# 准备工作节点

## Trident

NetApp
September 26, 2025

This PDF was generated from https://docs.netapp.com/zh-cn/trident-2410/trident-use/fcp.html on September 26, 2025. Always check docs.netapp.com for the latest.

# 目录


准备工作节点 1
选择合适的工具 1
节点服务发现 1
NFS卷 2
iSCSI 卷 2
iSCSI自我修复功能 2
安装iSCSI工具 3
配置或禁用iSCSI自我修复 5
NVMe/TCP卷 6
验证安装 7
安装FC工具 7
光纤通道(FC)支持 9
前提条件 9
创建后端配置 12
创建存储类。 12

# 准备工作节点

Kubernetes集群中的所有工作节点都必须能够挂载为Pod配置的卷。要准备工作节点、必须根据您选择的驱动程序安装NFS、iSCSI、NVMe/TCP或FC工具。

## 选择合适的工具

如果您使用的是驱动程序组合、则应安装驱动程序所需的所有工具。默认情况下、最新版本的RedHat CoreTM OS已安装这些工具。

**NFS工具**

"安装NFS工具"如果您使用的是： `ontap-nas`、 `ontap-nas-economy`、 `ontap-nas-flexgroup`、 `azure-netapp-files` `gcp-cvs`。

**iSCSI工具**

"安装iSCSI工具"如果您使用的是： `ontap-san`、 `ontap-san-economy`、 `solidfire-san`。

**NVMe工具**

"安装NVMe工具"用于 `ontap-san`基于TCP (NVMe/TCP)协议的非易失性内存标准(NVMe)。

> 对于NVMe/TCP、建议使用ONTAP 9.12或更高版本。

基于**FC**的**SCSI**工具

基于光纤通道的**SCSI (FC)**是**Trident 24.10**版本中的一项技术预览功能。

"安装FC工具"如果您使用的是 `ontap-san`sanType `fcp` (基于FC的SCSI)。

有关详细信息、请参见 "配置FC和FC-NVMe SAN主机的方式()" 。

## 节点服务发现

Trident会尝试自动检测节点是否可以运行iSCSI或NFS服务。

> 节点服务发现可识别已发现的服务、但无法保证服务已正确配置。相反、如果没有发现的服务、则无法保证卷挂载将失败。

查看事件

Trident会为此节点创建事件以确定发现的服务。要查看这些事件、请运行：

```
kubectl get event -A --field-selector involvedObject.name=<Kubernetes node name>
```

查看发现的服务

Trident标识为Trident节点CR上的每个节点启用的服务。要查看发现的服务、请运行：

```
tridentctl get node -o wide -n <Trident namespace>
```

# NFS卷

使用适用于您的操作系统的命令安装NFS工具。确保NFS服务已在启动期间启动。

**RHEL 8+**

```
sudo yum install -y nfs-utils
```

**Ubuntu**

```
sudo apt-get install -y nfs-common
```

安装NFS工具后重新启动工作节点、以防止在将卷连接到容器时失败。

# iSCSI 卷

Trident可以自动建立iSCSI会话、扫描LUN、发现多路径设备、对其进行格式化并将其挂载到Pod。

## iSCSI自我修复功能

对于ONTAP系统、Trident每五分钟运行一次iSCSI自我修复、以便：

1. *确定*所需的iSCSI会话状态和当前的iSCSI会话状态。
2. 将所需状态与当前状态进行比较、以确定所需的修复。Trident确定修复优先级以及何时抢占修复。
3. 需要执行*修复*才能将当前iSCSI会话状态恢复为所需的iSCSI会话状态。

自我修复活动日志位于相应的Demonset Pod上的容器中 `trident-main`。要查看日志、必须在Trident安装期间将设置 `debug`为"TRUE"。

Trident iSCSI自我修复功能有助于防止：

- 在网络连接问题描述 之后可能发生的陈旧或运行不正常的iSCSI会话。如果会话陈旧、Trident将等待七分钟后再注销、以便重新建立与门户的连接。

  例如、如果在存储控制器上轮换了CHAP密钥、而网络断开了连接、则旧的(*stal*) CHAP密钥可能会持续存在。自修复功能可以识别此问题、并自动重新建立会话以应用更新后的CHAP密码。

- 缺少iSCSI会话
- 缺少LUN

升级**Trident**前的注意事项

- 如果仅使用了每个节点的igroup (在23.04及更高版本中推出)、则iSCSI自我修复功能将对SCSI总线中的所有设备启动SCSI重新检查。
- 如果仅使用后端范围的igroup (自23.04起已弃用)、则iSCSI自行恢复功能将启动SCSI重新检查、以确定SCSI总线中的确切LUN ID。
- 如果混合使用了每个节点的igroub和后端范围的igroux、则iSCSI自我修复功能将对SCSI总线中的确切LUN ID启动SCSI重新检查。

## 安装iSCSI工具

使用适用于您的操作系统的命令安装iSCSI工具。

开始之前

- Kubernetes 集群中的每个节点都必须具有唯一的 IQN 。* 这是必要的前提条件 * 。
- 如果在驱动程序和Element OS 12.5或更早版本中使用RHCOS 4.5或更高版本或其他与RHEL兼容的Linux分发版 `solidfire-san`、请确保在中将CHAP身份验证算法设置为MD5 `/etc/iscsi/iscsid.conf` 。Element 12.7可使用安全FIPS兼容CHAP算法SHA1、SHA-256和SHA3-256。

  ```
  sudo sed -i 's/^\(node.session.auth.chap_algs\).*/\1 = MD5/'
  /etc/iscsi/iscsid.conf
  ```

