



升级 Trident Trident

NetApp
October 29, 2026

目录

- 升级 Trident 1
 - 升级 Trident 1
 - 升级前的注意事项 1
 - 步骤 1：选择一个版本 1
 - 步骤 2：确定原始安装方法 1
 - 步骤 3：选择一种升级方法 2
 - 使用 operator 进行升级 2
 - 了解 Trident Operator 升级工作流程 2
 - 使用 Trident 操作员或 Helm 升级 Trident 安装 3
 - 使用 tridentctl 升级 Trident 6

升级 Trident

升级 Trident

从 24.02 版本开始，Trident 遵循四个月的发布节奏，每个日历年发布三个主要版本。每个新版本都建立在先前版本的基础上，并提供新功能、性能增强、错误修复和改进。我们建议您每年至少升级一次，以利用 Trident 中的新功能。

升级前的注意事项

升级到最新版 Trident 时，请考虑以下事项：

- 在给定的 Kubernetes 集群中的所有命名空间中应仅安装一个 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。
- 升级时，请务必提供 Trident 使用的 `parameter.fsType`中的`StorageClasses`。您可以删除和重新创建 `StorageClasses` 而不会中断先前存在的卷。
 - 这是强制执行 "安全上下文" SAN 卷的要求。
 - [sample input](#) 目录包含示例，例如 `storage-class-basic.yaml.templ` 和 `storage-class-bronze-default.yaml`。
 - 有关详细信息，请参阅 ["已知问题"](#)。

步骤 1：选择一个版本

Trident 版本遵循基于日期的 `YY.MM`命名约定，其中"YY"是年份的最后两位数字，"MM"是月份。点版本遵循`YY.MM.X`约定，其中"X"是补丁级别。您将根据要升级的版本选择要升级到的版本。`

- 您可以直接升级到安装版本的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 dot 版本）升级到 25.06。
- 如果您要从四发布窗口之外的版本进行升级，请执行多步骤升级。使用您要从["早期版本"](#)升级的升级说明升级到适合四发布窗口的最新版本。例如，如果您运行的是 23.07 并希望升级到 25.06：
 - a. 首次从 23.07 升级到 24.06。
 - b. 然后从 24.06 升级到 25.06。



在 OpenShift Container Platform 上使用 Trident 操作员升级时，应升级到 Trident 21.01.1 或更高版本。随 21.01.0 发布的 Trident 操作员包含一个已在 21.01.1 中修复的已知问题。有关更多详细信息，请参阅 ["在 GitHub 上的问题详细信息"](#)。

步骤 2：确定原始安装方法

要确定您最初用于安装 Trident 的版本：

1. 使用 `kubectl get pods -n trident` 检查 pod。
 - 如果没有操作员舱，则说明 Trident 是使用 `tridentctl` 安装的。

- 如果有 operator pod，Trident 是使用 Trident operator 手动或使用 Helm 安装的。
- 2. 如果有操作员 pod，请使用 `kubectl describe torc` 来确定是否使用 Helm 安装了 Trident。
 - 如果有 Helm 标签，则说明 Trident 是使用 Helm 安装的。
 - 如果没有 Helm 标签，则使用 Trident 操作员手动安装 Trident。

步骤 3：选择一种升级方法

一般来说，您应该使用与初始安装相同的方法进行升级，但您可以["在安装方法之间移动"](#)。升级 Trident 有两种选择。

- ["使用 Trident 操作员进行升级"](#)



我们建议您在使用 operator 升级之前查看["了解 operator 升级工作流程"](#)。

*

使用 operator 进行升级

了解 Trident Operator 升级工作流程

在使用 Trident 操作员升级 Trident 之前，您应该了解升级过程中发生的后台过程。这包括对 Trident 控制器、控制器 Pod 和节点 Pod 以及启用滚动更新的节点 DaemonSet 的更改。

Trident 操作员升级处理

安装和升级 Trident 的众多["使用 Trident 运算符的好处"](#)之一是自动处理 Trident 和 Kubernetes 对象，而不会中断现有的已挂载卷。通过这种方式，Trident 可以支持零停机时间升级，或["滚动更新"](#)。特别是，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。
 - 如果节点未更新，则不会阻止更新剩余节点。
 - 只有运行 Trident Node Pod 的节点才能挂载卷。



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

Operator 升级工作流

使用 Trident 操作员启动升级时：

1. **Trident 操作员：**
 - a. 检测当前安装的 Trident 版本（version *n*）。
 - b. 更新所有 Kubernetes 对象，包括 CRD、RBAC 和 Trident SVC。

- c. 删除版本 n 的 Trident Controller 部署。
- d. 为版本 $n+1$ 创建 Trident Controller 部署。
2. **Kubernetes** 为 $n+1$ 创建 Trident Controller Pod。
3. **Trident** 操作员：
 - a. 删除 n 的 Trident 节点 DaemonSet。操作员不等待节点 Pod 终止。
 - b. 为 $n+1$ 创建 Trident 节点守护进程集。
4. **Kubernetes** 在不运行 Trident Node Pod n 的节点上创建 Trident Node Pod。这可确保一个节点上不会有多于一个任何版本的 Trident Node Pod。

