Update approval * ⓘ

☒ Automatic

☐ Manual

Install Cancel

- オペレーターのインストール後、*オペレーターの表示*を選択し、Trident Orchestratorのインスタンスを作成します。

iSCSI 用にワーカー ノードを準備する場合は、YAML ビューに切り替えて `iscsi` を `nodePrep` に追加します。

例を表示

Create TridentOrchestrator

Create by completing the form. Default values may be provided by the Operator authors.

Configure via: ☐ Form view ☒ YAML view

```
1 kind: TridentOrchestrator
2 apiVersion: trident.netapp.io/v1
3 metadata:
4   name: trident
5 spec:
6   IPv6: false
7   debug: true
8   nodePrep:
9     - iscsi
10  imagePullSecrets: []
11  imageRegistry: ''
12  namespace: trident
13  silenceAutosupport: false
14
```

4. すべての Trident ポッドがクラスター内で実行されていることを確認してください。

例を表示

```
[root@localhost ~]# oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-controller-84cb9bff89-lkx6k 6/6     Running   0           16h
trident-node-linux-d88b9            2/2     Running   0           16h
trident-node-linux-ld4b8            2/2     Running   0           16h
trident-node-linux-mj5r8            2/2     Running   0           16h
trident-node-linux-mkmmmp           2/2     Running   0           16h
trident-node-linux-qhgr7            2/2     Running   0           16h
trident-node-linux-vt9tp            2/2     Running   0           16h
[root@localhost ~]#
```

5. iSCSIノードの準備を有効にした場合は、ワーカーノードにログインして、`iscsid`と`multipathd`がアクティブであり、`multipath.conf`にエントリがあることを確認してください。

例を表示

```
sh-5.1# systemctl status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled; preset: disabled)
   Active: active (running) since Fri 2025-04-25 00:23:49 UTC; 3 days ago
 TriggeredBy: ● iscsid.socket
    Docs: man:iscsid(8)
          man:iscsiuio(8)
          man:iscsiadm(8)
   Main PID: 74787 (iscsid)
    Status: "Ready to process requests"
     Tasks: 1 (limit: 410912)
  Memory: 1.8M
     CPU: 6ms
    CGroup: /system.slice/iscsid.service
            └─74787 /usr/sbin/iscsid -f

Apr 25 00:23:49 ocp11-worker1 systemd[1]: Starting Open-iSCSI...
Apr 25 00:23:49 ocp11-worker1 systemd[1]: Started Open-iSCSI.
sh-5.1#
```

例を表示

```
sh-5.1# systemctl status multipathd
● multipathd.service - Device-Mapper Multipath Device Controller
   Loaded: loaded (/usr/lib/systemd/system/multipathd.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-04-25 00:23:50 UTC; 3 days ago
 TriggeredBy: ● multipathd.socket
    Process: 74905 ExecStartPre=/sbin/modprobe -a scsi_dh_alua scsi_dh_emc scsi_dh_rdac dm-multipath (code=exited, status=0/SUCCESS)
    Process: 74906 ExecStartPre=/sbin/multipath -A (code=exited, status=0/SUCCESS)
   Main PID: 74907 (multipathd)
    Status: "up"
     Tasks: 7
  Memory: 18.3M
     CPU: 23.008s
    CGroup: /system.slice/multipathd.service
            └─74907 /sbin/multipathd -d -s

Apr 25 00:23:50 ocp11-worker1 systemd[1]: Starting Device-Mapper Multipath Device Controller...
Apr 25 00:23:50 ocp11-worker1 multipathd[74907]: -----start up-----
Apr 25 00:23:50 ocp11-worker1 multipathd[74907]: read /etc/multipath.conf
Apr 25 00:23:50 ocp11-worker1 multipathd[74907]: path checkers start up
Apr 25 00:23:50 ocp11-worker1 systemd[1]: Started Device-Mapper Multipath Device Controller.
sh-5.1#
```

例を表示

```
sh-5.1# cat /etc/multipath.conf
defaults {
    find_multipaths no
}
blacklist {
    device {
        product .*
        vendor  .*
    }
}
blacklist_exceptions {
    device {
        product LUN
        vendor  NETAPP
    }
}
sh-5.1#
```

ビデオデモンストレーション

以下のビデオでは、Red Hat認定のTridentオペレーターを使用してTridentをインストールする方法のデモンストレーションを示しています。

[OpenShift で認定されたTrident Operator を使用してTrident 25.02.1 をインストールする](#)

ステップ2：環境のストレージバックエンドと**StorageClass**設定ファイルを準備する

環境用のTridentBackendConfigとStorageClassの定義を作成します。環境内で複数のストレージ プロトコルを設定できます。使用するプロトコルごとにYAMLファイルを作成し、ブレースホルダーの値を具体的な設定の詳細に置き換えてください。



環境に応じて、下記のオンプレミスまたは ROSA セクションのいずれかを完了してから、ステップ 3 に進んでください。

オンプレミスOpenShiftクラスター

設定したいプロトコルごとにYAMLファイルを作成してください。以下のプロトコルの1つ以上を設定できます：NFS ベースのファイル ストレージ用の NAS、iSCSI ブロック ストレージ用の iSCSI、TCP 経由のハイパフォーマンス NVMe ブロック ストレージ用の NVMe/TCP、または Fibre Channel ブロック ストレージ用の FC。

各プロトコルのTridentBackendConfigとStorageClassを作成します。バックエンド構成には、Kubernetes Secret に保存されている認証情報が含まれます。

バックエンド構成の基本オプションに関する重要なガイドライン

1. バックエンド構成ファイルは、**ONTAP SVM** と 管理 **LIF** を参照します。+
2. ユーザー名とパスワードを使用して ONTAP で認証できます。クラスター管理者または vservers 管理者のユーザー名とパスワードを使用できます。+
3. ONTAP との認証にはクライアント証明書も使用できます。クライアント証明書、秘密鍵、および CA 証明書の生成方法の詳細については、"[Tridentドキュメントはこちら](#)"を参照してください。

NASエコノミードライバーのバックエンド構成オプションに関する基本ガイドライン

NASエコノミードライバの設定には、以下のパラメータを含めることができます。

1. **qtreesPerFlexvol** 値は50～300の範囲内である必要があります。デフォルト値は200です。+
2. **limitVolumePoolSize** この値は、ontap-nas-economy バックエンドで qtrees を使用する際にリクエストする最大 FlexVol サイズです。デフォルト値は設定されておらず、このパラメータはデフォルトでは適用されません。このパラメータを設定すると、現在の FlexVol が指定されたサイズに達したときに、Trident オペレーターが新しい FlexVol を作成します。このパラメータは、PV のプロビジョニングに使用される FlexVols のサイズを制限することで、クラスターの容量使用率を制御するのに役立ちます。+
3. **denyNewVolumePools** このパラメータは、ontap-nas-economy バックエンドが qtree を格納するための新しい FlexVol ボリュームを作成することを制限します。新しい PV のプロビジョニングには、既存の FlexVol のみが使用されます。

SANドライバのバックエンド構成オプションに関する重要なガイドライン

SANドライバの設定には、以下のパラメータを含めることができます。

1. **limitVolumeSize** この値は、SANバックエンドを使用する際に要求する最大ボリュームサイズです。デフォルト値は設定されておらず、このパラメータはデフォルトでは適用されません。このパラメータを設定すると、要求されたボリュームサイズがこの値を超えた場合に Trident オペレーターはプロビジョニングに失敗します。このパラメータを使用すると、LUN 用に管理するボリュームの最大サイズを制限できます。+
2. **lunsPerFlexvol** FlexVol あたりの最大 LUN 数。範囲は [50, 200] で、デフォルト値は 100 です。このパラメータを設定すると、現在の FlexVol が指定された数の LUN に達したときに、Trident オペレーターが新しい FlexVol を作成します。このパラメータを使用すると、PV のプロビジョニングに使用される FlexVol あたりの LUN 数を制限することで、クラスターの容量使用率を制御できます。

SANエコノミードライバの設定には、以下のパラメータを含めることができます。

1. **limitVolumePoolSize** この値は、SANエコノミーバックエンドを使用する際に要求する最大FlexVolサイズです。デフォルト値は設定されておらず、このパラメータはデフォルトでは適用されません。このパラメータを設定すると、Tridentオペレータは、flexvol内にLUNを作成するために作成できるflexvolの最大サイズを制限します。このパラメータは、PVのプロビジョニングに使用されるFlexVolsのサイズを制限することで、クラスターの容量使用率を制御するのに役立ちます。+
2. **denyNewVolumePools** このパラメータは、SAN エコノミー バックエンドが LUN を格納するための新しい FlexVol ボリュームを作成することを制限します。新しい PV のプロビジョニングには、既存の FlexVol のみを使用されます。



各プロトコル用に提供されているサンプルYAMLファイルでは、storagePrefixが指定されていない場合は、デフォルトの接頭辞「trident」が使用されます。ontap-nas-economyと24文字以上のstoragePrefixを使用する場合、qtreeにはストレージプレフィックスは埋め込まれませんが、ボリューム名には含まれます。

ONTAP のシステムマネージャまたは CLI から直接ボリュームを作成する際に、すべてのオプションパラメータに値を指定しない場合、ボリュームのプロビジョニング中に以下に示すようないくつかのデフォルト値が自動的に設定されます。

```
NSOL-NetApp-A70-T19U05:> volume create -vserver oc-test-automation -volume test_ontap_default -aggregate NSOL_NetApp_A70_T19U05a_NVME_SSD_1 -size 10GB
[Job 22836] Job succeeded: Successful
```

```
NSOL-NetApp-A70-T19U05:> volume show -vserver oc-test-automation -volume test_ontap_default -fields snapshot-policy,qos-adaptive-policy-group,snapshot-reserve-available,tiering-policy,unix-permissions,snapdir-access,space-guarantee
vserver      volume      unix-permissions space-guarantee snapdir-access snapshot-policy qos-adaptive-policy-group snapshot-reserve-available tiering-policy
oc-test-automation test_ontap_default ---rwxr-xr-x      none          true          default      -          511.8MB      none
```

以下は、Trident バックエンド設定を使用して作成されたボリュームの例です。デフォルトセクションでは、名前テンプレート以外に明示的に設定されているオプションはありません。

```
oc describe tbc -n trident tbc-nas
### output truncated for brevity
Spec:
  Backend Name:  tbc-nas
  Credentials:
    Name:  tbc-nas-secret
  Defaults: # There are no defaults set except for the volume name
template.
  Name Template:      {{ .config.StoragePrefix }}_{{
.volume.Namespace }}_{{ .volume.RequestName }}
  Management LIF:     10.192.102.50
  Storage Driver Name: ontap-nas
  Storage Prefix:     testautomation
  Svm:                oc-test-automation
```

このバックエンド構成を使用してボリュームを作成すると、バックエンド構成ファイルで設定されていないオプションは、以下に示すようにデフォルト値に設定されます。

```
NSOL-NetApp-A70-T19U05:> volume show -vserver oc-test-automation -volume testautomation_default_test_trident_defaults_91653 -fields snapshot-policy,qos-adaptive-policy-group,snapshot-reserve-available,tiering-policy,unix-permissions,snapdir-access,space-guarantee
vserver      volume      unix-permissions space-guarantee snapdir-access snapshot-policy qos-adaptive-policy-group snapshot-reserve-available tiering-policy
oc-test-automation testautomation_default_test_trident_defaults_91653 ---rwxrwxrwx      none          false         none         -          0B          none
```

ただし、Tridentバックエンド構成のデフォルトセクションで値を設定すると、それらのパラメータに特定の値を持つボリュームを作成できます。次の例では、snapshotPolicyはデフォルトポリシーに設定されています（明示的に何も設定しない場合はnoneになります）。また、snapshotDirはバックエンド設定YAMLのdefaultセクションでtrueに設定されています。これらの設定により、バックエンドボリュームがこれらの値を取得することがわかります。（snapdir-accessはtrue、snapshot-policyはdefault）

```
[root@00-50-56-b5-01-5b pvcs-br]# oc describe tbc -n trident tbc-nas
Name:          tbc-nas
Namespace:     trident
Labels:        <none>
Annotations:   <none>
API Version:   trident.netapp.io/v1
Kind:          TridentBackendConfig
Metadata:
  Creation Timestamp: 2026-03-30T20:06:59Z
  Finalizers:
    trident.netapp.io
  Generation: 3
  Resource Version: 81358056
  UID: 4dc24339-ff66-4345-acc1-37ec73d14140
Spec:
  Backend Name: tbc-nas
  Credentials:
    Name: tbc-nas-secret
  Defaults:
    Name Template: {{ .config.StoragePrefix }}_{{ .volume.Namespace }}_{{ .volume.RequestName }}
    Snapshot Dir: true
    Snapshot Policy: default
    Management LIF: 10.192.102.50
    Storage Driver Name: ontap-nas
    Storage Prefix: testautomation
    Svm: oc-test-automation
    Version: 1
Status:
  Backend Info:
    Backend Name: tbc-nas
    Backend UUID: 210cc722-b394-489d-8e9e-49018d3e8710
    Deletion Policy: delete
    Last Operation Status: Success
    Message: Backend 'tbc-nas' updated
    Phase: Bound
```

```
NSOL-NetApp-A70-T19U05:> volume show -vserver oc-test-automation -volume testautomation_default_test_trident_set_defaults_7567a -fields snapshot-policy,qos-adaptive-policy-group,snapshot-reserve-available,tiering-policy,unix-permissions,snapdir-access,space-guarantee
vserver      volume                                     unix-permissions space-guarantee snapdir-access snapshot-policy qos-adaptive-policy-group snapshot-reserve-availab
le tiering-policy
-----
oc-test-automation testautomation_default_test_trident_set_defaults_7567a ---rwxrwxrwx none true default - 269.5MB
none
```

```
NSOL-NetApp-A70-T19U05:> volume show -vserver oc-test-automation -volume testautomation_default_test_trident_set_defaults_7567a -fields snapshot-policy,snapdir-access
vserver      volume                                     snapdir-access snapshot-policy
oc-test-automation testautomation_default_test_trident_set_defaults_7567a true default
```

