



升级版Trident Trident

NetApp
January 15, 2026

目录

- 升级版Trident 1
 - 升级版Trident 1
 - 升级前的注意事项 1
 - 步骤 1：选择版本 1
 - 步骤二：确定原始安装方法 2
 - 步骤 3：选择升级方式 2
 - 通过运营商进行升级 2
 - 了解操作员升级工作流程 2
 - 使用Trident Operator 或 Helm 升级Trident安装 3
 - 使用 tridentctl 进行升级 7

升级版Trident

升级版Trident

从 24.02 版本开始，Trident遵循四个月的发布节奏，每年发布三个主要版本。每个新版本都以之前的版本为基础，并提供新功能、性能增强、错误修复和改进。我们建议您至少每年升级一次，以充分利用Trident的新功能。

升级前的注意事项

升级到最新版Trident时，请注意以下事项：

- 在给定的 Kubernetes 集群的所有命名空间中，应该只安装一个Trident实例。
- Trident 23.07 及更高版本需要 v1 卷快照，不再支持 alpha 或 beta 快照。
- 如果您在 Google Cloud 中创建了Cloud Volumes Service，"[CVS 服务类型](#)"您必须更新后端配置才能使用 ``standardsw`` 或者 ``zoneredundantstandardsw`` 从Trident 23.01 升级时的服务级别。未能更新 ``serviceLevel`` 后端故障可能导致卷出现故障。参考 "[CVS 服务类型示例](#)" 了解详情。
- 升级时，请务必提供以下信息 ``parameter.fsType`` 在 ``StorageClasses`` 由Trident使用。您可以删除并重新创建 ``StorageClasses`` 在不影响原有流量的情况下。
 - 这是强制执行的要求 "[安全上下文](#)"适用于 SAN 卷。
 - <https://github.com/NetApp/trident/tree/master/trident-installer/sample-input> [示例输入^] 目录包含示例，例如<https://github.com/NetApp/trident/blob/master/trident-installer/sample-input/storage-class-samples/storage-class-basic.yaml.template> [storage-class-basic.yaml.template ^] 和链接：[storage-class-bronze-default.yaml](#)。
 - 更多信息，请参阅"[已知问题](#)"。

步骤 1：选择版本

Trident 的版本遵循基于日期的命名规则。`YY.MM` 命名规则中，“YY”是年份的最后两位数字，“MM”是月份。Dot 版本遵循 `YY.MM.X` 按照惯例，其中“X”是补丁级别。您需要根据当前版本选择要升级到的版本。

- 您可以直接升级到与已安装版本相差不超过四个版本窗口内的任何目标版本。例如，您可以直接从 24.06（或任何 24.06 的小版本）升级到 25.06。
- 如果您要从四版本窗口之外的版本升级，请执行多步骤升级。请按照升级说明进行操作。"[早期版本](#)"您正在从当前版本升级到符合四版本发布窗口的最新版本。例如，如果您当前运行的是 23.07 版本，并且想要升级到 25.06 版本：
 - a. 首次升级从 23.07 升级到 24.06。
 - b. 然后从 24.06 升级到 25.06。



在 OpenShift 容器平台上使用Trident操作符进行升级时，应升级到Trident 21.01.1 或更高版本。随 21.01.0 版本发布的Trident操作符存在一个已知问题，该问题已在 21.01.1 版本中修复。更多详情请参阅..... "[GitHub 上的问题详情](#)"。

步骤二：确定原始安装方法

要确定您最初安装Trident时使用的版本：

1. 使用 `kubectl get pods -n trident` 检查豆荚。
 - 如果没有操作员舱，Trident是通过以下方式安装的： `tridentctl`。
 - 如果存在 operator pod，则Trident是通过Trident operator 手动安装的，或者使用 Helm 安装的。
2. 如果有操作员舱，请使用 `kubectl describe torc` 确定Trident是否使用 Helm 安装。
 - 如果存在 Helm 标签，则表示Trident是使用 Helm 安装的。
 - 如果没有 Helm 标签，则表示Trident是使用Trident操作员手动安装的。

