Upgrade Trident

Trident

NetApp
February 02, 2026

Table of Contents

Upgrade Trident

Upgrade Trident

Beginning with the 24.02 release, Trident follows a four-month release cadence, delivering three major releases every calendar year. Each new release builds on the previous releases and provides new features, performance enhancements, bug fixes, and improvements. We encourage you to upgrade at least once a year to take advantage of the new features in Trident.

Considerations before upgrading

When upgrading to the latest release of Trident, consider the following:

- There should be only one Trident instance installed across all the namespaces in a given Kubernetes cluster.
- Trident 23.07 and later requires v1 volume snapshots and no longer supports alpha or beta snapshots.
- When upgrading, it is important you provide `parameter.fsType` in `StorageClasses` used by Trident. You can delete and re-create `StorageClasses` without disrupting pre-existing volumes.
  - This is a **requirement** for enforcing security contexts for SAN volumes.
  - The sample input directory contains examples, such as `storage-class-basic.yaml.templ` and `storage-class-bronze-default.yaml`.
  - For more information, refer to Known Issues.

Step 1: Select a version

Trident versions follow a date-based `YY.MM` naming convention, where "YY" is the last two digits of the year and "MM" is the month. Dot releases follow a `YY.MM.X` convention, where "X" is the patch level. You will select the version to upgrade to based on the version you are upgrading from.

- You can perform a direct upgrade to any target release that is within a four-release window of your installed version. For example, you can directly upgrade from 24.06 (or any 24.06 dot release) to 25.06.
- If you are upgrading from a release outside of the four-release window, perform a multi-step upgrade. Use the upgrade instructions for the earlier version you are upgrading from to upgrade to the most recent release that fits the four-release window. For example, if you are running 23.07 and want to upgrade to 25.06:
  1. First upgrade from 23.07 to 24.06.
  2. Then upgrade from 24.06 to 25.06.

When upgrading using the Trident operator on OpenShift Container Platform, you should upgrade to Trident 21.01.1 or later. The Trident operator released with 21.01.0 contains a known issue that has been fixed in 21.01.1. For more details, refer to the issue details on GitHub.

Step 2: Determine the original installation method

To determine which version you used to originally install Trident:

1. Use `kubectl get pods -n trident` to examine the pods.
   - If there is no operator pod, Trident was installed using `tridentctl`.
   - If there is an operator pod, Trident was installed using the Trident operator either manually or using Helm.
2. If there is an operator pod, use `kubectl describe torc` to determine if Trident was installed using Helm.
   - If there is a Helm label, Trident was installed using Helm.
   - If there is no Helm label, Trident was installed manually using the Trident operator.

Step 3: Select an upgrade method

Generally, you should upgrade using the same method you used for the initial installation, however you can move between installation methods. There are two options to upgrade Trident.

- Upgrade using the Trident operator

  

  We suggest you review Understand the operator upgrade workflow before upgrading with the operator.

- Upgrade using `tridentctl`

Upgrade with the operator

Understand the operator upgrade workflow

Before using the Trident operator to upgrade Trident, you should understand the background processes that occur during upgrade. This includes changes to the Trident controller, controller Pod and node Pods, and node DaemonSet that enable rolling updates.

Trident operator upgrade handling

One of the many benefits of using the Trident operator to install and upgrade Trident is the automatic handling of Trident and Kubernetes objects without disrupting existing mounted volumes. In this way, Trident can support upgrades with zero downtime, or *rolling updates*. In particular, the Trident operator communicates with the Kubernetes cluster to:

- Delete and recreate the Trident Controller deployment and node DaemonSet.
- Replace the Trident Controller Pod and Trident Node Pods with new versions.
  - If a node is not updated, it does not prevent remaining nodes from being updated.
  - Only nodes with a running Trident Node Pod can mount volumes.

For more information about Trident architecture on the Kubernetes cluster, refer to Trident architecture.

Operator upgrade workflow

When you initiate an upgrade using the Trident operator:

1. The **Trident operator**:
   a. Detects the currently installed version of Trident (version *n*).
   b. Updates all Kubernetes objects including CRDs, RBAC, and Trident SVC.
   c. Deletes the Trident Controller deployment for version *n*.
   d. Creates the Trident Controller deployment for version *n+1*.
2. **Kubernetes** creates Trident Controller Pod for *n+1*.
3. The **Trident operator**:
   a. Deletes the Trident Node DaemonSet for *n*. The operator does not wait for Node Pod termination.
   b. Creates the Trident Node Daemonset for *n+1*.
4. **Kubernetes** creates Trident Node Pods on nodes not running Trident Node Pod *n*. This ensures there is never more than one Trident Node Pod, of any version, on a node.

Upgrade a Trident installation using Trident operator or Helm

You can upgrade Trident using the Trident operator either manually or using Helm. You can upgrade from a Trident operator installation to another Trident operator installation or upgrade from a `tridentctl` installation to a Trident operator version. Review Select an upgrade method before upgrading a Trident operator installation.

Upgrade a manual installation

You can upgrade from a cluster-scoped Trident operator installation to another cluster-scoped Trident operator installation. All Trident versions use a cluster-scoped operator.

> To upgrade from Trident that was installed using the namespace-scoped operator (versions 20.07 through 20.10), use the upgrade instructions for your installed version of Trident.

About this task

Trident provides a bundle file you can use to install the operator and create associated objects for your Kubernetes version.

- For clusters running Kubernetes 1.24, use bundle_pre_1_25.yaml.
- For clusters running Kubernetes 1.25 or later, use bundle_post_1_25.yaml.

Before you begin

Ensure you are using a Kubernetes cluster running a supported Kubernetes version.

