



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

NetApp virtualization solutions

NetApp  
August 18, 2025

# 目次

Red Hat OpenShift クラスタにTridentをインストールし、ストレージ オブジェクトを作成する . . . . .	1
ビデオデモンストレーション . . . . .	6
オンプレミス OpenShift クラスタのTrident構成 . . . . .	6
FSxNストレージを使用したROSAクラスタのTrident構成 . . . . .	11
Tridentボリューム スナップショット クラスの作成 . . . . .	12
Tridentストレージとスナップショットクラスでデフォルトを設定する . . . . .	13

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

OpenShift クラスターに Red Hat Certified Trident Operator を使用して Tridentをインストールし、ブロック アクセス用にワーカー ノードを準備します。ONTAPおよびFSxNストレージ用のTridentバックエンドおよびストレージ クラス オブジェクトを作成して、コンテナと VM の動的ボリューム プロビジョニングを有効にします。

OpenShift Virtualization で VM を作成する必要がある場合は、OpenShift Virtualization をクラスター (オンプレミスおよび ROSA) にインストールする前に、Tridentをインストールし、バックエンド オブジェクトとストレージ クラス オブジェクトを openShift クラスターに作成する必要があります。デフォルトのストレージ クラスとデフォルトのボリューム スナップショット クラスは、クラスター内のTridentストレージとスナップショット クラスに設定する必要があります。これが設定されている場合にのみ、OpenShift Virtualization はテンプレートを使用して VM を作成するためにローカルでゴールデン イメージを使用できるようになります。

Tridentをインストールする前に OpenShift Virtualization Operator がインストールされている場合は、以下のコマンドを使用して、別のストレージ クラスを使用して作成されたゴールデン イメージを削除し、Tridentストレージおよびボリューム スナップショット クラスのデフォルトが設定されていることを確認して、OpenShift Virtualization がTridentストレージ クラスを使用してゴールデン イメージを作成できるようにします。

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

ROSA クラスターの FSxN ストレージ用の trident オブジェクトを作成するためのサンプル yaml ファイルと、VolumeSnapshotClass のサンプル yaml ファイルを取得するには、このページを下にスクロールしてください。

- Tridentのインストール\*\*

## Red Hat Certified Operator を使用してTridentをインストールする

このセクションでは、Red Hat Certified Trident Operator を使用してTridentをインストールする詳細を説明します。"Tridentのドキュメントを参照してください" Tridentをインストールする他の方法については、こちらをご覧ください。Trident 25.02 のリリースにより、オンプレミスおよびクラウドの Red Hat OpenShift や、AWS 上の Red Hat OpenShift Service などのマネージド サービスのTridentユーザーは、Operator Hub からTrident Certified Operator を使用してTridentをインストールできるようになりました。Trident はこれまでコミュニティ オペレーターとしてのみ利用可能であったため、これはOpenShift ユーザー コミュニティにとって重要です。

Red Hat 認定Tridentオペレーターの利点は、OpenShift (オンプレミス、クラウド、または ROSA によるマネージド サービス) で使用する場合、オペレーターとそのコンテナの基盤がNetAppによって完全にサポートされていることです。さらに、NetApp Trident はお客様に無料で提供されるため、Red Hat OpenShift とシームレスに動作することが検証され、ライフサイクル管理が容易になるようにパッケージ化された認定オペレータを使用してインストールするだけで済みます。

さらに、Trident 25.02 オペレーター (および将来のバージョン) では、iSCSI 用にワーカー ノードを準備するというオプションの利点が提供されます。これは、ワークロードを ROSA クラスターにデプロイし、特に OpenShift Virtualization VM ワークロードで iSCSI プロトコルを FSxN で使用する予定がある場合に特に有利です。FSxN を使用する ROSA クラスター上の iSCSI のワーカー ノードの準備の課題は、クラスターにTrident をインストールするときにこの機能によって軽減されました。

オペレーターを使用したインストール手順は、オンプレミス クラスターにインストールする場合でも、ROSA にインストールする場合でも同じです。Operator を使用してTridentをインストールするには、Operator ハブをクリックし、Certified NetApp Tridentを選択します。インストール ページでは、最新バージョンがデフォルトで選択されます。「インストール」をクリックします。

The screenshot shows the Red Hat OperatorHub interface. On the left, there's a sidebar with navigation links: Home, Operators (selected), OperatorHub (selected), Installed Operators, Workloads, Networking, Storage, Builds, Observe, and Compute. Under Operators, there's a dropdown menu. The main area is titled "OperatorHub" and contains a search bar with the text "Trident". Below the search bar, there are two columns: "All Items" and "All Items". The "All Items" column lists categories: AI/Machine Learning, Application Runtime, Big Data, Cloud Provider, Database, Developer Tools, Development Tools, Drivers and plugins, Integration & Delivery, Logging & Tracing, Modernization & Migration, and Monitoring. The "All Items" column lists categories: AI/Machine Learning, Application Runtime, Big Data, Cloud Provider, Database, Developer Tools, Development Tools, Drivers and plugins, Integration & Delivery, Logging & Tracing, Modernization & Migration, and Monitoring. There are two operator cards displayed: "NetApp Trident" (Certified, provided by NetApp, Inc.) and "NetApp Trident" (Community, provided by NetApp, Inc.). The "NetApp Trident" card has a green checkmark indicating it is installed.