その他の追加パラメータについては、Trident のドキュメントを参照してください：

1. "Trident NASバックエンド構成オプション" +
2. "Trident SANバックエンド構成オプション"

NAS

TridentBackendConfigとStorageClassを作成して、ONTAP NASでNFSベースの永続ストレージ プロビジョニングを有効にします。バックエンド構成には、Kubernetes Secretに保存されている認証情報が含まれており、ONTAP SVMと管理LIFを参照します。

ユーザー名とパスワード認証を使用したバックエンドシークレットとバックエンド設定ファイルのサンプル（`tbc-nas.yaml`として保存）：

```
# tbc-nas.yaml
apiVersion: v1
kind: Secret
metadata:
  name: tbc-nas-secret
type: Opaque
stringData:
  username: <cluster admin username>
  password: <cluster admin password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-nas
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: <ONTAP management LIF>
  backendName: tbc-nas
  svm: zoneb #<replace with your SVM name>
  storagePrefix: testzoneb #<replace with your prefix>
  defaults:
    nameTemplate: "{{ .config.StoragePrefix }}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"
  credentials:
    name: tbc-nas-secret
```

クライアント証明書認証を使用するバックエンドシークレットとバックエンド構成ファイルのサンプル（`tbc-nas.yaml`として保存）：

```
# tbc-nas.yaml
apiVersion: v1
kind: Secret
metadata:
  name: ontap-nas-secret
  namespace: trident
type: Opaque
stringData:
  clientPrivateKey: <client private>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: <ONTAP management LIF>
  backendName: tbc-nas
  svm: zoneb #<replace with your SVM name>
  storagePrefix: testzoneb #<replace with your prefix>
  credentials:
    name: ontap-nas-secret
    clientCertificate: <client certificate>
```

StorageClass定義（`sc-nas.yaml`として保存）：

```
# sc-nas.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-nas
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
allowVolumeExpansion: true
```

NAS エコノミー

TridentBackendConfig と StorageClass を ONTAP NAS エコノミードライバー用に作成し、qtree を使用した NFS ベースの永続的なストレージ プロビジョニングを有効にします。バックエンド構成に

は、Kubernetes Secret に保存されている認証情報が含まれており、ONTAP SVM と管理 LIF を参照します。

ユーザー名とパスワード認証を使用したバックエンドシークレットとバックエンド設定ファイルのサンプル（tbc-nas-economy.yaml として保存）：

```
# tbc-nas-economy.yaml
apiVersion: v1
kind: Secret
metadata:
  name: tbc-nas-economy-secret
type: Opaque
stringData:
  username: <cluster admin username>
  password: <cluster admin password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-nas-economy
spec:
  version: 1
  storageDriverName: ontap-nas-economy
  managementLIF: <ONTAP management LIF>
  backendName: tbc-nas-eco
  svm: <replace with your SVM name>
  storagePrefix: <replace with your prefix>
  defaults:
    nameTemplate: "{{ .config.StoragePrefix }}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"
  credentials:
    name: tbc-nas-economy-secret
```

StorageClass定義（`sc-nas-economy.yaml`として保存）：

```
# sc-nas-economy.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-nas-economy
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas-economy"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
allowVolumeExpansion: true
```



ontap-nas-economyドライバのバックエンド構成ファイルに含めることができる追加パラメータの詳細については、上記の「NASエコノミードライバのバックエンド構成オプションに関する基本ガイドライン」セクションを参照してください。

iSCSI SAN

iSCSI ベースのブロックストレージ プロビジョニングを有効にするために、ONTAP SAN with iSCSI 用の TridentBackendConfig と StorageClass を作成します。バックエンド構成では、ontap-san ドライバを使用し、Kubernetes Secret に格納された認証情報が含まれます。sanType パラメータは、省略した場合、デフォルトで iSCSI プロトコルが使用されます。

ユーザー名とパスワードによる認証を使用するバックエンドシークレットとバックエンド構成ファイル（保存先： tbc-iscsi.yaml）：

```
# tbc-iscsi.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-iscsi-secret
type: Opaque
stringData:
  username: <cluster admin username>
  password: <cluster admin password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: ontap-iscsi
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <management LIF>
  backendName: ontap-iscsi
  svm: <SVM name>
  credentials:
    name: backend-tbc-ontap-iscsi-secret
```

クライアント証明書認証を使用するバックエンドシークレットとバックエンド構成ファイル（`tbc-iscsi.yaml`として保存）：

```
# tbc-iscsi.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-iscsi-secret
type: Opaque
stringData:
  clientPrivateKey: <client private key>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: ontap-iscsi
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <management LIF>
  backendName: ontap-iscsi
  svm: <SVM name>
  credentials:
    name: backend-tbc-ontap-iscsi-secret
    clientCertificate: <client certificate>
```

StorageClass定義（`sc-iscsi.yaml`として保存）：

```
# sc-iscsi.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-iscsi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  fsType: ext4
  snapshots: "true"
allowVolumeExpansion: true
```

NVMe/TCP SAN

TridentBackendConfigとStorageClassを作成して、TCP経由のNVMeを使用したONTAP SANでハイパフォーマンスブロックストレージ プロビジョニングを有効にします。バックエンド構成では、NVMe/TCP トランスポート用に最適化されたontap-sanドライバを使用し、Kubernetes Secretに格納された認証情報が含まれています。

ユーザー名とパスワードによる認証を使用するバックエンドシークレットとバックエンド構成ファイル（保存先： tbc-nvme.yaml）：

```
# tbc-nvme.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-nvme-secret
type: Opaque
stringData:
  username: <cluster admin username>
  password: <cluster admin password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nvme
spec:
  version: 1
  storageDriverName: ontap-san
  sanType: nvme
  managementLIF: <ONTAP management LIF>
  backendName: backend-tbc-ontap-nvme
  svm: <SVM name>
  credentials:
    name: backend-tbc-ontap-nvme-secret
```

クライアント証明書認証を使用するバックエンドシークレットとバックエンド構成ファイル（`tbc-nvme.yaml`として保存）：

```
# tbc-nvme.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-nvme-secret
type: Opaque
stringData:
  clientPrivateKey: <client private key>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nvme
spec:
  version: 1
  storageDriverName: ontap-san
  sanType: nvme
  managementLIF: <ONTAP management LIF>
  backendName: backend-tbc-ontap-nvme
  svm: <SVM name>
  credentials:
    name: backend-tbc-ontap-nvme-secret
    clientCertificate: <client certificate>
```

StorageClass定義（`sc-nvme.yaml`として保存）：

```
# sc-nvme.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-nvme
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  fsType: ext4
  snapshots: "true"
allowVolumeExpansion: true
```



ReadWriteMany (RWX) ボリュームをNVMeプロトコルで使用する予定がある場合、Kubernetesクラスタでは、単一のRWX NVMeボリュームを64ノード以上でマウントすることはできません。この制限の詳細、およびTrident 26.02で導入されたSuper-subsystemモデルとTrident 26.02以前のPer-volume subsystemモデルについては、Tridentのドキュメントを参照してください：<https://docs.netapp.com/us-en/trident/trident-use/ontap-san-prep.html#nvme-tcp-considerations>。

FC SAN

TridentBackendConfigとStorageClassを作成して、Fibre ChannelでONTAP SANのFCベースのブロックストレージ プロビジョニングを有効にします。バックエンド構成では、FCPプロトコルを指定したontap-sanドライバを使用し、Kubernetes Secretに格納された認証情報を含みます。

ユーザー名/パスワード認証方式またはクライアント証明書認証方式を使用して、ONTAP で認証します。クライアント証明書、秘密鍵、および CA 証明書の生成方法の詳細については、"[Tridentドキュメントはこちら](#)"を参照してください。両方の方法の例を以下に示します。

例を表示

バックエンドシークレットとバックエンド構成ファイルを使用した ONTAP ユーザー名とパスワードによる認証（`tbc-fc.yaml`として保存）：

```
# tbc-fc.yaml
apiVersion: v1
kind: Secret
metadata:
  name: tbc-fc-secret
type: Opaque
stringData:
  username: <cluster admin username>
  password: <cluster admin password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-fc
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <ONTAP management LIF>
  backendName: tbc-fc
  svm: openshift-fc #<replace with your SVM name>
  sanType: fcp
  storagePrefix: demofc #<replace with your prefix>
  defaults:
    nameTemplate: "{{ .config.StoragePrefix }}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"
  credentials:
    name: tbc-fc-secret
```

クライアント証明書認証を使用するバックエンドシークレットとバックエンド構成ファイル（`tbc-fc.yaml`として保存）：

```
# tbc-fc.yaml
apiVersion: v1
kind: Secret
metadata:
  name: tbc-fc-secret
type: Opaque
stringData:
  clientPrivateKey: <client private key>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-fc
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <ONTAP management LIF>
  backendName: tbc-fc
  svm: openshift-fc #<replace with your SVM name>
  sanType: fcp
  storagePrefix: demofc #<replace with your prefix>
  defaults:
    nameTemplate: "{{ .config.StoragePrefix }}_{{ .volume.Namespace
  }}"_{{ .volume.RequestName }}"
  credentials:
    name: tbc-fc-secret
    clientCertificate: <client certificate>
```

```
# sc-fc.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-fc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  fsType: ext4
  snapshots: "true"
allowVolumeExpansion: true
```

TridentBackendConfig と StorageClass を作成して、iSCSI を使用した ONTAP SAN economy で iSCSI ベースのブロックストレージ プロビジョニングを有効にします。バックエンド構成では、iSCSI プロトコルを使用する ontap-san-economy ドライバーを使用し、Kubernetes Secret に格納された認証情報が含まれます。

クライアント証明書認証を使用するバックエンドシークレットとバックエンド構成ファイル（`tbc-san-eco.yaml`として保存）：

```
# tbc-san-eco.yaml
---
apiVersion: v1
kind: Secret
metadata:
  name: ontap-san-cert-auth
  namespace: trident
type: Opaque
stringData:
  clientPrivateKey: <client private key>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-san-eco-cert
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-san-economy
  managementLIF: "<ONTAP management LIF>"
  backendName: tbc_san_eco
  svm: <replace with your SVM name>
  storagePrefix: <replace with your prefix>
  credentials:
    name: ontap-san-cert-auth
    clientCertificate: <client certificate>
```

```
# sc-san-eco.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-san-eco
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san-economy"
  provisioningType: "thin"
  snapshots: "true"
```

SANドライバおよびNASドライバでサポートされているアクセスモードとボリュームモードに関する情報、およびYAMLファイルの追加サンプルについては、Tridentのドキュメントを参照してください。

1. ["NASドライバを使用したサンプル"](#)
2. ["SANドライバを使用したサンプル"](#)

Amazon FSx for NetApp ONTAPを使用したROSAクラスター

設定したいプロトコルごとにYAMLファイルを作成してください。以下のプロトコルのいずれか、または両方を設定できます：NFSベースのファイルストレージにはNAS、ブロックストレージにはiSCSIを使用します。

NAS

TridentBackendConfigとStorageClassを作成して、Amazon FSx for NetApp ONTAPとONTAP NASを使用し、ROSAクラスター上でNFSベースの永続ストレージ プロビジョニングを有効にします。バックエンド構成では、管理LIFとデータLIF用のAmazon FSx for NetApp ONTAP DNS名を使用し、trident名前空間のKubernetes Secretに格納されている認証情報が含まれます。

バックエンドシークレットとバックエンド設定ファイル（`tbc-fsx-nas.yaml`として保存）：

```
# tbc-fsx-nas.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-fsx-ontap-nas-secret
type: Opaque
stringData:
  username: <FSx for ONTAP, for example fsxadmin>
  password: <FSx for ONTAP password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-fsx-ontap-nas
spec:
  version: 1
  backendName: fsx-ontap
  storageDriverName: ontap-nas
  managementLIF: <Management DNS name>
  dataLIF: <NFS DNS name>
  svm: <SVM NAME>
  credentials:
    name: backend-fsx-ontap-nas-secret
```

StorageClass定義（`sc-fsx-nas.yaml`として保存）：

```
# sc-fsx-nas.yaml (storage class name is trident-csi)
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: trident-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "ext4"
allowVolumeExpansion: true
reclaimPolicy: Retain
```

iSCSI

TridentBackendConfigとStorageClassを作成して、Amazon FSx for NetApp ONTAP with ONTAP SANでROSAクラスター上のiSCSIベースのブロックストレージプロビジョニングを有効にします。バックエンド構成では、ontap-sanドライバを使用し、Kubernetes Secretに格納された認証情報が含まれます。ワーカーノードがiSCSIアクセス用に準備されていることを確認してください。

