



Install using tridentctl

Trident

NetApp
October 30, 2026

Table of Contents

- Install using tridentctl 1
 - Install Trident using tridentctl 1
 - Critical information about Trident10 1
 - Install Trident using tridentctl 1
 - Step 1: Download the Trident installer package 2
 - Step 2: Install Trident 2
 - Verify the installation 3
 - Sample configurations 4
 - Customize Trident tridentctl installation 4
 - Learn about the installer 4

Install using tridentctl

Install Trident using tridentctl

You can install Trident using `tridentctl`. This process applies to installations where the container images required by Trident are stored either in a private registry or not. To customize your `tridentctl` deployment, refer to [Customize tridentctl deployment](#).

Critical information about Trident10

You must read the following critical information about Trident.

▼ Critical information about Trident

- Kubernetes 1.27 is now supported in Trident. Upgrade Trident prior to upgrading Kubernetes.
- Trident strictly enforces the use of multipathing configuration in SAN environments, with a recommended value of `find_multipaths: no` in `multipath.conf` file.

Use of non-multipathing configuration or use of `find_multipaths: yes` or `find_multipaths: smart` value in `multipath.conf` file will result in mount failures. Trident has recommended the use of `find_multipaths: no` since the 21.07 release.

Install Trident using tridentctl

Review [the installation overview](#) to ensure you've met installation prerequisites and selected the correct installation option for your environment.

Before you begin

Before you begin installation, log in to the Linux host and verify it is managing a working, [supported Kubernetes cluster](#) and that you have the necessary privileges.



With OpenShift, use `oc` instead of `kubectl` in all of the examples that follow, and log in as **system:admin** first by running `oc login -u system:admin` or `oc login -u kube-admin`.

▼ Details

1. Verify your Kubernetes version:

```
kubectl version
```

2. Verify cluster administrator privileges:

```
kubectl auth can-i '*' '*' --all-namespaces
```

3. Verify you can launch a pod that uses an image from Docker Hub and reach your storage system over the pod network:

```
kubectl run -i --tty ping --image=busybox --restart=Never --rm -- \
ping <management IP>
```

Step 1: Download the Trident installer package

The Trident installer package creates a Trident pod, configures the CRD objects that are used to maintain its state, and initializes the CSI sidecars to perform actions such as provisioning and attaching volumes to the cluster hosts. Download and extract the latest version of the Trident installer from [the Assets section on GitHub](#). Update `<trident-installer-XX.XX.X.tar.gz>` in the example with your selected Trident version.

```
wget https://github.com/NetApp/trident/releases/download/v26.06.0/trident-
installer-26.06.0.tar.gz
tar -xf trident-installer-26.06.0.tar.gz
cd trident-installer
```

Step 2: Install Trident

Install Trident in the desired namespace by executing the `tridentctl install` command. You can add additional arguments to specify image registry location.

Standard mode

```
./tridentctl install -n trident
```

Images in one registry

```
./tridentctl install -n trident --image-registry <your-registry>
--autosupport-image <your-registry>/trident-autosupport:26.06 --trident
-image <your-registry>/trident:26.06.0
```

Images in different registries

```
./tridentctl install -n trident --image-registry <your-registry>
--autosupport-image <your-registry>/trident-autosupport:26.06 --trident
-image <your-registry>/trident:26.06.0
```

Your installation status should look something like this.

```
....
INFO Starting Trident installation.                namespace=trident
INFO Created service account.
INFO Created cluster role.
```

```

INFO Created cluster role binding.
INFO Added finalizers to custom resource definitions.
INFO Created Trident service.
INFO Created Trident secret.
INFO Created Trident deployment.
INFO Created Trident daemonset.
INFO Waiting for Trident pod to start.
INFO Trident pod started.                                namespace=trident
pod=trident-controller-679648bd45-cv2mx
INFO Waiting for Trident REST interface.
INFO Trident REST interface is up.                        version=26.10.0
INFO Trident installation succeeded.
....

```

Verify the installation

You can verify your installation using pod creation status or `tridentctl`.

Using pod creation status

You can confirm if the Trident installation completed by reviewing the status of the created pods:

```
kubectl get pods -n trident
```

NAME	READY	STATUS	RESTARTS	AGE
trident-controller-679648bd45-cv2mx	6/6	Running	0	5m29s
trident-node-linux-vgc8n	2/2	Running	0	5m29s



If the installer does not complete successfully or `trident-controller-<generated id>` (`trident-csi-<generated id>` in versions prior to 23.01) does not have a **Running** status, the platform was not installed. Use `-d` to [turn on debug mode](#) and troubleshoot the issue.

Using `tridentctl`

You can use `tridentctl` to check the version of Trident installed.

```

./tridentctl -n trident version

+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 26.06.0       | 26.06.0       |
+-----+-----+

```

Sample configurations

The following examples provide sample configurations for installing Trident using `tridentctl`.

▼ Windows nodes

To enable Trident to run on Windows nodes:

```
tridentctl install --windows -n trident
```

▼ Force detach

For information, see [Automating the failover of stateful applications with Trident](#).

```
tridentctl install --enable-force-detach=true -n trident
```

▼ Enable concurrent Trident controller operations

To enable concurrent Trident controller operations for improved throughput, add the `--enable-concurrency` option during the installation as shown in this example. For more information on supported drivers, see the xref:./trident-install/ [Controller Scalability documentation](#).

```
tridentctl install --enable-concurrency -n trident
```

Customize Trident tridentctl installation

You can use the Trident installer to customize installation.

Learn about the installer

The Trident installer enables you to customize attributes. For example, if you have copied the Trident image to a private repository, you can specify the image name by using `--trident-image`. If you have copied the Trident image as well as the needed CSI sidecar images to a private repository, it might be preferable to specify the location of that repository by using the `--image-registry` switch, which takes the form `<registry FQDN>[:port]`.



When installing Trident in a private repository, if you are using the `--image-registry` switch to specify the repository location, do not use `/netapp/` in the repository path. For example:

```
./tridentctl install --image-registry <image-registry> -n <namespace>
```

If you are using a distribution of Kubernetes, where `kubelet` keeps its data on a path other than the usual `/var/lib/kubelet`, you can specify the alternate path by using `--kubelet-dir`.

If you need to customize the installation beyond what the installer's arguments allow, you can also customize the deployment files. Using the `--generate-custom-yaml` parameter creates the following YAML files in the installer's setup directory:

- `trident-clusterrolebinding.yaml`

- trident-deployment.yaml
- trident-crds.yaml
- trident-clusterrole.yaml
- trident-daemonset.yaml
- trident-service.yaml
- trident-namespace.yaml
- trident-serviceaccount.yaml
- trident-resourcequota.yaml
- *

After you have generated these files, you can modify them according to your needs and then use `--use -custom-yaml` to install your custom deployment.

```
./tridentctl install -n trident --use-custom-yaml
```

Copyright information

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.