Upgrade Astra Trident

Astra Trident

NetApp
March 05, 2024

This PDF was generated from https://docs.netapp.com/us-en/trident/trident-managing-k8s/upgrade-trident.html on March 05, 2024. Always check docs.netapp.com for the latest.
Upgrade Astra Trident

Considerations before upgrading

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

- There should be only one Astra Trident instance installed across all the namespaces in a given Kubernetes cluster.
- Astra Trident 23.07 and later requires v1 volume snapshots and no longer supports alpha or beta snapshots.
- If you created Cloud Volumes Service for Google Cloud in the CVS service type, you must update the backend configuration to use the standardsw or zoneredundantstandardsw service level when upgrading from Astra Trident 23.01. Failure to update the serviceLevel in the backend could cause volumes to fail. Refer to CVS service type samples for details.
- When upgrading, it is important you provide parameter.fsType in StorageClasses used by Astra 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

Astra 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 22.10 (or any 22.10 dot release) to 24.02.
- 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 21.10 and want to upgrade to 24.02:
  1. First upgrade from 21.10 to 22.10.
  2. Then upgrade from 22.10 to 24.02.

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 Astra Trident:

1. Use `kubectl get pods -n trident` to examine the pods.
   - If there is no operator pod, Astra Trident was installed using `tridentctl`.
   - If there is an operator pod, Astra Trident was installed using the Trident operator either manually or using Helm.

2. If there is an operator pod, use `kubectl describe tproc trident` to determine if Astra Trident was installed using Helm.
   - If there is a Helm label, Astra Trident was installed using Helm.
   - If there is no Helm label, Astra 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 Astra 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 Astra 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 Astra Trident is the automatic handling of Astra Trident and Kubernetes objects without disrupting existing mounted volumes. In this way, Astra 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.
Operator upgrade workflow

When you initiate an upgrade using the Trident operator:

1. The **Trident operator**:
   a. Detects the currently installed version of Astra 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 operator installation

You can upgrade Astra 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 Astra Trident versions 21.01 and above use a cluster-scoped operator.

To upgrade from Astra Trident that was installed using the namespace-scoped operator (versions 20.07 through 20.10), use the upgrade instructions for your installed version of Astra 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 or earlier, 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 Astra Trident version:
2. Delete the Trident operator that was used to install the current Astra Trident instance. For example, if you are upgrading from 23.07, run the following command:

```
kubectl delete -f 23.07.0/trident-installer/deploy/<bundle.yaml> -n trident
```

3. 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.

4. Install Astra 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 Astra Trident 24.02, run the following command:

```
kubectl create -f 24.02.0/trident-installer/deploy/<bundle.yaml> -n trident
```

### Upgrade a Helm installation

You can upgrade an Astra Trident Helm installation.

![Tip]

When upgrading a Kubernetes cluster from 1.24 to 1.25 or later that has Astra 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.

**Steps**

1. If you installed Astra Trident using Helm, you can use `helm upgrade trident netapp-trident/trident-operator --version 24.02.0` to upgrade in one step. If you did not add the Helm repo or cannot use it to upgrade:
   a. Download the latest Astra Trident release from the [Assets section on GitHub](https://github.com/netapp/traienterprise/tree/master/releases).
   b. Use the `helm upgrade` command where `trident-operator-24.02.0.tgz` reflects the version that you want to upgrade to.

```
helm upgrade <name> trident-operator-24.02.0.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.

ℹ️ Before switching between installation methods, review [Moving between installation methods](#).

**Steps**

1. Download the latest Astra Trident release.

   ```
   # Download the release required [24.020.0]
   mkdir 24.02.0
   cd 24.02.0
   wget https://github.com/NetApp/trident/releases/download/v24.02.0/trident-installer-24.02.0.tar.gz
   tar -xf trident-installer-24.02.0.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
   ```

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>trident-controller-79df798bdc-m79dc</td>
<td>6/6</td>
<td>Running</td>
<td>0</td>
<td>150d</td>
</tr>
<tr>
<td>trident-node-linux-xrst8</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>150d</td>
</tr>
<tr>
<td>trident-operator-5574dbbc68-nthjv</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m30s</td>
</tr>
</tbody>
</table>

4. Create a `TridentOrchestrator` CR for installing Astra 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

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>trident-csi-79df798bdc-m79dc</td>
<td>6/6</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>trident-csi-xrst8</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>trident-operator-5574dbbc68-nthjv</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>5m41s</td>
</tr>
</tbody>
</table>

5. Confirm Trident was upgraded to the intended version.

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

<table>
<thead>
<tr>
<th>Message:</th>
<th>Trident installed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Namespace:</td>
<td>trident</td>
</tr>
<tr>
<td>Status:</td>
<td>Installed</td>
</tr>
<tr>
<td>Version:</td>
<td>v24.02.0</td>
</tr>
</tbody>
</table>

**Upgrade with tridentctl**

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

**About this task**

Uninstalling and reinstalling Astra Trident acts as an upgrade. When you uninstall Trident, the Persistent Volume Claim (PVC) and Persistent Volume (PV) used by the Astra Trident deployment are not deleted. PVs that have already been provisioned will remain available while Astra Trident is offline, and Astra Trident will provision volumes for any PVCs that are created in the interim once 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 Astra Trident except for the CRDs and related objects.
2. Reinstall Astra Trident. Refer to Install Astra Trident using tridentctl.

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