バックエンドシークレットとバックエンド設定ファイル（`tbc-fsx-iscsi.yaml`として保存）：

```
# tbc-fsx-iscsi.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-fsx-iscsi-secret
type: Opaque
stringData:
  username: <FSx for ONTAP, for example fsxadmin>
  password: <FSx for ONTAP password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: fsx-iscsi
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <Management DNS name>
  backendName: fsx-iscsi
  svm: <SVM name>
  credentials:
    name: backend-tbc-fsx-iscsi-secret
```

StorageClass定義（`sc-fsx-iscsi.yaml`として保存）：

```
# sc-fsx-iscsi.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-fsx-iscsi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  fsType: ext4
  snapshots: "true"
allowVolumeExpansion: true
```

ステップ3：VolumeSnapshotClass設定ファイルを作成する

オンプレミスとROSAの両方の導入に対してVolumeSnapshotClass定義を作成します。この構成により、永続ボリュームに対するスナップショットベースの操作が可能になります。

VolumeSnapshotClass定義（`snapshot-class.yaml`として保存）：

```
# snapshot-class.yaml
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: trident-snapshotclass
driver: csi.trident.netapp.io
deletionPolicy: Retain
```

ステップ4：構成ファイルをクラスタに適用する

前の手順で作成した構成ファイルをOpenShiftクラスタに適用します。

手順

1. 設定したプロトコルごとにTridentBackendConfigとStorageClassファイルを適用します。

オンプレミス クラスタの場合：

```
oc create -f tbc-nas.yaml -n trident
oc create -f sc-nas.yaml

oc create -f tbc-nas-economy.yaml -n trident
oc create -f sc-nas-economy.yaml

oc create -f tbc-iscsi.yaml -n trident
oc create -f sc-iscsi.yaml

oc create -f tbc-nvme.yaml -n trident
oc create -f sc-nvme.yaml

oc create -f tbc-fc.yaml -n trident
oc create -f sc-fc.yaml
```

ROSAクラスタの場合：

```
oc create -f tbc-fsx-nas.yaml -n trident
oc create -f sc-fsx-nas.yaml

oc create -f tbc-fsx-iscsi.yaml -n trident
oc create -f sc-fsx-iscsi.yaml
```

2. VolumeSnapshotClass構成を適用します。

```
oc create -f snapshot-class.yaml
```

3. リソースが正常に作成されたことを確認してください。

TridentBackendConfigオブジェクトを確認：

```
oc get tbc -n trident
```

StorageClassオブジェクトを確認します：

```
oc get storageclass
```

VolumeSnapshotClassを確認：

```
oc get volumesnapshotclass
```

ステップ5：デフォルトのTridentストレージおよびスナップショットクラスを設定する

OpenShift クラスターで Trident の StorageClass と VolumeSnapshotClass をデフォルトに設定します。これは、OpenShift Virtualization が VM テンプレート用のゴールデンイメージソースを作成するために必要です。

手順

1. デフォルトのTrident StorageClassを設定します。

Trident-backed StorageClassをクラスターのデフォルトとして設定すると、ストレージクラスが指定されていない場合にPersistentVolumeClaimsが自動的にそれを使用します。2つのアノテーションを設定する必要があります（1つはクラスター全体のデフォルト用、もう1つはOpenShift Virtualization専用）。

- a. クラスター全体のデフォルトのStorageClassアノテーションを設定します。

1つのStorageClassのみがデフォルトとして設定されていることを確認してください。別のStorageClassが既にデフォルトに設定されている場合は、そのアノテーションを `false` に設定します。

コンソールで、注釈を編集します：

```
storageclass.kubernetes.io/is-default-class: "true"
```

CLIから：

```
kubectl patch storageclass <storage class name> -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

- b. OpenShift Virtualization固有のデフォルトアノテーションを設定します。

OpenShift Virtualization では、クラスターの一般的な `is-default-class` 注釈よりも優先される特定の注釈が使用されます。別のStorageClassが既にデフォルトに設定されている場合は、その注釈を `false` に設定します。

コンソールで、注釈を編集します：

```
storageclass.kubevirt.io/is-default-virt-class: "true"
```

CLIから：

```
kubectl patch storageclass <storage class name> -p '{"metadata": {"annotations":{"storageclass.kubevirt.io/is-default-virt-class":"true"}}}'
```

2. デフォルトのTrident VolumeSnapshotClassを設定します。

TridentベースのVolumeSnapshotClassをクラスタのデフォルトとして設定し、永続ボリュームのスナップショットベースの操作を有効にします。これにより、スナップショットクラスが指定されていない場合、VolumeSnapshotsは自動的にTrident CSIドライバを使用し、OpenShift Virtualizationがゴールデンイメージのスナップショットを作成できるようになります。

1つのVolumeSnapshotClassのみがデフォルトとして設定されていることを確認してください。別のVolumeSnapshotClassが既にデフォルトに設定されている場合は、そのアノテーションをfalseに設定してください。

コンソールで、注釈を編集します：