步骤 3：选择升级方式

通常情况下，您应该使用与初始安装相同的方法进行升级，但是您也可以.....["安装方法之间的转换"](#)。升级Trident有两种选择。

- ["使用Trident操作员进行升级"](#)



我们建议您查看["了解操作员升级工作流程"](#)在与运营商升级之前。

*

通过运营商进行升级

了解操作员升级工作流程

在使用Trident操作员升级Trident之前，您应该了解升级期间发生的后台进程。这包括对Trident控制器、控制器 Pod 和节点 Pod 以及节点 DaemonSet 的更改，以启用滚动更新。

Trident操作员升级处理

众多之一["使用Trident运算符的好处"](#)安装和升级Trident 的过程是自动处理Trident和 Kubernetes 对象，而不会中断现有的已挂载卷。这样一来，Trident就能支持零停机时间的升级，或者说，["滚动更新"](#)。具体来说，Trident 操作符与 Kubernetes 集群通信以：

- 删除并重新创建Trident Controller 部署和节点 DaemonSet。
- 将Trident Controller Pod 和Trident Node Pod 替换为新版本。
 - 如果某个节点没有更新，并不妨碍其他节点的更新。
 - 只有运行了Trident Node Pod 的节点才能挂载卷。



有关 Kubernetes 集群上Trident架构的更多信息，请参阅：["Trident架构"](#)。

操作员升级工作流程

当您使用Trident操作符发起升级时：

1. Trident运算符：
 - a. 检测当前安装的Trident版本（版本 n ）。
 - b. 更新所有 Kubernetes 对象，包括 CRD、RBAC 和Trident SVC。
 - c. 删除版本为 n 的Trident Controller 部署。
 - d. 创建版本 $n+1$ 的Trident Controller 部署。
2. **Kubernetes** 为 $n+1$ 创建Trident Controller Pod。
3. Trident运算符：
 - a. 删除 n 的Trident节点守护进程集。操作员不会等待节点 Pod 终止。
 - b. 为 $n+1$ 创建Trident节点守护进程集。
4. **Kubernetes** 在未运行Trident Node Pod n 的节点上创建Trident Node Pod。这样可以确保一个节点上永远不会有超过一个Trident Node Pod，无论版本如何。

使用Trident Operator 或 Helm 升级Trident安装

您可以使用Trident Operator 手动升级Trident ，也可以使用 Helm 进行升级。您可以从一个Trident Operator 安装升级到另一个Trident Operator 安装，或者从一个 Trident Operator 安装升级到另一个 Trident Operator 安装。`tridentctl`安装到Trident操作员版本。审查["选择升级方式"](#)在升级Trident操作程序之前。

升级手动安装

您可以从集群范围的Trident操作员安装升级到另一个集群范围的Trident操作员安装。所有Trident版本都使用集群范围的操作符。



要从使用命名空间作用域运算符安装的Trident （版本 20.07 至 20.10）进行升级，请使用以下升级说明：["您已安装的版本"](#)Trident。

关于此任务

Trident提供了一个捆绑文件，您可以使用该文件安装操作员并为您的 Kubernetes 版本创建关联对象。

- 对于运行 Kubernetes 1.24 的集群，请使用["bundle_pre_1_25.yaml"](#)。
- 对于运行 Kubernetes 1.25 或更高版本的集群，请使用["bundle_post_1_25.yaml"](#)。

开始之前

请确保您使用的是正在运行的 Kubernetes 集群。["受支持的 Kubernetes 版本"](#)。

步骤

1. 请验证您的Trident版本：

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

2. 更新 `operator.yaml`, `tridentorchestrator_cr.yaml`, 和 `'post_1_25_bundle.yaml'` 使用要升级到的版本（例如 25.06）的注册表和映像路径，以及正确的密钥。
3. 删除用于安装当前Trident实例的Trident操作员。例如，如果您从 25.02 升级，请运行以下命令：