使用 Trident 操作员或 Helm 升级 Trident 安装

您可以使用 Trident 操作员手动或使用 Helm 升级 Trident。您可以从 Trident 操作员安装升级到另一个 Trident 操作员安装，或从 `tridentctl` 安装升级到 Trident 操作员版本。在升级 Trident 操作员安装之前，请查看["选择升级方法"](#)。

升级手动安装

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



要从使用命名空间范围运算符（版本 20.07 至 20.10）安装的 Trident 进行升级，请使用 ["您安装的版本"](#) 版本的 Trident 升级说明。

关于此任务

Trident 提供了一个捆绑文件，可用于安装操作员并为您的 Kubernetes 版本创建关联的对象。

- 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

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

步骤

1. 验证您的 Trident 版本：

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

2. 使用要升级到的版本（例如 25.06）的注册表和图像路径以及正确的密码更新 `operator.yaml`、`tridentorchestrator_cr.yaml` 和 `post_1_25_bundle.yaml`。
3. 删除用于安装当前 Trident 实例的 Trident 操作员。例如，如果要从 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. 如果使用 `TridentOrchestrator` 属性自定义初始安装，则可以编辑 `TridentOrchestrator` 对象以修改安装参数。这可能包括为离线模式指定镜像的 `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. 使用适合您环境的正确捆绑包 YAML 文件安装 `Trident`，其中 `<bundle.yaml/>` 是 `bundle_pre_1_25.yaml` 或 `bundle_post_1_25.yaml`，具体取决于您的 Kubernetes 版本。例如，如果要安装 `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 `Trident torc` 以包括图像 `25.06.0`。

升级 Helm 安装

您可以升级 `Trident Helm` 安装。



在将安装了 `Trident` 的 Kubernetes 集群从 1.24 升级到 1.25 或更高版本时，必须更新 `values.yaml` 以将 `'excludePodSecurityPolicy'` 设置为 `'true'` 或将 `'--set excludePodSecurityPolicy=true'` 添加到 `'helm upgrade'` 命令中，然后才能升级集群。

如果您已经将 Kubernetes 集群从 1.24 升级到 1.25，而没有升级 `Trident helm`，则 `helm` 升级将失败。要完成 `helm` 升级，请执行以下步骤作为先决条件：

1. 从 <https://github.com/helm/helm-mapkubeapis> 安装 `helm-mapkubeapis` 插件。
2. 在安装 `Trident` 的命名空间中对 `Trident` 版本执行试运行。这列出了将要清除的资源。

```
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. 如果您"使用 [Helm 安装了 Trident](#)"，您可以使用 `'helm upgrade trident netapp-trident/trident-operator --version 100.2606.0'` 一步升级。如果您没有添加 `Helm` 存储库或无法使用它进行升级：
 - a. 从 "[GitHub 上的 Assets 部分](#)" 下载最新的 `Trident` 版本。
 - b. 请使用 `helm upgrade` 命令，其中 `trident-operator-26.06.0.tgz` 反映要升级到的版本。

```
helm upgrade <name> trident-operator-26.06.0.tgz
```



如果您在初始安装期间设置了自定义选项（例如为 `Trident` 和 `CSI` 镜像指定私有、镜像仓库），请使用 `helm upgrade` 命令并附加 `'--set'`，以确保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. 运行 `helm list` 以确认图表和应用版本都已升级。运行 `tridentctl logs` 以查看所有调试消息。

从 `tridentctl` 安装升级到 **Trident** 操作员

您可以从 `tridentctl` 安装升级到最新版本的 Trident 操作员。现有的后端和 PVC 将自动可用。



在切换安装方法之前，请查看 ["在安装方法之间切换"](#)。

步骤

1. 下载最新的 Trident 版本。

```
# Download the release required [26.06.0]
mkdir 26.06.0
cd 26.06.0
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
```

2. 从清单中创建 `tridentorchestrator` 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作用域运算符。

```
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 TridentOrchestrator CR 以安装 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. 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:          v26.06.0
```

使用 tridentctl 升级 Trident

您可以使用 `tridentctl` 轻松升级现有 Trident 安装。

关于此任务

卸载和重新安装 Trident 相当于升级。卸载 Trident 时，不会删除 Trident 部署使用的 Persistent Volume Claim (PVC) 和 Persistent Volume (PV)。已配置的 PV 将在 Trident 脱机时保持可用，Trident 将在恢复联机后为在此期间创建的任何 PVC 配置卷。

开始之前

在使用 `tridentctl` 进行升级之前，请先["选择升级方法"](#)审核。

步骤

1. 在 `tridentctl` 中运行卸载命令，以删除与 Trident 关联的所有资源，但 CRD 和相关对象除外。

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

2. 重新安装 Trident。请参阅 ["使用 tridentctl 安装 Trident"](#)。



不要中断升级过程。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。