Install Trident protect

Trident

NetApp
February 20, 2025

This PDF was generated from https://docs.netapp.com/us-en/trident/trident-protect/trident-protect-requirements.html on February 20, 2025. Always check docs.netapp.com for the latest.

Table of Contents

Install Trident protect

Trident protect requirements

Get started by verifying the readiness of your operational environment, application clusters, applications, and licenses. Ensure that your environment meets these requirements to deploy and operate Trident protect.

Trident protect Kubernetes cluster compatibility

Trident protect is compatible with a wide range of fully managed and self-managed Kubernetes offerings, including:

- Amazon Elastic Kubernetes Service (EKS)
- Google Kubernetes Engine (GKE)
- Microsoft Azure Kubernetes Service (AKS)
- Red Hat OpenShift
- SUSE Rancher
- VMware Tanzu Portfolio
- Upstream Kubernetes

> ⓘ Ensure that the cluster on which you install Trident protect is configured with a running snapshot controller and the related CRDs. To install a snapshot controller, refer to these instructions.

Trident protect storage backend compatibility

Trident protect supports the following storage backends:

- Amazon FSx for NetApp ONTAP
- Cloud Volumes ONTAP
- ONTAP storage arrays
- Google Cloud NetApp Volumes
- Azure NetApp Files

Ensure that your storage backend meets the following requirements:

- Ensure that NetApp storage connected to the cluster is using Astra Trident 24.02 or newer (Trident 24.10 is recommended).
  - If Astra Trident is older than version 24.06.1 and you plan to use NetApp SnapMirror disaster recovery functionality, you need to manually enable Astra Control Provisioner.
- Ensure that you have the latest Astra Control Provisioner (installed and enabled by default as of Astra Trident 24.06.1).
- Ensure that you have a NetApp ONTAP storage backend.
- Ensure that you have configured an object storage bucket for storing backups.

- Create any application namespaces that you plan to use for applications or application data management operations. Trident protect does not create these namespaces for you; if you specify a nonexistent namespace in a custom resource, the operation will fail.

Requirements for nas-economy volumes

Trident protect supports backup and restore operations to nas-economy volumes. Snapshots, clones, and SnapMirror replication to nas-economy volumes are not currently supported. You need to enable a snapshot directory for each nas-economy volume you plan to use with Trident protect.

Some applications are not compatible with volumes that use a snapshot directory. For these applications, you need to hide the snapshot directory by running the following command on the ONTAP storage system:

```
nfs modify -vserver <svm> -v3-hide-snapshot enabled
```

You can enable the snapshot directory by running the following command for each nas-economy volume, replacing `<volume-UUID>` with the UUID of the volume you want to change:

```
tridentctl update volume <volume-UUID> --snapshot-dir=true --pool-level
=true -n trident
```

You can enable snapshot directories by default for new volumes by setting the Trident backend configuration option `snapshotDir` to `true`. Existing volumes are not affected.

Protecting data with KubeVirt VMs

Trident protect 24.10 and 24.10.1 and newer have different behavior when you protect applications running on KubeVirt VMs. For both versions, you can enable or disable filesystem freezing and unfreezing during data protection operations.

For all Trident protect versions, to enable or disable automatic freeze functionality in OpenShift environments, you might need to grant the application namespace privileged permissions. For example:

```
oc adm policy add-scc-to-user privileged -z default -n
<application-namespace>
```

Trident protect 24.10

Trident protect 24.10 does not automatically ensure a consistent state for KubeVirt VM filesystems during data protection operations. If you want to protect your KubeVirt VM data using Trident protect 24.10, you need to manually enable the freeze/unfreeze functionality for the filesystems before the data protection operation. This ensures that the filesystems are in a consistent state.

You can configure Trident protect 24.10 to manage the freezing and unfreezing of the VM filesystem during data protection operations by configuring virtualization and then using the following command:

```
kubectl set env deployment/trident-protect-controller-manager
NEPTUNE_VM_FREEZE=true -n trident-protect
```

Trident protect 24.10.1 and newer

Beginning with Trident protect 24.10.1, Trident protect automatically freezes and unfreezes KubeVirt filesystems during data protection operations. Optionally, you can disable this automatic behavior using the following command:

```
kubectl set env deployment/trident-protect-controller-manager
NEPTUNE_VM_FREEZE=false -n trident-protect
```

Requirements for SnapMirror replication

NetApp SnapMirror is available for use with Trident protect for the following ONTAP solutions:

- NetApp ASA
- NetApp AFF
- NetApp FAS
- NetApp ONTAP Select
- NetApp Cloud Volumes ONTAP
- Amazon FSx for NetApp ONTAP

ONTAP cluster requirements for SnapMirror replication

Ensure your ONTAP cluster meets the following requirements if you plan to use SnapMirror replication:

- **Astra Control Provisioner or Trident**: Astra Control Provisioner or Trident must exist on both the source and destination Kubernetes clusters that utilize ONTAP as a backend. Trident protect supports replication with NetApp SnapMirror technology using storage classes backed by the following drivers:
  - `ontap-nas`
  - `ontap-san`
- **Licenses**: ONTAP SnapMirror asynchronous licenses using the Data Protection bundle must be enabled on both the source and destination ONTAP clusters. Refer to SnapMirror licensing overview in ONTAP for more information.