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

4. 如果您使用自定义方式对初始安装进行了设置 `'TridentOrchestrator'` 属性，您可以编辑 `'TridentOrchestrator'` 用于修改安装参数的对象。这可能包括为离线模式指定镜像Trident和 CSI 映像注册表、启用调试日志或指定映像拉取密钥所做的更改。
5. 使用适用于您环境的正确 `bundle.yaml` 文件安装Trident，其中 `<bundle.yaml>` 是 `'bundle_pre_1_25.yaml'` 或者 `'bundle_post_1_25.yaml'` 根据您的 Kubernetes 版本。例如，如果您正在安装Trident 25.06.0，请运行以下命令：

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

6. 编辑三叉戟项圈以包含图像 25.06.0。

升级 Helm 安装

您可以升级Trident Helm 安装。



当将已安装Trident的Kubernetes 集群从 1.24 版本升级到 1.25 或更高版本时，必须更新 `values.yaml` 文件进行设置。将 `'excludePodSecurityPolicy'` 到 `'true'` 或添加 `'--set excludePodSecurityPolicy=true'` 到 `'helm upgrade'` 升级集群前必须先执行此命令。

如果您已将 Kubernetes 集群从 1.24 升级到 1.25，但未升级Trident helm，则 helm 升级将失败。要成功升级Helm，请先执行以下步骤：

1. 从以下位置安装 `helm-mapkubeapis` 插件 <https://github.com/helm/helm-mapkubeapis>。
2. 在Trident安装所在的命名空间中，对Trident版本执行一次试运行。这里列出了需要清理的资源。

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

3. 使用 Helm 执行完整运行以进行清理。

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

步骤

1. 如果你"使用 Helm 安装了Trident"。你可以使用 `helm upgrade trident netapp-trident/trident-operator --version 100.2506.0` 一步即可升级。如果您没有添加 Helm 仓库或无法使用它进行升级：
 - a. 从此处下载最新版Trident"[GitHub 上的 Assets 部分](#)"。
 - b. 使用 `helm upgrade` 其中命令 `trident-operator-25.06.0.tgz` 反映您想要升级到的版本。

```
helm upgrade <name> trident-operator-25.06.0.tgz
```



如果在初始安装期间设置了自定义选项（例如，为Trident和 CSI 映像指定私有镜像注册表），请附加以下内容：`helm upgrade` 命令使用 `--set` 确保将这些选项包含在升级命令中，否则这些值将重置为默认值。

2. 跑步 `helm list` 确认图表和应用程序版本均已升级。跑步 `tridentctl logs` 查看所有调试信息。

从 `tridentctl` 安装到Trident操作员

您可以从以下位置升级到最新版本的Trident操作符：`tridentctl` 安装。现有的后端和PVC将自动可用。



在切换安装方法之前，请先查看"[安装方法之间的转换](#)"。

步骤

1. 下载最新版的Trident。

```
# Download the release required [25.06.0]
mkdir 25.06.0
cd 25.06.0
wget
https://github.com/NetApp/trident/releases/download/v25.06.0/trident-
installer-25.06.0.tar.gz
tar -xf trident-installer-25.06.0.tar.gz
cd trident-installer
```

2. 创建 `tridentorchestrator` 来自清单文件的 CRD。

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

3. 在同一命名空间中部署集群范围的操作符。

```
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. 创建一个 `TridentOrchestrator` 安装Trident的 CR。

```
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. 确认Trident已升级到预期版本。

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

```
Message:          Trident installed
Namespace:        trident
Status:           Installed
Version:          v25.06.0
```

使用 **tridentctl** 进行升级

您可以使用以下方法轻松升级现有的Trident安装 **tridentctl**。

关于此任务

卸载并重新安装Trident相当于升级。卸载Trident时，Trident部署使用的持久卷声明 (PVC) 和持久卷 (PV) 不会被删除。在Trident离线期间，已配置的 PV 仍将可用；Trident恢复在线后，将为在此期间创建的任何 PVC 配置卷。

开始之前

审查["选择升级方式"](#)升级前使用 **tridentctl**。

步骤

1. 运行卸载命令 ``tridentctl`` 移除与Trident关联的所有资源，但保留 CRD 和相关对象。

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

2. 重新安装Trident。参考["使用 **tridentctl** 安装Trident"](#)。



请勿中断升级过程。确保安装程序运行完成。

版权信息

版权所有 © 2026 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。