```
snapshot.storage.kubernetes.io/is-default-class: "true"
```

CLIから：

```
oc patch volumesnapshotclass <snapshot class name> --type=merge -p
'{"metadata":{"annotations":{"snapshot.storage.kubernetes.io/is-default-
class":"true"}}}'
```

ONTAPを使用した Red Hat OpenShift Virtualization の導入要件

OpenShift 仮想化を ONTAP ストレージシステムにデプロイするための前提条件と追加の詳細を確認してください。

前提条件

- RHCOS ワーカーノードを備えたベアメタル インフラストラクチャにインストールされた Red Hat OpenShift クラスター (バージョン 4.6 以降)
- VMのHAを維持するためにマシンヘルスチェックを導入する
- 正しいプロトコルで設定された SVM を備えたNetApp ONTAPクラスター。
- OpenShift クラスタにインストールされたTrident
- Tridentバックエンド構成が作成されました
- Tridentをプロビジョナーとして OpenShift クラスター上に構成された StorageClass
- TridentをプロビジョナーとしてOpenShiftクラスタに設定されたTridentのVolumeSnapshotClass
- Trident StorageClassが OpenShift クラスターまたは OpenShift Virtualization のデフォルトとして設定されています。

上記のTridentの前提条件については、["Trident設置部"](#)詳細については。

- Red Hat OpenShift クラスターへのクラスター管理者アクセス
- NetApp ONTAPクラスタへの管理者アクセス

- tridentctl および oc ツールがインストールされ、\$PATH に追加された管理ワークステーション

OpenShift Virtualization は OpenShift クラスターにインストールされたオペレーターによって管理されるため、メモリ、CPU、ストレージに追加のオーバーヘッドが発生します。クラスターのハードウェア要件を計画するには、この点を考慮する必要があります。詳細については、ドキュメント "[ここをクリックしてください。](#)" を参照してください。

オプションで、ノード配置ルールを設定することにより、OpenShift Virtualization のオペレーター、コントローラー、および VM をホストする OpenShift クラスターノードのサブセットを指定することもできます。OpenShift Virtualization のノード配置ルールを設定するには、ドキュメント "[ここをクリックしてください。](#)" を参照してください。

OpenShift Virtualization を支えるストレージについては、NetApp、専用の SVM によって支えられている特定の Trident バックエンドからストレージを要求する専用の StorageClass を用意することを推奨しています。これにより、OpenShift クラスター上の VM ベースのワークロードに提供されるデータに関して、マルチテナントのレベルが維持されます。

NetApp Trident CSI、ストレージプロファイル、OpenShift Virtualization のボリュームアクセスモード

OpenShift Virtualization では、ブートソースとは、テンプレートから仮想マシン (VM) をプロビジョニングするために使用されるディスクイメージのことです。これらは通常、openshift-virtualization-os-images 名前空間内の DataVolumes または永続ボリューム要求 (PVC) として保存されます。OpenShift Virtualization が正しく機能するためには、オペレータを使用して OpenShift Virtualization をインストールする前に、必須のストレージ構成タスクが 2 つあります。

1. クラスターのデフォルトの StorageClass を設定する必要があります。設定しない場合、OpenShift Virtualization はブートソースイメージを自動的にインポートできません。
2. ストレージプロバイダが CDI によって認識されない場合は、ストレージプロファイルを設定する必要があります。ストレージプロファイルは、関連付けられている StorageClass に基づいて推奨されるストレージ設定を提供します。StorageProfile の重要な機能の 1 つは、StorageClass のクローニングストラテジを定義することです。これにより、ボリュームのクローニング方法 (スナップショット、CSI クローン、ホスト支援コピーの使用など) が決定され、基盤となるストレージベンダーの機能に基づいてパフォーマンスとリソースの使用状況が最適化されます。+ Trident CSI ドライバーは CDI によって認識され、クラスター内に作成されたすべての Trident StorageClass に対してデフォルトのストレージプロファイルが自動的に作成されます。このデフォルトのストレージプロファイルのクローニングストラテジはスナップショットに設定されています。これは、Trident CSI ドライバーに推奨されるクローニングストラテジです。OpenShift Virtualization をインストールする前に、Trident 用の VolumeSnapshotClass と StorageClass を作成し、その StorageClass をデフォルトとして設定する必要があります。これにより、ブートソースイメージが openshift-virtualization-os-images ネームスペースで VolumeSnapshots として確実に利用可能になり、エラーが発生せずに VM を効率的にプロビジョニングできるようになります。

特定のサポートされているオペレーティングシステム (RHEL、Fedora、CentOS など) の場合、OpenShift 仮想化はブートソースを自動的に提供および更新します。これらは OpenShift 仮想化のインストール時にダウンロードされ、自動的に管理されて最新の OS バージョンに更新されます。OS が自動的に提供されない場合、管理者はカスタムイメージを手動で準備してアタッチする必要があります。CSI として NetApp Trident を使用すると、次の追加メリットが得られます：

1. ブートソースイメージは、openshift-virtualization-os-images 名前空間のスナップショットとして利用可能になります。仮想マシンがプロビジョニングされる際、ブートイメージはスナップショットからクローンされるため、仮想マシンの起動が大幅に高速化されます。
2. 各 Trident StorageClass に対してストレージプロファイルが自動的に作成され、クローン戦略がスナップ

ショットに設定された状態で自動的に構成されます。アクセスモードは `ReadWriteMany` に設定されています。手動設定は不要です。

クラスタに OpenShift Virtualization をインストールする前に、Trident StorageClass と Trident VolumeSnapshotClass を作成し、それらをデフォルトとして設定することが非常に重要です。これにより、ブートソースイメージが `openshift-virtualization-os-images` ネームスペースで VolumeSnapshots として利用可能になり、エラーを発生させることなく効率的に VM をプロビジョニングできるようになります。

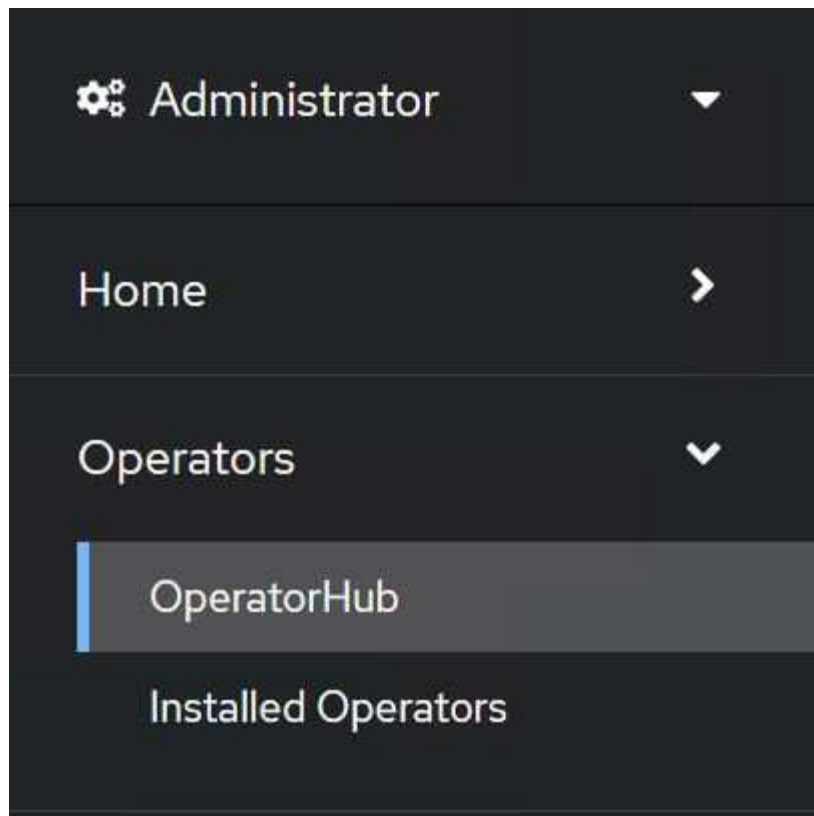
Trident CSIを使用するもう1つのメリットは次のとおりです：OpenShift Virtualizationでは、VMのライブ マイグレーションを実行するために、VMのブート ディスクにRWXアクセス モードが必要です。Tridentは、NAS プロトコルを使用してプロビジョニングされたボリュームだけでなく、SANプロトコルを使用してプロビジョニングされたボリューム（volumeModeがFileSystemではなくBlockに設定されている場合）に対しても、**ReadWriteMany**アクセス モードをサポートしています。ブート ディスクにはどちらのプロトコルも使用できます。Storage profileオブジェクトはStorageClassの設定に基づいて自動的に構成され、VMのプロビジョニング時にブート ディスクのアクセス モードが適切に設定されます。

クラスターのデフォルトとして Trident StorageClass を設定した後、異なるパラメータを持つ Trident 用の追加の StorageClasses を作成できます。これらは、適切なストレージプロファイルが作成された状態で OpenShift Virtualization によって自動的に認識されます。その後、VM 定義において VM のブートディスクおよびデータディスクに使用する StorageClass を指定できます。

OpenShiftクラスタへのOpenShift Virtualizationのインストールは、OpenShift OperatorHubで提供されているOpenShift Virtualizationオペレータを使用する簡単なプロセスです。OpenShift Virtualizationのインストール手順の詳細については、ドキュメント"[ここをクリックしてください](#)。"を参照してください。

Red Hat OpenShift ベアメタル クラスターに OpenShift Virtualization をインストールします。この手順には、cluster-admin アクセスでのログイン、OperatorHub への移動、OpenShift Virtualization オペレーターのインストールが含まれます。

1. cluster-admin アクセスを使用して Red Hat OpenShift ベアメタル クラスターにログインします。
2. 「パースペクティブ」ドロップダウンから「管理者」を選択します。
3. 「Operators」>「OperatorHub」に移動し、「OpenShift Virtualization」を検索します。



4. OpenShift Virtualization タイルを選択し、「インストール」をクリックします。

**OpenShift Virtualization**
2.6.2 provided by Red Hat

×

Install

Latest version
2.6.2

Capability level

- ☒ Basic Install
- ☒ Seamless Upgrades
- ☒ Full Lifecycle
- ☐ Deep Insights
- ☐ Auto Pilot

Provider type
Red Hat

Provider
Red Hat

Requirements

Your cluster must be installed on bare metal infrastructure with Red Hat Enterprise Linux CoreOS workers.

Details

OpenShift Virtualization extends Red Hat OpenShift Container Platform, allowing you to host and manage virtualized workloads on the same platform as container-based workloads. From the OpenShift Container Platform web console, you can import a VMware virtual machine from vSphere, create new or clone existing VMs, perform live migrations between nodes, and more. You can use OpenShift Virtualization to manage both Linux and Windows VMs.

The technology behind OpenShift Virtualization is developed in the [KubeVirt](#) open source community. The KubeVirt project extends [Kubernetes](#) by adding additional virtualization resource types through [Custom Resource Definitions](#) (CRDs). Administrators can use Custom Resource Definitions to manage [VirtualMachine](#) resources alongside all other resources that Kubernetes provides.

5. Install Operator 画面で、すべてのパラメータをデフォルトのままにして、「インストール」をクリックします。

Update channel *

- ☐ 2.1
- ☐ 2.2
- ☐ 2.3
- ☐ 2.4
- ☒ stable

Installation mode *

- ☐ All namespaces on the cluster (default)
This mode is not supported by this Operator
- ☒ A specific namespace on the cluster
Operator will be available in a single Namespace only.

Installed Namespace *

- ☒ Operator recommended Namespace: **PR** openshift-cnv

i Namespace creation

Namespace **openshift-cnv** does not exist and will be created.

- ☐ Select a Namespace

Approval strategy *

- ☒ Automatic
- ☐ Manual

Install

Cancel



OpenShift Virtualization
provided by Red Hat

Provided APIs



OpenShift
Virtualization
Deployment

i Required

Represents the deployment of
OpenShift Virtualization

6. オペレータのインストールが完了するまで待ちます。



OpenShift Virtualization
2.6.2 provided by Red Hat



Installing Operator

The Operator is being installed. This may take a few minutes.

[View installed Operators in Namespace openshift-cnv](#)

7. オペレータのインストールが完了したら、「Create HyperConverged」をクリックします。



Installed operator - operand required

The Operator has installed successfully. Create the required custom resource to be able to use this Operator.



HyperConverged



Required

Creates and maintains an OpenShift Virtualization Deployment

Create HyperConverged

[View installed Operators in Namespace openshift-cnv](#)

8. 「HyperConverged の作成」画面で、すべてのデフォルト パラメータを受け入れて「作成」をクリックします。この手順では、OpenShift Virtualization のインストールが開始されます。

Name *

kubevirt-hyperconverged

Labels

app=frontend

Infra >

infra HyperConvergedConfig influences the pod configuration (currently only placement) for all the infra components needed on the virtualization enabled cluster but not necessarily directly on each node running VMs/VMLs.

Workloads >

workloads HyperConvergedConfig influences the pod configuration (currently only placement) of components which need to be running on a node where virtualization workloads should be able to run. Changes to Workloads HyperConvergedConfig can be applied only without existing workload.

Bare Metal Platform

☒ true

BareMetalPlatform indicates whether the infrastructure is baremetal.

Feature Gates >

featureGates is a map of feature gate flags. Setting a flag to "true" will enable the feature. Setting "false" or removing the feature gate, disables the feature.

Local Storage Class Name

LocalStorageClassName the name of the local storage class.

Create **Cancel**

- すべてのポッドが openshift-cnv 名前空間で実行状態に移行し、OpenShift Virtualization オペレーターが成功状態になると、オペレーターは使用できるようになります。OpenShift クラスター上に VM を作成できるようになりました。

Project: openshift-cnv ▼

Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the [Understanding Operators documentation](#). Or create an Operator and ClusterServiceVersion using the [Operator SDK](#).

Name ▼ Search by name... 

Name ↑	Managed Namespaces ↑	Status	Last updated	Provided APIs
 OpenShift Virtualization 2.6.2 provided by Red Hat	 openshift-cnv	 Succeeded Up to date	 May 18, 8:02 pm	OpenShift Virtualization Deployment HostPathProvisioner deployment

Red Hat OpenShift Virtualization で ONTAP ストレージを使用して VM を作成する

OpenShift Virtualization を使用して VM を作成します。この手順には、オペレーティングシステムのテンプレートの選択、StorageClasses の設定、および特定の要件を満たす

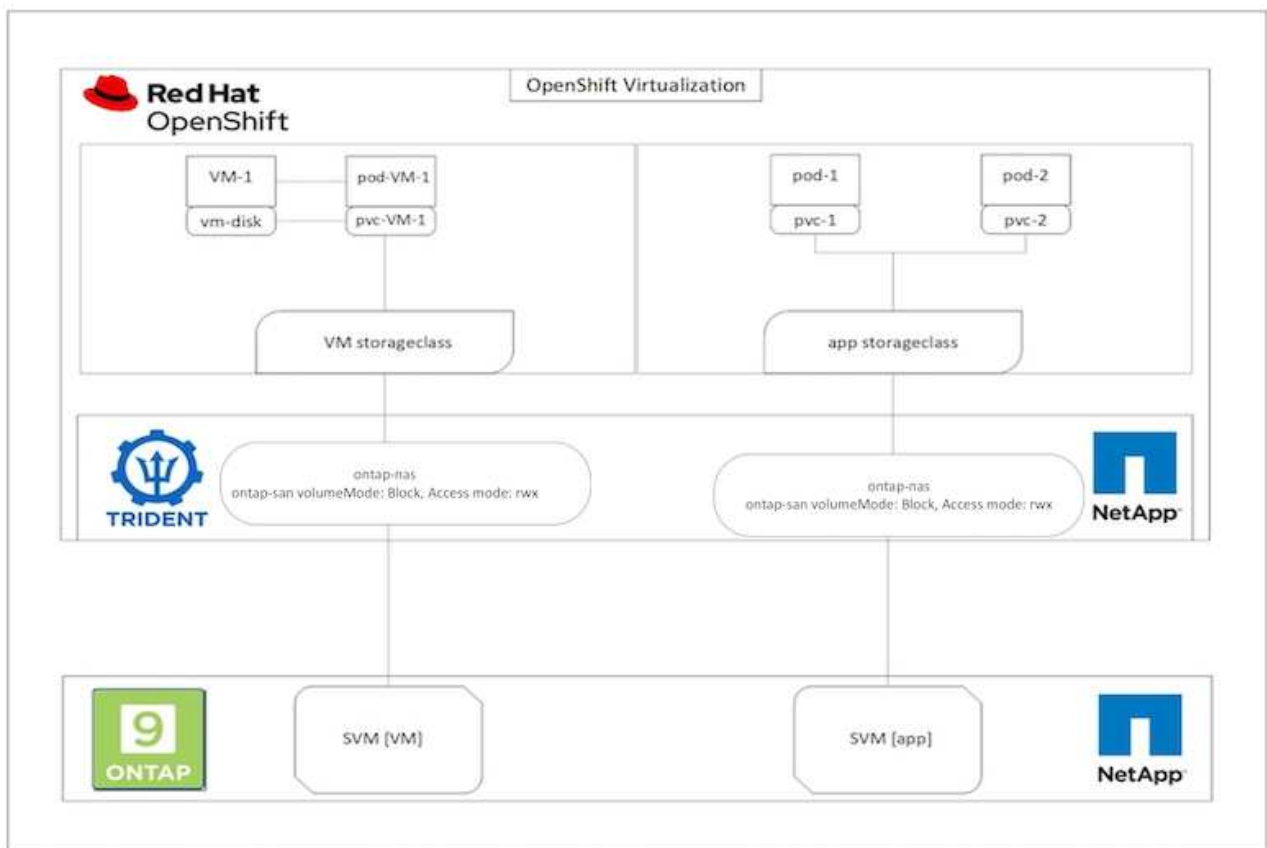
ための VM パラメータのカスタマイズが含まれます。前提条件として、Trident バックエンド、StorageClass、およびボリュームスナップショットクラスオブジェクトをすでに作成済みである必要があります。詳細については、"[Trident設置部](#)" を参照してください。

VMの作成

VM は、オペレーティング システムとデータをホストするためのボリュームを必要とするステートフルな展開です。CNV では、VM はポッドとして実行されるため、VM はTridentを介してNetApp ONTAPでホストされている PV によってバックアップされます。これらのボリュームはディスクとして接続され、VM のブート ソースを含むファイル システム全体を保存します。



Trident StorageClassは、VMディスク用に作成されたPVCがTridentバックエンドを使用してプロビジョニングされるように、クラスター上のデフォルトStorageClassとして設定する必要があります。クラスター上に複数のStorageClassesを設定できますが、VMブートディスク用のPVCのプロビジョニングにはデフォルトのStorageClassを使用する必要があります。追加のディスクは、必要に応じて他のStorageClassesを使用できます。



OpenShift クラスター上に仮想マシンをすばやく作成するには、次の手順を実行します。

1. [仮想化] > [仮想マシン] に移動し、[作成] をクリックします。
2. テンプレートから選択します。
3. ブート ソースが利用可能な目的のオペレーティング システムを選択します。

4. 作成後に仮想マシンを起動するチェックボックスをオンにします。

5. 「仮想マシンのクイック作成」をクリックします。

仮想マシンが作成されて起動し、「実行中」状態になります。デフォルトのStorageClassを使用して、ブートディスク用のPVCと対応するPVを自動的に作成します。将来的にVMをライブマイグレーションできるようにするには、ディスクに使用されるStorageClassがRWXボリュームをサポートできることを確認する必要があります。これはライブマイグレーションの要件です。ontap-nasおよびontap-san (iSCSIおよびNVMe/TCPプロトコル用のvolumeModeブロック) は、それぞれのStorageClassesを使用して作成されたボリュームのRWXアクセスモードをサポートできます。

クラスターで ontap-san StorageClass を設定するには、"[StorageClass を設定するセクション](#)"を参照してください。



「Quick create VirtualMachine」をクリックすると、デフォルトの StorageClass を使用して、VM の起動可能なルートディスク用の PVC と PV が作成されます。ディスクに別の StorageClass を選択するには、「Customize VirtualMachine」 > 「Customize VirtualMachine parameters」 > 「Disks」を選択し、必要な StorageClass を使用するようにディスクを編集します。

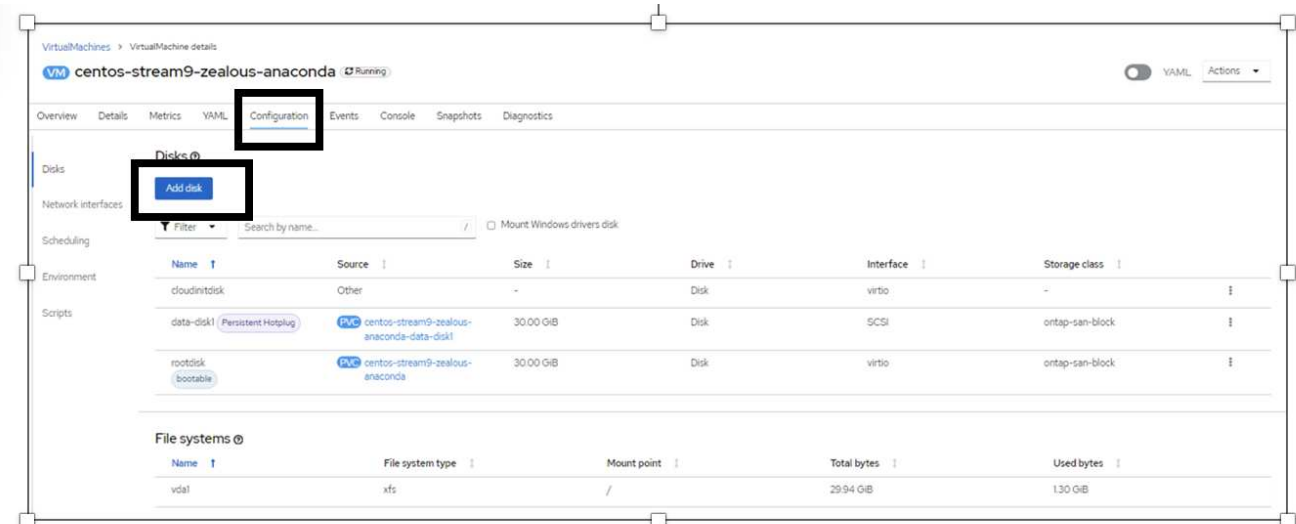
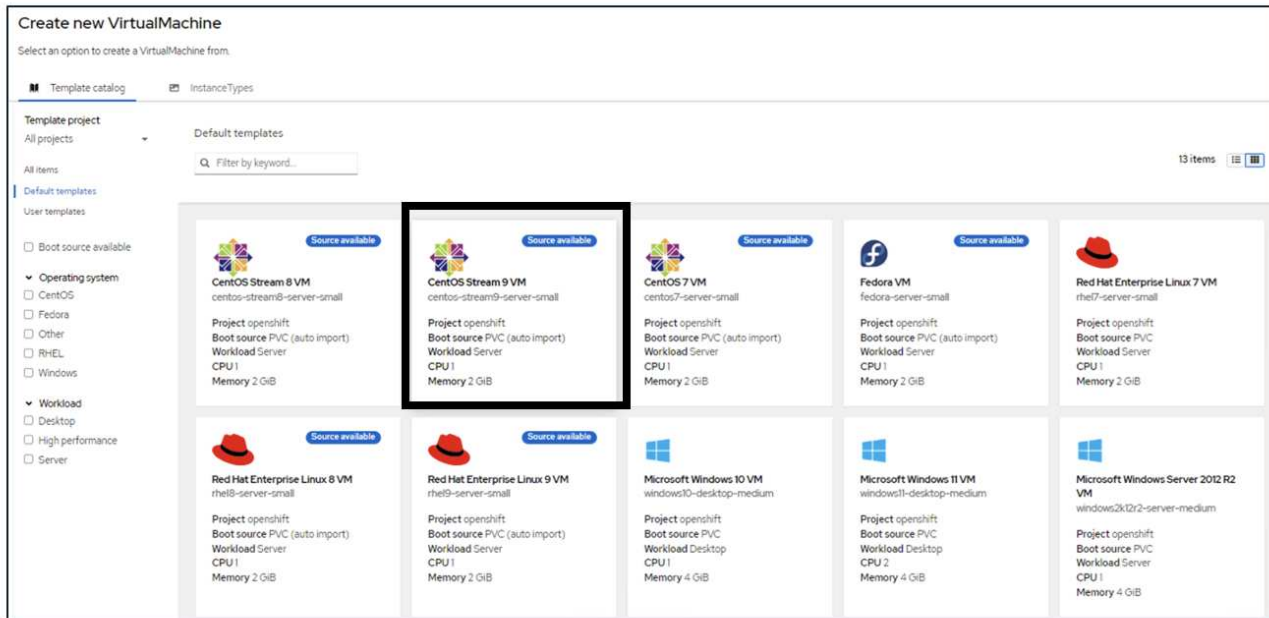
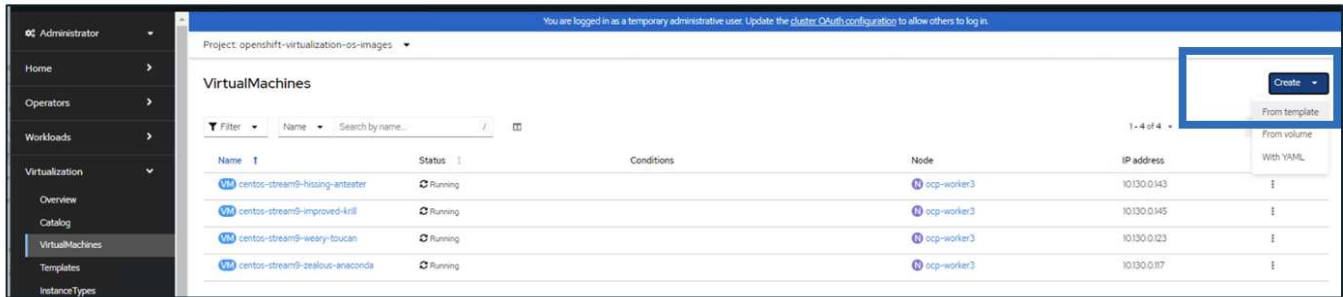
通常、VM ディスクをプロビジョニングする際は、ファイル システムよりもブロック アクセス モードが優先されます。

OS テンプレートを選択した後、仮想マシンの作成をカスタマイズするには、「クイック作成」ではなく「仮想マシンのカスタマイズ」をクリックします。

1. 選択したオペレーティング システムにブート ソースが設定されている場合は、[仮想マシン パラメータのカスタマイズ] をクリックできます。
2. 選択したオペレーティング システムにブート ソースが設定されていない場合は、それを設定する必要があります。手順の詳細については、"[ドキュメント](#)"。
3. ブート ディスクを構成した後、[仮想マシン パラメータのカスタマイズ] をクリックできます。
4. このページのタブから VM をカスタマイズできます。たとえば、*ディスク*タブをクリックし、*ディスクの追加*をクリックして、VM に別のディスクを追加します。
5. 「仮想マシンの作成」をクリックして仮想マシンを作成します。これにより、対応するポッドがバックグラウンドで起動します。



URLまたはレジストリからテンプレートまたはオペレーティングシステムのブートソースが設定されると、PVCが`openshift-virtualization-os-images`プロジェクトを実行し、KVM ゲスト イメージを PVC にダウンロードします。テンプレート PVC に、対応する OS の KVM ゲスト イメージを収容するのに十分なプロビジョニングされた領域があることを確認する必要があります。これらの PVC は、任意のプロジェクトでそれぞれのテンプレートを使用して仮想マシンが作成されると、複製され、ルートディスクとして仮想マシンに接続されます。



Project: openshift-virtualization-os-images

Catalog > Customize template parameters > Customize VirtualMachine

Customize and create VirtualMachine

Template: CentOS Stream 9 VM

Overview | YAML | Scheduling | Environment | Network interfaces | Disks | Scripts | Metadata

Name
centos-stream9-pleased-hamster

Namespace
openshift-virtualization-os-images

Description
Not available

Operating system
CentOS Stream 9 VM

CPU | Memory
1 CPU | 2 GiB Memory

Machine type
pc-q35-rhel9.2.0

Boot mode
BIOS

Start in pause mode
☐

Workload profile
Server

Network interfaces (1)

Name	Network	Type
default	Pod networking	Masquerade

Disks (2)

Name	Drive	Size
rootdisk	Disk	30 GiB
cloudinitdisk	Disk	-

Hardware devices

GPU devices
Not available

Host devices
Not available

Headless mode
☐

Hostname
centos-stream9-pleased-hamster

☒ Start this VirtualMachine after creation

Create VirtualMachine Cancel

ビデオデモンストレーション

次のビデオは、iSCSI ストレージを使用して OpenShift Virtualization で VM を作成するデモを示しています。

[ブロックストレージを使用して OpenShift Virtualization で VM を作成する](#)

Red Hat OpenShift Virtualization で、代替手段を使用して ONTAP ストレージを通じて VM を作成する

OpenShift Virtualization UI を使用して VM を作成する方法に加えて、OpenShift Virtualization で VM を作成する他の方法もあります。このセクションでは、それらの方法のいくつかについて概説します。

Red Hat OpenShift Virtualization を使用して YAML ファイルから VM を作成する

VM は、YAML ファイルを `oc apply` コマンドで適用することによって作成することもできます。この方法は、CLI の使用を好むユーザーや、スクリプトを使用して VM の作成を自動化したいユーザーにとって便利です。この方法で VM を作成するには、VM とその関連リソース（VM ディスクの PVC など）を定義する YAML ファイルを作成します。次に、`oc apply -f <yaml-file>` コマンドを使用して YAML ファイルを適用し、OpenShift クラスタ上に VM を作成できます。

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  name: test-vms-vm
  namespace: test-5vms
  labels:
    app: test-vms-vm
    vm.kubevirt.io/template: centos-stream9-server-small
spec:
  dataVolumeTemplates:
    - apiVersion: cdi.kubevirt.io/v1beta1
      kind: DataVolume
      metadata:
        name: test-vms-vm-rootdisk
      spec:
        sourceRef:
          kind: DataSource
          name: centos-stream9
          namespace: openshift-virtualization-os-images
        storage:
          resources:
            requests:
              storage: 30Gi
  runStrategy: Always
  template:
    metadata:
      labels:
        kubevirt.io/domain: test-vms-vm
    spec:
      domain:
        cpu:
          cores: 1
          sockets: 1
          threads: 1
        devices:
          disks:
            - bootOrder: 1
              disk:
                bus: virtio
                name: rootdisk
            - disk:
                bus: virtio
                name: cloudinitdisk
```

```

    interfaces:
      - masquerade: {}
        model: virtio
        name: default
    resources:
      requests:
        memory: 2Gi
    networks:
      - name: default
        pod: {}
    terminationGracePeriodSeconds: 180
    volumes:
      - dataVolume:
          name: test-vms-vm-rootdisk
          name: rootdisk
      - cloudInitNoCloud:
          userData: |-
            #cloud-config
            user: admin
            password: <replace with your password>
            chpasswd: { expire: False }
          name: cloudinitdisk