Peering considerations for SnapMirror replication

Ensure your environment meets the following requirements if you plan to use storage backend peering:

- **Cluster and SVM**: The ONTAP storage backends must be peered. Refer to Cluster and SVM peering overview for more information.

  > ⓘ Ensure that the SVM names used in the replication relationship between two ONTAP clusters are unique.

- **Astra Control Provisioner or Trident and SVM**: The peered remote SVMs must be available to Astra Control Provisioner or Trident on the destination cluster.
- **Managed backends**: You need to add and manage ONTAP storage backends in Trident protect to create a replication relationship.
- **NVMe over TCP**: Trident protect does not support NetApp SnapMirror replication for storage backends that are using the NVMe over TCP protocol.

Trident / ONTAP configuration for SnapMirror replication

Trident protect requires that you configure at least one storage backend that supports replication for both the source and destination clusters. If the source and destination clusters are the same, the destination application should use a different storage backend than the source application for the best resiliency.

Install and configure Trident protect

If your environment meets the requirements for Trident protect, you can follow these steps to install Trident protect on your cluster. You can obtain Trident protect from NetApp, or install it from your own private registry. Installing from a private registry is helpful if your cluster cannot access the Internet.

> ⓘ By default, Trident protect collects support information that helps with any NetApp support cases that you might open, including logs, metrics, and topology information about clusters and managed applications. Trident protect sends these support bundles to NetApp on a daily schedule. You can optionally disable this support bundle collection when you install Trident protect. You can manually generate a support bundle at any time.

Install Trident protect

Install Trident protect from NetApp

Steps

1. Add the Trident Helm repository:

   ```
   helm repo add netapp-trident-protect
   https://netapp.github.io/trident-protect-helm-chart
   ```

2. Install the Trident protect CRDs:

   ```
   helm install trident-protect-crds netapp-trident-protect/trident-
   protect-crds --version 100.2410.1 --create-namespace --namespace
   trident-protect
   ```

3. Use Helm to install Trident protect using one of the following commands. Replace `<name_of_cluster>` with a cluster name, which will be assigned to the cluster and used to identify the cluster's backups and snapshots:

   - Install Trident protect normally:

     ```
     helm install trident-protect netapp-trident-protect/trident-
     protect --set clusterName=<name_of_cluster> --version 100.2410.1
     --create-namespace --namespace trident-protect
     ```

   - Install Trident protect and disable the scheduled daily Trident protect AutoSupport support bundle uploads:

     ```
     helm install trident-protect netapp-trident-protect/trident-
     protect --set autoSupport.enabled=false --set
     clusterName=<name_of_cluster> --version 100.2410.1 --create
     -namespace --namespace trident-protect
     ```

Install Trident protect from a private registry

You can install Trident protect from a private image registry if your Kubernetes cluster is unable to access the Internet. In these examples, replace values in brackets with information from your environment:

Steps

1. Pull the following images to your local machine, update the tags, and then push them to your private registry:

```
netapp/controller:24.10.1
netapp/restic:24.10.1
netapp/kopia:24.10.1
netapp/trident-autosupport:24.10.0
netapp/exechook:24.10.1
netapp/resourcebackup:24.10.1
netapp/resourcerestore:24.10.1
netapp/resourcedelete:24.10.1
bitnami/kubectl:1.30.2
kubebuilder/kube-rbac-proxy:v0.16.0
```

For example:

```
docker pull netapp/controller:24.10.1
```

```
docker tag netapp/controller:24.10.1 <private-registry-
url>/controller:24.10.1
```

```
docker push <private-registry-url>/controller:24.10.1
```

2. Create the Trident protect system namespace:

```
kubectl create ns trident-protect
```

3. Log in to the registry:

```
helm registry login <private-registry-url> -u <account-id> -p <api-
token>
```

4. Create a pull secret to use for private registry authentication:

```
kubectl create secret docker-registry regcred --docker
-username=<registry-username> --docker-password=<api-token> -n
trident-protect --docker-server=<private-registry-url>
```

5. Add the Trident Helm repository:

```
helm repo add netapp-trident-protect
https://netapp.github.io/trident-protect-helm-chart
```

6. Create a file named `protectValues.yaml`. Ensure that it contains the following Trident protect settings:

```yaml
---
image:
  registry: <private-registry-url>
imagePullSecrets:
  - name: regcred
controller:
  image:
    registry: <private-registry-url>
rbacProxy:
  image:
    registry: <private-registry-url>
crCleanup:
  imagePullSecrets:
    - name: regcred
webhooksCleanup:
  imagePullSecrets:
    - name: regcred
```

7. Install the Trident protect CRDs:

```
helm install trident-protect-crds netapp-trident-protect/trident-
protect-crds --version 100.2410.1 --create-namespace --namespace
trident-protect
```

8. Use Helm to install Trident protect using one of the following commands. Replace `<name_of_cluster>` with a cluster name, which will be assigned to the cluster and used to identify the cluster’s backups and snapshots:

   - Install Trident protect normally:

     ```
     helm install trident-protect netapp-trident-protect/trident-
     protect --set clusterName=<name_of_cluster> --version 100.2410.1
     --create-namespace --namespace trident-protect -f
     protectValues.yaml
     ```

   - Install Trident protect and disable the scheduled daily Trident protect AutoSupport support bundle uploads:

```
helm install trident-protect netapp-trident-protect/trident-
protect --set autoSupport.enabled=false --set
clusterName=<name_of_cluster> --version 100.2410.1 --create
-namespace --namespace trident-protect -f protectValues.yaml
```

Specify Trident protect container resource limits

You can use a configuration file to specify resource limits for Trident protect containers after you install Trident protect. Setting resource limits enables you to control how much of the cluster's resources are consumed by Trident protect operations.

Steps

1. Create a file named `resourceLimits.yaml`.
2. Populate the file with resource limit options for Trident protect containers according to the needs of your environment.

   The following example configuration file shows the available settings and contains the default vaules for each resource limit:

```yaml
---
jobResources:
  defaults:
    limits:
      cpu: 8000m
      memory: 10000Mi
      ephemeralStorage: ""
    requests:
      cpu: 100m
      memory: 100Mi
      ephemeralStorage: ""
  resticVolumeBackup:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  resticVolumeRestore:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  kopiaVolumeBackup:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  kopiaVolumeRestore:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
```

3. Apply the values from the `resourceLimits.yaml` file:

```
helm upgrade trident-protect -n trident-protect -f <resourceLimits.yaml>
--reuse-values
```

Install the Trident protect CLI plugin

You can use the Trident protect command line plugin, which is an extension of the Trident `tridentctl` utility, to create and interact with Trident protect custom resources (CRs).

Install the Trident protect CLI plugin

Before using the command line utility, you need to install it on the machine you use to access your cluster. Follow these steps, depending on if your machine uses an x64 or ARM CPU.

Download plugin for Linux AMD64 CPUs

Steps

1. Download the Trident protect CLI plugin:

   ```
   curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/24.10.1/tridentctl-protect-linux-amd64
   ```

Download plugin for Linux ARM64 CPUs

Steps

1. Download the Trident protect CLI plugin:

   ```
   curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/24.10.1/tridentctl-protect-linux-arm64
   ```

Download plugin for Mac AMD64 CPUs

Steps

1. Download the Trident protect CLI plugin:

   ```
   curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/24.10.1/tridentctl-protect-macos-amd64
   ```

Download plugin for Mac ARM64 CPUs

Steps

1. Download the Trident protect CLI plugin:

   ```
   curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/24.10.1/tridentctl-protect-macos-arm64
   ```

2. Enable execute permissions for the plugin binary:

   ```
   chmod +x tridentctl-protect
   ```

3. Copy the plugin binary to a location that is defined in your PATH variable. For example, `/usr/bin` or `/usr/local/bin` (you might need elevated privileges):

   ```
   cp ./tridentctl-protect /usr/local/bin/
   ```

4. Optionally, you can copy the plugin binary to a location in your home directory. In this case, it is recommended to ensure the location is part of your PATH variable:

   ```
   cp ./tridentctl-protect ~/bin/
   ```

> Copying the plugin to a location in your PATH variable enables you to use the plugin by typing `tridentctl-protect` or `tridentctl protect` from any location.

View Trident CLI plugin help

You can use the built-in plugin help features to get detailed help on the capabilities of the plugin:

Steps

1. Use the help function to view usage guidance:

   ```
   tridentctl-protect help
   ```

Enable command auto-completion

After you have installed the Trident protect CLI plugin, you can enable auto-completion for certain commands.

Enable auto-completion for the Bash shell

Steps

1. Download the completion script:

   ```
   curl -L -O https://github.com/NetApp/tridentctl-
   protect/releases/download/24.10.1/tridentctl-completion.bash
   ```

2. Make a new directory in your home directory to contain the script:

   ```
   mkdir -p ~/.bash/completions
   ```

3. Move the downloaded script to the `~/.bash/completions` directory:

   ```
   mv tridentctl-completion.bash ~/.bash/completions/
   ```

4. Add the following line to the `~/.bashrc` file in your home directory:

   ```
   source ~/.bash/completions/tridentctl-completion.bash
   ```

Enable auto-completion for the Z shell

Steps

1. Download the completion script:

   ```
   curl -L -O https://github.com/NetApp/tridentctl-
   protect/releases/download/24.10.1/tridentctl-completion.zsh
   ```

2. Make a new directory in your home directory to contain the script:

   ```
   mkdir -p ~/.zsh/completions
   ```

3. Move the downloaded script to the `~/.zsh/completions` directory:

   ```
   mv tridentctl-completion.zsh ~/.zsh/completions/
   ```

4. Add the following line to the `~/.zprofile` file in your home directory:

   ```
   source ~/.zsh/completions/tridentctl-completion.zsh
   ```

Result

Upon your next shell login, you can use command auto-completion with the tridentctl-protect plugin.

Copyright information

Copyright © 2025 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.