Steps

1. Verify your Trident version:

```
./tridentctl -n trident version
```

2. Update the `operator.yaml`, `tridentorchestrator_cr.yaml`, and `post_1_25_bundle.yaml` with the registry and imagepaths for the version you are upgrading to (e.g.25.06), and the correct secret.
3. Delete the Trident operator that was used to install the current Trident instance. For example, if you are upgrading from 25.02, run the following command:

```
kubectl delete -f 25.02.0/trident-installer/deploy/<bundle.yaml> -n
trident
```

4. If you customized your initial installation using `TridentOrchestrator` attributes, you can edit the `TridentOrchestrator` object to modify the installation parameters. This might include changes made to specify mirrored Trident and CSI image registries for offline mode, enable debug logs, or specify image pull secrets.
5. Install Trident using the correct bundle YAML file for your environment, where <*bundle.yaml*> is `bundle_pre_1_25.yaml` or `bundle_post_1_25.yaml` based on your Kubernetes version. For example, if you are installing Trident 25.06.0, run the following command:

```
kubectl create -f 25.06.0/trident-installer/deploy/<bundle.yaml> -n
trident
```

6. Edit the trident torc to include the image 25.06.0.

Upgrade a Helm installation

You can upgrade a Trident Helm installation.

> When upgrading a Kubernetes cluster from 1.24 to 1.25 or later that has Trident installed, you must update values.yaml to set `excludePodSecurityPolicy` to `true` or add `--set excludePodSecurityPolicy=true` to the `helm upgrade` command before you can upgrade the cluster.

If you have already upgraded your Kubernetes cluster from 1.24 to 1.25 without upgrading the Trident helm, the helm upgrade fails. For the helm upgrade to go through, perform these steps as pre-requisites:

1. Install the helm-mapkubeapis plugin from https://github.com/helm/helm-mapkubeapis.
2. Perform a dry run for the Trident release in the namespace where Trident is installed. This lists out the resources, which will be cleaned up.

```
helm mapkubeapis --dry-run trident  --namespace trident
```

3. Perform a full run with helm to do the cleanup.

```
helm mapkubeapis trident  --namespace trident
```

Steps

1. If you installed Trident using Helm, you can use `helm upgrade trident netapp-trident/trident-operator --version 100.2506.0` to upgrade in one step. If you did not add the Helm repo or cannot use it to upgrade:
   a. Download the latest Trident release from the *Assets* section on GitHub.
   b. Use the `helm upgrade` command where `trident-operator-25.10.0.tgz` reflects the version that you want to upgrade to.

```
helm upgrade <name> trident-operator-25.10.0.tgz
```

> If you set custom options during the initial installation (such as specifying private, mirrored registries for Trident and CSI images), append the `helm upgrade` command using `--set` to ensure those options are included in the upgrade command, otherwise the values will reset to default.

2. Run `helm list` to verify that the chart and app version have both been upgraded. Run `tridentctl logs` to review any debug messages.

Upgrade from a `tridentctl` installation to Trident operator

You can upgrade to the latest release of the Trident operator from a `tridentctl` installation. The existing backends and PVCs will automatically be available.

> Before switching between installation methods, review Moving between installation methods.

Steps

1. Download the latest Trident release.

```
# Download the release required [25.10.0]
mkdir 25.10.0
cd 25.10.0
wget
https://github.com/NetApp/trident/releases/download/v25.10.0/trident-installer-25.10.0.tar.gz
tar -xf trident-installer-25.10.0.tar.gz
cd trident-installer
```

2. Create the `tridentorchestrator` CRD from the manifest.

```
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
```

3. Deploy the cluster-scoped operator in the same namespace.

```
kubectl create -f deploy/<bundle-name.yaml>

serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created

#Examine the pods in the Trident namespace
NAME                                  READY   STATUS    RESTARTS   AGE
trident-controller-79df798bdc-m79dc   6/6     Running   0          150d
trident-node-linux-xrst8              2/2     Running   0          150d
trident-operator-5574dbbc68-nthjv     1/1     Running   0          1m30s
```

4. Create a `TridentOrchestrator` CR for installing Trident.

```
cat deploy/crds/tridentorchestrator_cr.yaml
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident

kubectl create -f deploy/crds/tridentorchestrator_cr.yaml

#Examine the pods in the Trident namespace
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-79df798bdc-m79dc        6/6     Running   0          1m
trident-csi-xrst8                   2/2     Running   0          1m
trident-operator-5574dbbc68-nthjv   1/1     Running   0          5m41s
```

5. Confirm Trident was upgraded to the intended version.

```
kubectl describe torc trident | grep Message -A 3

Message:                Trident installed
Namespace:              trident
Status:                 Installed
Version:                v25.10.0
```

Upgrade with tridentctl

You can easily upgrade an existing Trident installation using `tridentctl`.

About this task

Uninstalling and reinstalling Trident acts as an upgrade. When you uninstall Trident, the Persistent Volume Claim (PVC) and Persistent Volume (PV) used by the Trident deployment are not deleted. PVs that have already been provisioned will remain available while Trident is offline, and Trident will provision volumes for any PVCs that are created in the interim after it is back online.

Before you begin

Review Select an upgrade method before upgrading using `tridentctl`.

Steps

1. Run the uninstall command in `tridentctl` to remove all of the resources associated with Trident except for the CRDs and related objects.

```
./tridentctl uninstall -n <namespace>
```

2. Reinstall Trident. Refer to Install Trident using tridentctl.

Do not interrupt the upgrade process. Ensure the installer runs to completion.

Copyright information

Trademark information