```

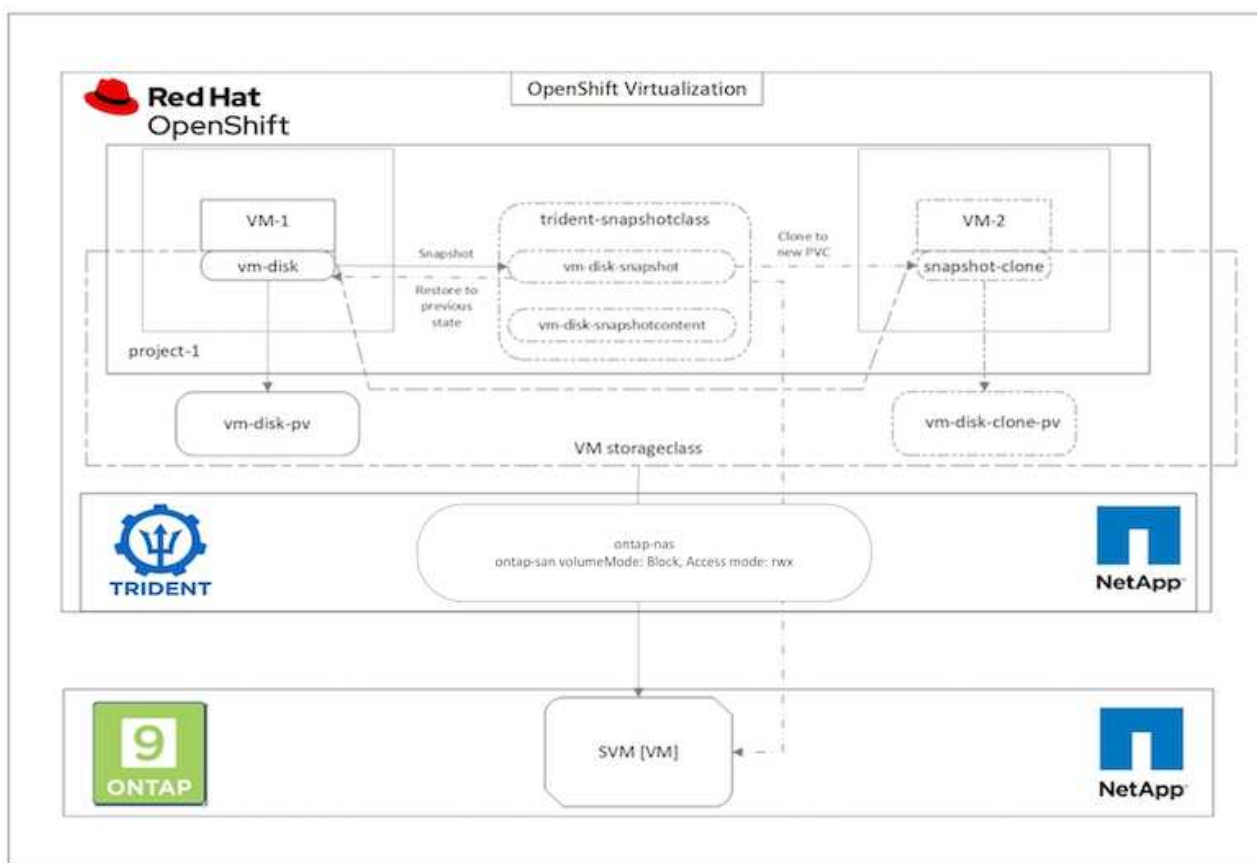
Red Hat OpenShift Virtualization を使用してスナップショット コピーから VM を作成する