## 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

オペレーターがインストールされたら、オペレーターの表示をクリックし、Trident Orchestrator のインスタンスを作成します。iSCSI ストレージ アクセス用にワーカー ノードを準備する場合は、yaml ビューに移動し、iscsi を追加して nodePrep パラメータを変更します。

## 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
```

これで、クラスター内ですべての Trident ポッドが実行されるはずです。

```
[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-1d4b8              2/2    Running   0          16h
trident-node-linux-mj5r8              2/2    Running   0          16h
trident-node-linux-mkmmpp            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 ~]#
```

OpenShift クラスターのワーカー ノードで iSCSI ツールが有効になっていることを確認するには、ワーカー ノードにログインし、iscsid、multipathd active、および multipath.conf ファイル内のエントリが次のように表示されていることを確認します。

```
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#
```

```
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#
```

```
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

次のビデオは、Red Hat Certified Trident Operator を使用してTridentをインストールするデモを示しています。

[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](#)

オンプレミス OpenShift クラスタのTrident構成

## NAS 向けTridentバックエンドとストレージクラス

```
cat 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: <cluster management lif>
  backendName: tbc-nas
  svm: zoneb
  storagePrefix: testzoneb
  defaults:
    nameTemplate: "{{ .config.StoragePrefix }}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"
  credentials:
    name: tbc-nas-secret
```

```
cat 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
```

## iSCSI 用のTridentバックエンドとストレージ クラス

```
# cat 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
```

```
# cat 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 用のTridentバックエンドとストレージクラス

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

```
# cat 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
```

## FC 用のTridentバックエンドとストレージ クラス

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

```
# cat 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
```

# FSxNストレージを使用したROSAクラスタのTrident構成

## FSxN NAS のTridentバックエンドとストレージクラス

```
#cat tbc-fsx-nas.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-fsx-ontap-nas-secret
  namespace: trident
type: Opaque
stringData:
  username: <cluster admin lif>
  password: <cluster admin passwd>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-fsx-ontap-nas
  namespace: trident
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
```

```
# cat sc-fsx-nas.yaml
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
```

## FSxN iSCSI のTridentバックエンドとストレージ クラス

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

```
# cat 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
```

## Tridentボリューム スナップショット クラスの作成

## Tridentボリューム スナップショット クラス

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

バックエンド構成、ストレージクラス構成、スナップショット構成に必要な yaml ファイルを用意したら、次のコマンドを使用して、トライデントバックエンド、ストレージクラス、スナップショットクラスのオブジェクトを作成できます。

```
oc create -f <backend-filename.yaml> -n trident
oc create -f <storageclass-filename.yaml>
oc create -f <snapshotclass-filename.yaml>
```

## Tridentストレージとスナップショットクラスでデフォルトを設定する

## Tridentストレージとスナップショットクラスでデフォルトを設定する

これで、必要な Trident ストレージ クラスとボリューム スナップショット クラスを OpenShift クラスターのデフォルトとして設定できるようになりました。前述のように、OpenShift Virtualization がデフォルトのテンプレートから VM を作成するためにゴールデン イメージ ソースを使用できるようにするには、デフォルトのストレージ クラスとボリューム スナップショット クラスを設定する必要があります。

コンソールから注釈を編集するか、次のようにコマンド ラインからパッチを適用することで、Tridentストレージ クラスとスナップショット クラスをデフォルトとして設定できます。

```
storageclass.kubernetes.io/is-default-class:true  
or  
kubectl patch storageclass standard -p '{"metadata": {"annotations": {"storageclass.kubernetes.io/is-default-class":"true"}}}'  
  
storageclass.kubevirt.io/is-default-virt-class: true  
or  
kubectl patch storageclass standard -p '{"metadata": {"annotations": {"storageclass.kubevirt.io/is-default-virt-class": "true"}}}'
```

これを設定すると、次のコマンドを使用して、既存の dv および VolumeSnapshot オブジェクトを削除できます。

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

## 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隨時、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 5225.227-19（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権を有し、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権については、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。