OpenShift Virtualization を使用して、スナップショットから VM を作成できます。この手順には、VolumeSnapshotClass の作成、VM の永続ボリューム要求 (PVC) のスナップショットの取得、スナップショットの新しい PVC への復元、および復元された PVC をルートディスクとして使用する新しい VM のデプロイが含まれます。

Trident と Red Hat OpenShift を使用すると、ユーザーは、それによってプロビジョニングされたストレージクラス上の永続ボリュームのスナップショットを取得できます。この機能を使用すると、ユーザーはボリュームのポイントインタイム コピーを取得し、それを使用して新しいボリュームを作成したり、同じボリュームを以前の状態に復元したりできます。これにより、ロールバックからクローン、データの復元まで、さまざまなユースケースが可能またはサポートされます。

OpenShift でのスナップショット操作では、リソース VolumeSnapshotClass、VolumeSnapshot、および VolumeSnapshotContent を定義する必要があります。

- VolumeSnapshotContent は、クラスター内のボリュームから取得された実際のスナップショットです。これは、ストレージ用の PersistentVolume に類似したクラスター全体のリソースです。
- VolumeSnapshot は、ボリュームのスナップショットを作成するための要求です。これは PersistentVolumeClaim に類似しています。
- VolumeSnapshotClass を使用すると、管理者は VolumeSnapshot にさまざまな属性を指定できます。同じボリュームから取得した異なるスナップショットに異なる属性を設定できます。



前提条件

PVCのスナップショットを作成するには、Trident Storage Class上にプロビジョニングされ、PVCが接続された既存のVMが存在し、そのVMが実行状態である必要があります。VolumeSnapshotの作成に使用できるVolumeSnapshotClassが作成されている必要があります。これは、OpenShift Virtualizationをインストールするための前提条件の一部として作成されている必要があります。詳細については、"[VolumeSnapshotClass 構成を作成するステップ 3](#)"を参照してください。

仮想マシンのスナップショットを作成するには、以下の手順を実行してください。

1. ソース VM に接続されている PVC を識別し、その PVC のスナップショットを作成します。移動先 Storage > VolumeSnapshots[ボリュームスナップショットの作成]をクリックします。
2. スナップショットを作成する PVC を選択し、スナップショットの名前を入力するかデフォルトを受け入れ、適切な VolumeSnapshotClass を選択します。次に、「作成」をクリックします。
3. これにより、その時点での PVC のスナップショットが作成されます。

Create VolumeSnapshot

[Edit YAML](#)**PersistentVolumeClaim ***

PVC rhel8-short-frog-rootdisk-28dvv

Name *

rhel8-short-frog-rootdisk-28dvv-snapshot

Snapshot Class *

VSC trident-snapshot-class

Create

Cancel

スナップショットから新しい仮想マシンを作成する

1. まず、スナップショットを新しい PVC に復元します。[ストレージ] > [VolumeSnapshots] に移動し、復元するスナップショットの横にある省略記号をクリックして、[新しい PVC として復元] をクリックします。
2. 新しい PVC の詳細を入力し、「復元」をクリックします。これにより、新しい PVC が作成されます。

Restore as new PVC

When restore action for snapshot **rhel8-short-frog-rootdisk-28dvb-snapshot** is finished a new crash-consistent PVC copy will be created.

Name *

rhel8-short-frog-rootdisk-28dvb-snapshot-restore

Storage Class *



basic

Access Mode *

☐ Single User (RWO) ☒ Shared Access (RWX) ☐ Read Only (ROX)

Size *

20

GiB

VolumeSnapshot details

Created at

 May 21, 12:46 am

Namespace



default

Status

 Ready

API version

snapshot.storage.k8s.io/v1

Size

20 GiB

1. 次に、この PVC から新しい VM を作成します。[仮想化] > [仮想マシン] に移動し、[作成] > [YAML を使用] をクリックします。
2. spec > template > spec > volumes セクションで、コンテナー ディスクではなくスナップショットから作成された新しい PVC を指定します。要件に応じて、新しい VM のその他すべての詳細を指定します。

```
- name: rootdisk
  persistentVolumeClaim:
    claimName: rhel8-short-frog-rootdisk-28dvb-snapshot-restore
```

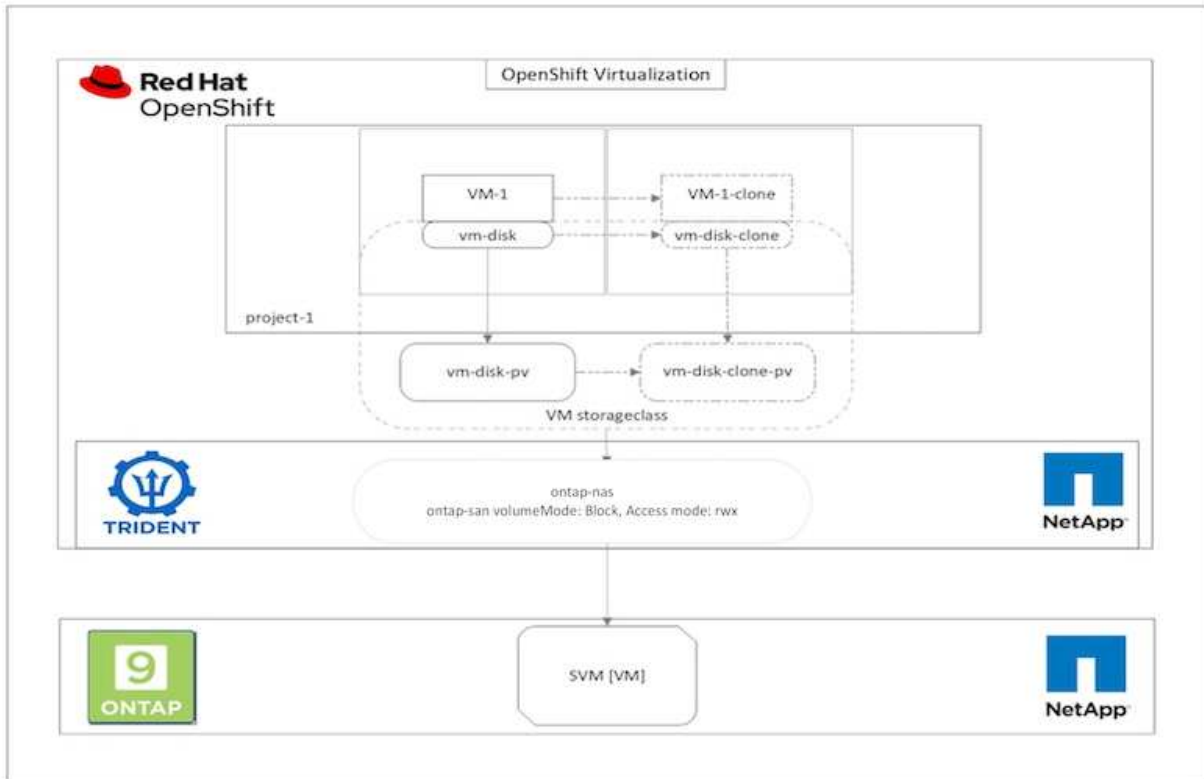
1. [作成] をクリックして新しい VM を作成します。
2. VM が正常に作成されたら、アクセスして、新しい VM の状態が、スナップショットの作成時に使用された PVC を持つ VM の状態 (スナップショットの作成時) と同じであることを確認します。

OpenShift Virtualization で Trident を使用して VM をクローンする

この手順では、Trident CSI ボリュームクローニングを活用することで、ソース VM をシャットダウンして新しい VM を作成するか、ソース VM を実行したままにするかを選択できます。

VMクローニング

OpenShift での既存の VM のクローン作成は、Trident のボリューム CSI クローン作成機能のサポートによって実現されます。CSI ボリューム クローンを使用すると、既存の PVC の PV を複製して、それをデータソースとして使用して新しい PVC を作成できます。新しい PVC が作成されると、それはソース PVC へのリンクや依存関係のない別個のエンティティとして機能します。



CSI ボリュームのクローン作成には、考慮すべき特定の制限があります。

1. ソース PVC と宛先 PVC は同じプロジェクト内にある必要があります。
2. 同じストレージ クラス内でのクローン作成がサポートされます。
3. クローン作成は、ソース ボリュームと宛先ボリュームが同じ VolumeMode 設定を使用している場合にのみ実行できます。たとえば、ブロック ボリュームは別のブロック ボリュームにのみクローン作成できます。

OpenShift クラスター内の VM は、次の 2 つの方法で複製できます。

1. ソースVMをシャットダウンすることで
2. ソースVMを稼働させておくことで

ソース**VM**をシャットダウンすることによって

VM をシャットダウンして既存の VM を複製する機能は、Tridentのサポートを受けて実装されたネイティブの OpenShift 機能です。VM のクローンを作成するには、次の手順を実行します。

1. [ワークロード] > [仮想化] > [仮想マシン] に移動し、クローンを作成する仮想マシンの横にある省略記号をクリックします。
2. 「仮想マシンの複製」をクリックし、新しい VM の詳細を入力します。

Clone Virtual Machine

Name *

rhel8-short-frog-clone

Description

Namespace *

default



Start virtual machine on clone

Configuration

Operating System

Red Hat Enterprise Linux 8.0 or higher

Flavor

Small: 1 CPU | 2 GiB Memory

Workload Profile

server

NICs

default - virtio

Disks

cloudinitdisk - cloud-init disk

rootdisk - 20Gi - basic



The VM rhel8-short-frog is still running. It will be powered off while cloning.

Cancel

Clone Virtual Machine

3. [仮想マシンのクローン作成] をクリックすると、ソース VM がシャットダウンされ、クローン VM の作成が開始されます。

4. この手順が完了すると、クローンされた VM の内容にアクセスして検証できるようになります。

ソースVMを稼働状態に保つことで

既存の VM は、ソース VM の既存の PVC を複製し、複製された PVC を使用して新しい VM を作成することによって複製することもできます。この方法では、ソース VM をシャットダウンする必要はありません。VM をシャットダウンせずにクローンするには、次の手順を実行します。

1. [ストレージ] > [PersistentVolumeClaims] に移動し、ソース VM に接続されている PVC の横にある省略記号をクリックします。
2. 「PVC の複製」をクリックし、新しい PVC の詳細を入力します。

Clone

Name *

rhel8-short-frog-rootdisk-28dvb-clone

Access Mode *

☐ Single User (RWO) ☒ Shared Access (RWX) ☐ Read Only (ROX)

Size *

20 GiB ▼

PVC details

Namespace

 default

Requested capacity

20 GiB

Access mode

Shared Access (RWX)

Storage Class

 basic

Used capacity

2.2 GiB

Volume mode

Filesystem

Cancel

Clone

3. 次に、「クローン」をクリックします。これにより、新しい VM の PVC が作成されます。
4. [ワークロード] > [仮想化] > [仮想マシン] に移動し、[作成] > [YAML を使用] をクリックします。
5. spec > template > spec > volumes セクションで、コンテナ ディスクの代わりにクローンされた PVC を接続します。要件に応じて、新しい VM のその他すべての詳細を指定します。

```
- name: rootdisk
  persistentVolumeClaim:
    claimName: rhel8-short-frog-rootdisk-28dvvb-clone
```

1. [作成] をクリックして新しい VM を作成します。
2. VM が正常に作成されたら、新しい VM にアクセスして、それがソース VM のクローンであることを確認します。

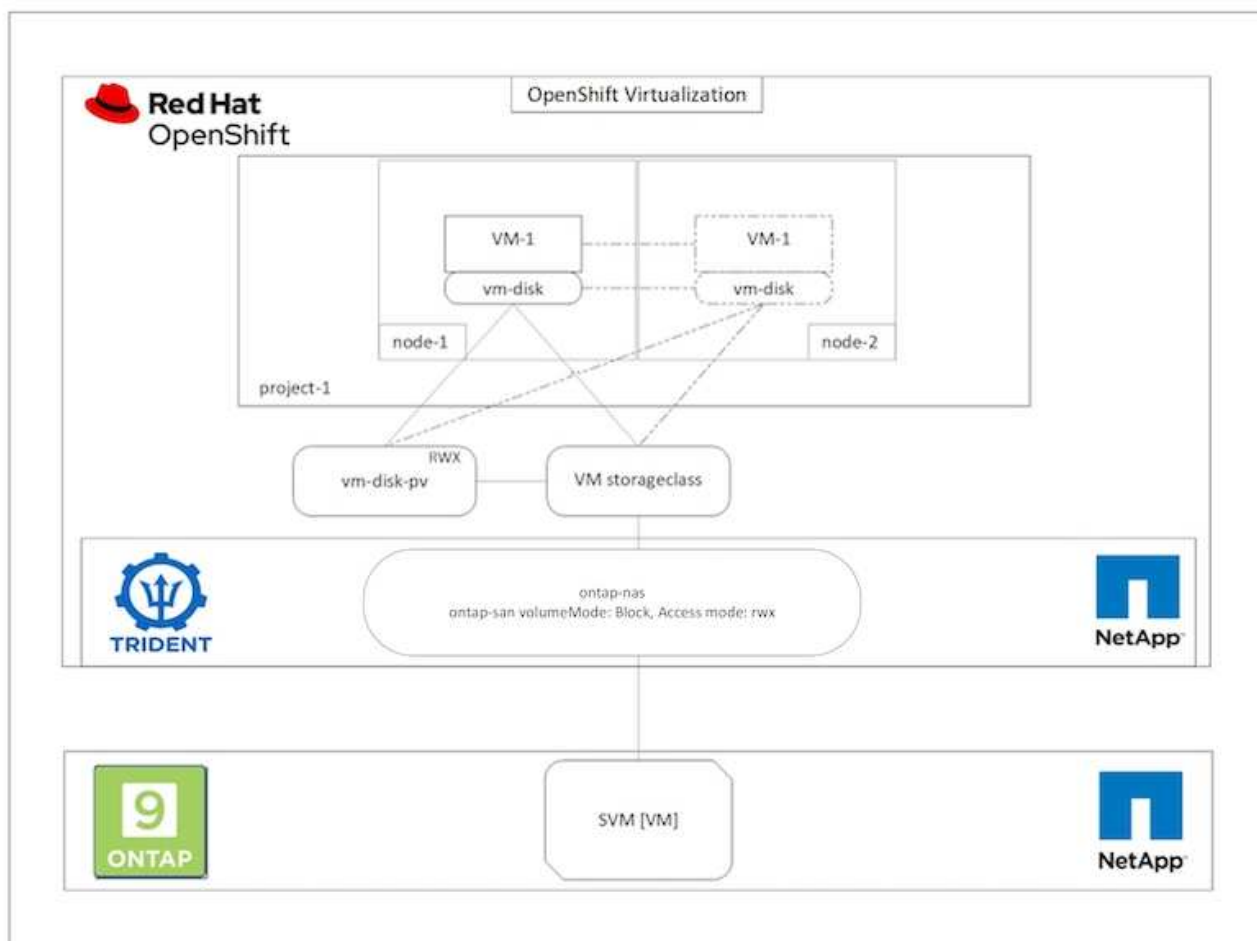
Red Hat OpenShift クラスター内の 2 つのノード間で VM を移行する

OpenShift Virtualization の VM を、ダウンタイムなしでクラスター内の 2 つのノード間で移行します。この手順には、ディスクが RWX 互換のストレージ クラスを使用していることを確認し、移行を開始し、進行状況を監視することが含まれます。

VM ライブマイグレーション

ライブ マイグレーションは、OpenShift クラスター内のあるノードから別のノードにダウンタイムなしで VM インスタンスを移行するプロセスです。OpenShift クラスターでライブ マイグレーションを機能させるには、VM を共有 ReadWriteMany アクセス モードで PVC にバインドする必要があります。ontap-nas ドライバーを使用して構成された Trident バックエンドは、FileSystem プロトコル nfs および smb の RWX アクセス モードをサポートします。ドキュメントを参照してください["ここをクリックしてください。"](#)。ontap-san ドライバーを使用して構成された Trident バックエンドは、iSCSI および NVMe/TCP プロトコルのブロック ボリューム モードの RWX アクセス モードをサポートします。ドキュメントを参照してください["ここをクリックしてください。"](#)。

したがって、ライブ マイグレーションを成功させるには、ontap-nas または ontap-san (volumeMode: Block) ストレージ クラスを使用して、PVC でディスク (ブート ディスクと追加のホット プラグ ディスク) を VM にプロビジョニングする必要があります。PVC が作成されると、Trident は NFS 対応または iSCSI 対応の SVM に ONTAP ボリュームを作成します。

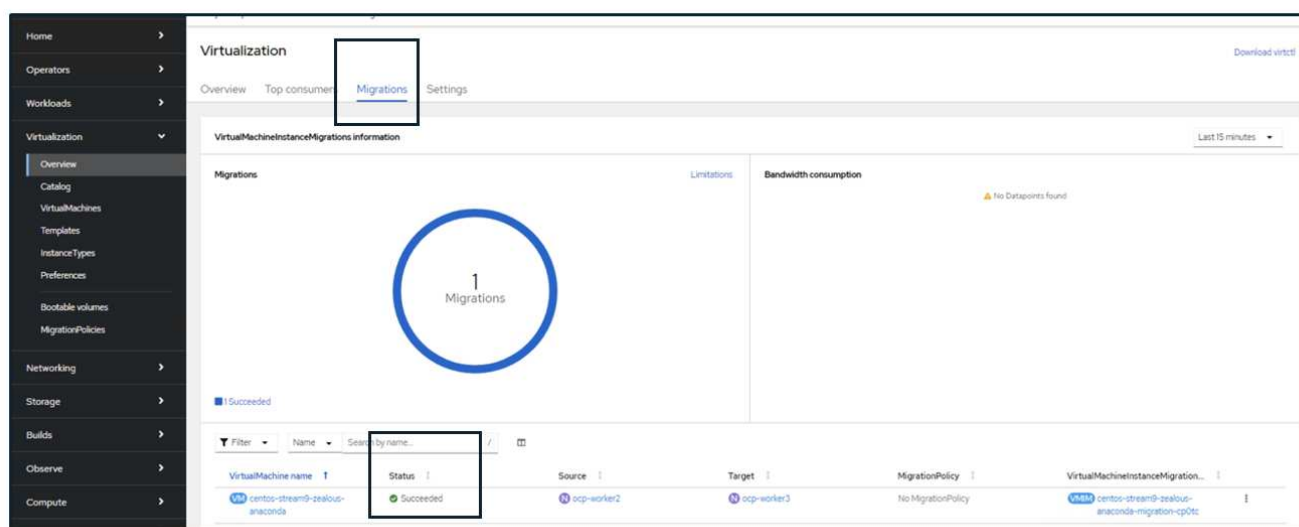
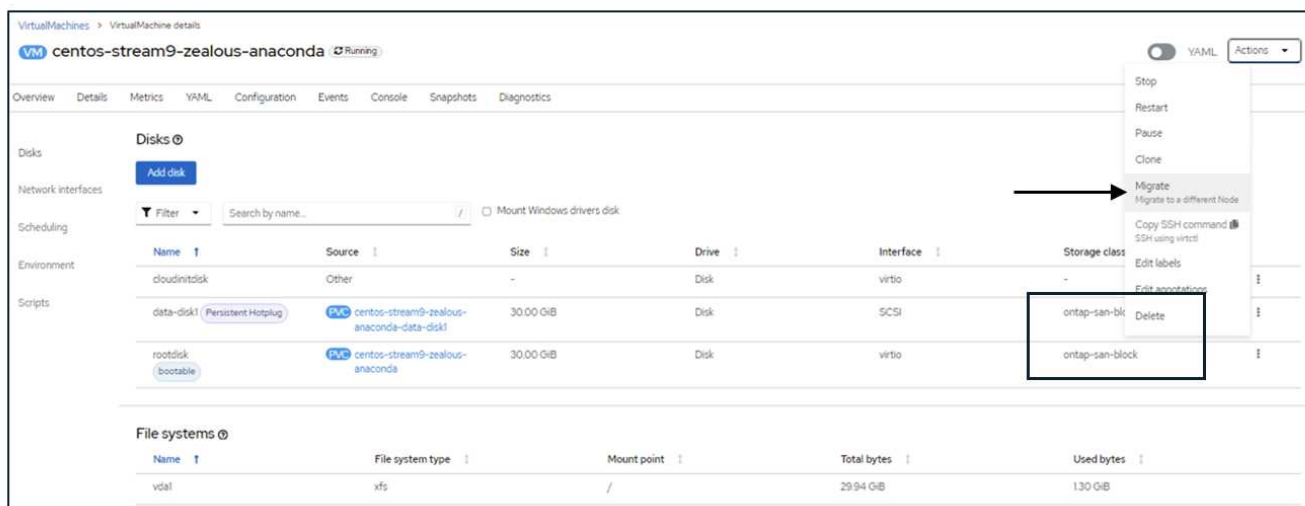


以前に作成され、実行状態にある VM のライブ マイグレーションを実行するには、次の手順を実行します。

1. ライブマイグレーションする VM を選択します。
2. *構成*タブをクリックします。
3. VM のすべてのディスクが、RWX アクセス モードをサポートできるストレージ クラスを使用して作成されていることを確認します。
4. 右隅の*アクション*をクリックし、*移行*を選択します。
5. 移行の進行状況を確認するには、左側のメニューで [仮想化] > [概要] に移動し、[移行] タブをクリックします。VM の移行は、保留中*から*スケジュール、そして*成功*へと移行します。



evictionStrategy が LiveMigrate に設定されている場合、OpenShift クラスター内の VM インスタンスは、元のノードがメンテナンス モードになると、自動的に別のノードに移行します。



OpenShift Virtualization におけるストレージクラス間の VM のストレージ移行

OpenShift Virtualization では、仮想マシン（VM）を削除せずに、Web コンソールから直接、VM のディスクを別のストレージクラスに移行できます。このプロセスにより、VM ストレージ全体または特定のボリュームを移動できます。

仮想マシンのストレージクラスを移行する方法を、以下の手順で説明します。

1. ターゲットストレージクラスが正しく構成され、OpenShift クラスタで使用可能であることを確認してください。これには、必要なストレージバックエンド（例：NetApp ONTAP）が設定済みでアクセス可能であることが含まれます。

例を表示

```
[root@00-50-56-b5-2b-9b ~]# oc get tbe -n trident
NAME          BACKEND          BACKEND UUID
tbe-k2vqg     tbc-san          3e9dc8ae-3b29-44c8-99a9-affdeaa6b4cc
tbe-mtp5n     tbc-nas-eco      a59e8053-bb00-4410-9b12-1dc4d283928b
tbe-nwqvq     tbc-nas-50       8b49f02f-3bc3-4720-b60f-bf90c3600008
tbe-wtp9h     tbc_san_eco      61745526-04cd-4acb-8f1b-daa9aa79f05a
tbe-zfp46     tbc-nas          259a1c2a-0a84-488d-ab62-ccea45499e09
```

```
[root@00-50-56-b5-2b-9b ~]# oc get sc
NAME          PROVISIONER          RECLAIMPOLICY  VOLUMEBINDINGMODE  ALLOWVOLUMEEXPANSION  AGE
sc-nas (default)  csi.trident.netapp.io  Delete         Immediate          true                  35d
sc-nas-50        csi.trident.netapp.io  Delete         Immediate          true                  22d
sc-nas-economy    csi.trident.netapp.io  Delete         Immediate          true                  20d
sc-san           csi.trident.netapp.io  Delete         Immediate          false                 21d
```

2. OpenShift Container Platform の Web コンソールで、仮想化パースペクティブに移動し、移行する VM を選択します。

例を表示

VirtualMachines > VirtualMachine details

VM rhel9-sc-migration Running

OverviewMetricsYAMLConfigurationEventsConsoleSnapshotsDiagnostics

Details

Name

rhel9-sc-migration

Status

Running

Created

May 14, 2026, 1:31 PM (17 minutes ago)

Operating system

Red Hat Enterprise Linux 9.7 (Plow)

CPU | Memory

1 CPU | 2 GiB Memory

Time zone

EDT

Template

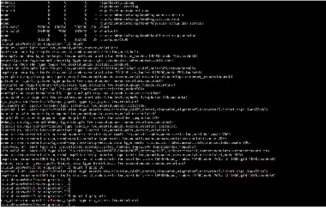
rhel9-server-small

Hostname

rhel9-sc-migration

VNC console

[Open web console](#)



Alerts (0)

General

Namespace

test-sc-migration

Node

00-50-56-b5-bd-fa

VirtualMachineInstance

rhel9-sc-migration

Pod

virt-launcher-rhel9-sc-migration-5jkdw

Owner

No owner

Snapshots (0)

[Take snapshot](#)

VirtualMachines > VirtualMachine details

VM rhel9-sc-migration Running

OverviewMetricsYAMLConfigurationEventsConsoleSnapshotsDiagnostics

Configuration

Disks

Add disk

Filter

Search by name...

☐ Mount Windows drivers disk

	Name	Source	Size	Drive	Interface	Storage class
cloudinitdisk		Other	-	Disk	virtio	-
disk-rhel9-nas		PVC dv-rhel9-migration-disk-emerald-wren-75-d88usc	53 GiB	Disk	virtio	sc-nas
rootdisk		PVC rhel9-migration	31.8 GiB	Disk	virtio	sc-nas

bootable

```
[root@00-50-56-b5-2b-9b ~]# oc get vm,pvc -n test-sc-migration
NAME                                AGE    STATUS    READY
virtualmachine.kubevirt.io/rhel9-sc-migration  4m19s  Running  True

NAME                                STATUS    VOLUME                                CAPACITY    ACCESS MODES    STORAGECLASS    VOLUMEATTRIBUTESCLASS    AGE
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc  Bound    pvc-bb54342f-c01f-453e-864d-e8f72e710a2f  53Gi        RWX              sc-nas          <unset>                  4m19s
persistentvolumeclaim/rhel9-migration  Bound    pvc-46a34903-942f-4ddc-8e6d-97a008f05d09  34144990004  RWX              sc-nas          <unset>                  4m19s
[root@00-50-56-b5-2b-9b ~]#
```

- 「アクション」→「移行」→「ストレージの移行」をクリックします。移行するボリュームと移行先のストレージクラスを選択し、「移行」をクリックします。移行プロセスが開始され、VMの「イベント」タブで進行状況を監視できます。移行が完了すると、仮想マシンのディスクは新しいストレージクラスを使用するようになります。

56

例を表示

VirtualMachines > VirtualMachine details

VM **rhel9-sc-migration** Running

Overview Metrics YAML Configuration Events Console Snapshots Diagnostics

Details

Name	rhel9-sc-migration
Status	Running
Created	May 14, 2026, 1:31 PM (19 minutes ago)
Operating system	Red Hat Enterprise Linux 9.7 (Plow)
CPU Memory	1 CPU 2 GiB Memory
Time zone	EDT
Template	rhel9-server-small
Hostname	rhel9-sc-migration

VNC console
[Open web console](#)

Alerts (0)

General Compute
Migrate VirtualMachine to a different Node

Namespace Storage
Migrate VirtualMachine storage to a different StorageClass

Node VirtualMachineInstance
rhel9-sc-migration

Pod
virt-launcher-rhel9-sc-migration-Sjkdw

Owner
No owner

Snapshots (0) [Take snapshot](#)

Migrate VirtualMachine storage

Migrate VirtualMachine storage to a different StorageClass.

- 1 Migration details
- 2 Destination StorageClass
- 3 Review

Migration details

Select the storage to migrate for VM **rhel9-sc-migration**

- ☒ The entire VirtualMachine
- ☐ Selected volumes

Back

Next

Cancel

Migrate VirtualMachine storage

Migrate VirtualMachine storage to a different StorageClass.

- 1 Migration details
- 2 Destination StorageClass
- 3 Review

Destination StorageClass

Select the destination storage for the VirtualMachine storage migration.

SC sc-nas-economy

Back

Next

Cancel

Migrate VirtualMachine storage

Migrate VirtualMachine storage to a different StorageClass.

1 Migration details

2 Destination StorageClass

3 Review

Review

Verify the details and click **Migrate VirtualMachine storage** to start the migration

Migration type

VirtualMachine storage

Destination StorageClass

sc-nas-economy

Migrating 1 out of 3

disk-rhel9-nas

Back

Migrate VirtualMachine storage

Cancel

Migrate VirtualMachine storage

Migrate VirtualMachine storage to a different StorageClass.

In progress

Started on

May 14, 2026, 1:54 PM

Status

Scheduling

Stop

Cancel

View report

4. 移行が完了したら、Web コンソールで VM の詳細を確認するか、OpenShift CLI を使用して、VM のディスクが新しいストレージクラスを使用していることを確認できます。

例を表示

```
[root@00-50-56-b5-2b-9b ~]# oc get vm,pvc -n test-sc-migration
NAME                                     AGE      STATUS    READY
virtualmachine.kubevirt.io/rhel9-sc-migration  24m      Migrating  True

NAME                                     STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc  Bound   pvc-bb54342f-c01f-453e-864d-e8f72e710a2f  53Gi      Rwx          sc-nas        <unset>                24m
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb  Bound   pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b  60322815673  Rwx          sc-nas-economy  <unset>                2m17s
persistentvolumeclaim/rhel9-migration  Bound   pvc-46a34903-942f-4ddc-8e6d-974008f05d09  34144990004  Rwx          sc-nas        <unset>                24m
```

```
[root@00-50-56-b5-2b-9b ~]# oc get all -n test-sc-migration
Warning: apps.openshift.io/v1 DeploymentConfig is deprecated in v1.24, unavailable in v1.10000+
Warning: kubevirt.io/v1 VirtualMachineInstancePresets is now deprecated and will be removed in v2.

NAME                                     READY  STATUS    RESTARTS  AGE
pod/virt-launcher-rhel9-sc-migration-44dc7  1/1    Running   0          2m50s
pod/virt-launcher-rhel9-sc-migration-5jkdw   0/1    Completed 0          25m

NAME      TYPE      CLUSTER-IP  EXTERNAL-IP  PORT(S)    AGE
service/headless  ClusterIP  None        <none>       5434/TCP   25m

NAME                                     PHASE    PROGRESS  RESTARTS  AGE
datavolume.cdi.kubevirt.io/dv-rhel9-migration-disk-emerald-wren-75-d88usc  Succeeded  100.0%    0          25m
datavolume.cdi.kubevirt.io/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb  Succeeded  100.0%    0          3m7s
datavolume.cdi.kubevirt.io/rhel9-migration  Succeeded  100.0%    0          25m

NAME                                     PHASE    VMI
virtualmachineinstancemigration.kubevirt.io/kubevirt-workload-update-lbfj9  Succeeded  rhel9-sc-migration

NAME                                     AGE    PHASE    IP          NODENAME  READY
virtualmachineinstance.kubevirt.io/rhel9-sc-migration  25m    Running  10.131.0.22  00-50-56-b5-84-74  True

NAME      AGE    STATUS    READY
virtualmachine.kubevirt.io/rhel9-sc-migration  25m    Running  True
[root@00-50-56-b5-2b-9b ~]#
[root@00-50-56-b5-2b-9b ~]# oc get vm,pvc -n test-sc-migration
NAME                                     AGE      STATUS    READY
virtualmachine.kubevirt.io/rhel9-sc-migration  26m      Running   True

NAME                                     STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc  Bound   pvc-bb54342f-c01f-453e-864d-e8f72e710a2f  53Gi      Rwx          sc-nas        <unset>                26m
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb  Bound   pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b  60322815673  Rwx          sc-nas-economy  <unset>                3m28s
persistentvolumeclaim/rhel9-migration  Bound   pvc-46a34903-942f-4ddc-8e6d-974008f05d09  34144990004  Rwx          sc-nas        <unset>                26m
[root@00-50-56-b5-2b-9b ~]#
```

```
[root@00-50-56-b5-2b-9b ~]# oc describe pv pvc-46a34903-942f-4ddc-8e6d-974008f05d09 -n test-sc-migration
Name:      pvc-46a34903-942f-4ddc-8e6d-974008f05d09
Labels:    <none>
Annotations:  pv.kubernetes.io/provisioned-by: csi.trident.netapp.io
             volume.kubernetes.io/provisioner-deletion-secret-name:
             volume.kubernetes.io/provisioner-deletion-secret-namespace:
Finalizers: [external-provisioner.volume.kubernetes.io/finalizer kubernetes.io/pv-protection external-attacher/csi-trident-netapp-io]
StorageClass:  sc-nas
Status:        Bound
Claim:         test-sc-migration/rhel9-migration
Reclaim Policy: Delete
Access Modes:  Rwx
VolumeMode:    Filesystem
Capacity:      34144990004
Node Affinity: <none>
Message:
Source:
  Type:          CSI (a Container Storage Interface (CSI) volume source)
  Driver:        csi.trident.netapp.io
  FSType:
  VolumeHandle:  pvc-46a34903-942f-4ddc-8e6d-974008f05d09
  ReadOnly:      false
  VolumeAttributes:  backendUUID=259a1c2a-0a84-488d-ab62-ccca45499e09
                    internalName=oc_automation_pvc_46a34903_942f_4ddc_8e6d_974008f05d09
                    name=pvc-46a34903-942f-4ddc-8e6d-974008f05d09
                    protocol=file
                    storage.kubernetes.io/csiProvisionerIdentity=1778505459781-8296-csi.trident.netapp.io
Events:         <none>
[root@00-50-56-b5-2b-9b ~]#
```

```

NSOL-NetApp-A70-T19U01:> volume show -vserver oc-automation-svm -volume oc_automation_pvc_46a34903_942f_4ddc_8e6d_974008f05d09

Vserver Name: oc-automation-svm
Volume Name: oc_automation_pvc_46a34903_942f_4ddc_8e6d_974008f05d09
Aggregate Name: NSOL_NetApp_A70_T19U01a_NVME_SSD_1
List of Aggregates for FlexGroup Constituents: NSOL_NetApp_A70_T19U01a_
NVME_SSD_1
Encryption Type: none
List of Nodes Hosting the Volume: NSOL-NetApp-A70-T19U01a
Volume Size: 31.80GB
Volume Data Set ID: 1716
Volume Master Data Set ID: 2152408880
Volume State: online
Volume Style: flex
Extended Volume Style: flexvol
FlexCache Endpoint Type: none
Is Cluster-Mode Volume: true
Is Constituent Volume: false
Number of Constituent Volumes: -
Export Policy: default
User ID: 0
Group ID: 0
Security Style: unix
UNIX Permissions: ---rwxrwxrwx
Junction Path: /oc_automation_pvc_46a34903_942f_4ddc_8e6d_974008f05d09
Junction Path Source: RW_volume
Junction Active: true
Junction Parent Volume: oc_automation-svm_root
Comment:
Available Size: 30.03GB
Filesystem Size: 31.80GB
Total User-Visible Size: 31.80GB
Used Size: 1.76GB
Used Percentage: 5%
Volume Nearly Full Threshold Percent: 95%
Volume Full Threshold Percent: 98%
Maximum Autosize: 38.16GB
Minimum Autosize: 31.80GB

```

```

[root@00-50-56-b5-2b-9b ~]# oc get all -n test-sc-migration
Warning: apps.openshift.io/v1 DeploymentConfig is deprecated in v4.14+, unavailable in v4.10000+
Warning: kubevirt.io/v1 VirtualMachineInstancePresets is now deprecated and will be removed in v2.
NAME                                READY    STATUS    RESTARTS   AGE
pod/virt-launcher-rhel9-sc-migration-44dc7    1/1      Running    0           2m50s
pod/virt-launcher-rhel9-sc-migration-5jdkw    0/1      Completed  0           25m

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/headless                    ClusterIP     None          <none>         5434/TCP   25m

NAME                                PHASE    PROGRESS    RESTARTS   AGE
datavolume.cdi.kubevirt.io/dv-rhel9-migration-disk-emerald-wren-75-d88usc    Succeeded  100.0%      0           25m
datavolume.cdi.kubevirt.io/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb    Succeeded  100.0%      0           3m7s
datavolume.cdi.kubevirt.io/rhel9-migration                                Succeeded  100.0%      0           25m

NAME                                PHASE    VMI
virtualmachineinstancemigration.kubevirt.io/kubevirt-workload-update-lbfj9    Succeeded  rhel9-sc-migration

NAME                                AGE    PHASE    IP            NODENAME    READY
virtualmachineinstance.kubevirt.io/rhel9-sc-migration    25m    Running  10.131.0.22   00-50-56-b5-84-74    True

NAME                                AGE    STATUS    READY
virtualmachine.kubevirt.io/rhel9-sc-migration    25m    Running    True
[root@00-50-56-b5-2b-9b ~]#
[root@00-50-56-b5-2b-9b ~]#
[root@00-50-56-b5-2b-9b ~]# oc get vm,pvc -n test-sc-migration
NAME                                AGE    STATUS    READY
virtualmachine.kubevirt.io/rhel9-sc-migration    26m    Running    True

NAME                                STATUS    VOLUME    CAPACITY    ACCESS MODES    STORAGECLASS    VOLUMEATTRIBUTESCLASS    AGE
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc    Bound    pvc-bb5d1342f-c01f-453e-864d-abf72e710a2f    536i    Rwx    sc-nas    <unset>    26m
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb    Bound    pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b    60322815673    Rwx    sc-nas-economy    <unset>    3m28s
persistentvolumeclaim/rhel9-migration                                Bound    pvc-46a34903-942f-4ddc-8e6d-974008f05d09    34144990004    Rwx    sc-nas    <unset>    26m
[root@00-50-56-b5-2b-9b ~]#

```

```

[root@00-50-56-b5-2b-9b ~]# oc describe pv -n test-sc-migration pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b
Name:
Labels:
Annotations:
pv.kubernetes.io/provisioned-by: csi.trident.netapp.io
volume.kubernetes.io/provisioner-deletion-secret-name:
volume.kubernetes.io/provisioner-deletion-secret-namespace:
Finalizers: [external-provisioner.volume.kubernetes.io/finalizer kubernetes.io/pv-protection external-attacher/csi-trident-netapp-io]
StorageClass: sc-nas-economy
Status: Bound
Claim: test-sc-migration/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb
Reclaim Policy: Delete
Access Modes: Rwx
VolumeMode: Filesystem
Capacity: 60322815673
Node Affinity: <none>
Message:
Source:
Type: CSI (a Container Storage Interface (CSI) volume source)
Driver: csi.trident.netapp.io
FSType:
VolumeHandle: pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b
ReadOnly: false
VolumeAttributes:
  backendUUID: a59e8053-bb00-4410-9b12-1dc4d283928b
  internalID: /svm/oc-automation-svm/flexvol/trident_qtree_pool_oc_automation_OGBGMIAOLG/qtree/oc_automation_pvc_75a885f0_824e_407c_ab29_8cc546...
  internalName: oc_automation_pvc_75a885f0_824e_407c_ab29_8cc546dbdb3b
  name: pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b
  protocol: file
  storage.kubernetes.io/csiProvisionerIdentity: 1778505459781-8296-csi.trident.netapp.io
Events:
<none>

```

Qtrees

+ Add

🔍 ⬇️ 🗃️ ☰

☐	Name	Storage VM	Volume name	Mount path
	oc_automation_pvc_75a885f0_824e_407c_ab29_8ccf			
▼	oc_automation_pvc_75a885f0_824e_407c_ab29_8cc546dbdb3b	oc-automation-svm	trident_qtree_pool_oc_automation_OGBGMIAOLG	/trident_qtree_pool_oc_automation_OGBGMIAOLG/oc_...

著作権に関する情報

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

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

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

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

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

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

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

商標に関する情報

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