- 如果在iSCSI PV中使用运行RHE/RedHat CoreTM OS的工作节点、请在StorageClass中指定`discard`mountOption以执行实时空间回收。请参阅 "Red Hat 文档"。

**RHEL 8+**

1. 安装以下系统软件包：

```
sudo yum install -y lsscsi iscsi-initiator-utils device-mapper-multipath
```

2. 检查 iscsi-initiator-utils 版本是否为 6.2.0.877-2.el7 或更高版本：

```
rpm -q iscsi-initiator-utils
```

3. 启用多路径：

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```

> 确保 `etc/multipath.conf`包含 `find_multipaths no`在下 `defaults。

4. 确保 `iscsid`和 `multipathd`正在运行：

```
sudo systemctl enable --now iscsid multipathd
```

5. 启用并启动 `iscsi`：

```
sudo systemctl enable --now iscsi
```

**Ubuntu**

1. 安装以下系统软件包：

```
sudo apt-get install -y open-iscsi lsscsi sg3-utils multipath-tools scsitools
```

2. 检查 open-iscsi 版本是否为 2.0.877-5ubuntu2.10 或更高版本（对于双子系统）或 2.0.877-7.1ubuntu6.1 或更高版本（对于 Focal ）：

```
dpkg -l open-iscsi
```

3. 将扫描设置为手动：

```
sudo sed -i 's/^\(node.session.scan\).*/\1 = manual/'
/etc/iscsi/iscsid.conf
```

4. 启用多路径：

```
sudo tee /etc/multipath.conf <<-EOF
defaults {
    user_friendly_names yes
    find_multipaths no
}
EOF
sudo systemctl enable --now multipath-tools.service
sudo service multipath-tools restart
```

> 确保 `etc/multipath.conf`包含 `find_multipaths no`在下 `defaults`。

5. 确保 `open-iscsi`和 `multipath-tools`已启用且正在运行：

```
sudo systemctl status multipath-tools
sudo systemctl enable --now open-iscsi.service
sudo systemctl status open-iscsi
```

> 对于Ubuntu 18.04、您必须先使用发现目标端口 `iscsiadm`、然后 `open-iscsi`才能启动`iSCSI`守护进程。您也可以将此服务修改 `iscsi`为自动启动 `iscsid`。

## 配置或禁用iSCSI自我修复

您可以配置以下Trident iSCSI自我修复设置来修复陈旧会话：

- **iSCSI**自我修复间隔：确定调用iSCSI自我修复的频率(默认值：5分钟)。您可以将其配置为通过设置较小的数字来提高运行频率、也可以通过设置较大的数字来降低运行频率。

> 将iSCSI自我修复间隔设置为0可完全停止iSCSI自我修复。建议不要禁用iSCSI自我修复；只有在iSCSI自我修复功能无法正常工作或出于调试目的时、才应禁用它。

- **iSCSI**自我修复等待时间：确定在注销运行状况不正常的会话并尝试重新登录之前iSCSI自我修复等待的时间(默认值：7分钟)。您可以将其配置为较大的数字、以便确定为运行状况不正常的会话必须等待较长的时间才能注销、然后再尝试重新登录、或者配置为较小的数字以较早地注销和登录。

掌舵

要配置或更改iSCSI自我修复设置、请在Helm安装或Helm更新期间传递 `iscsiSelfHealingInterval`和 `iscsiSelfHealingWaitTime`参数。

以下示例将iSCSI自我修复间隔设置为3分钟、并将自我修复等待时间设置为6分钟：

```
helm install trident trident-operator-100.2410.0.tgz --set
iscsiSelfHealingInterval=3m0s --set iscsiSelfHealingWaitTime=6m0s -n
trident
```

**Tridentctl**

要配置或更改iSCSI自我修复设置、请在安装或更新dentctri期间传递 `iscsi-self-healing-interval`和 `iscsi-self-healing-wait-time`参数。

以下示例将iSCSI自我修复间隔设置为3分钟、并将自我修复等待时间设置为6分钟：

```
tridentctl install --iscsi-self-healing-interval=3m0s --iscsi-self
-healing-wait-time=6m0s -n trident
```

# NVMe/TCP卷

使用适用于您的操作系统的命令安装NVMe工具。

- NVMe需要RHEL 9或更高版本。
- 如果Kubelnetes节点的内核版本太旧、或者NVMe软件包不适用于您的内核版本、您可能需要将节点的内核版本更新为具有NVMe软件包的版本。

**RHEL 9**

```
sudo yum install nvme-cli
sudo yum install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

**Ubuntu**

```
sudo apt install nvme-cli
sudo apt -y install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

### 验证安装

安装后、使用命令验证Kubenetes集群中的每个节点是否都具有唯一的NQN：

```
cat /etc/nvme/hostnqn
```

> Trident会修改此 `ctrl_device_tmo`值、以确保NVMe在路径发生故障时不会放弃此路径。请勿更改此设置。

## 安装FC工具

使用适用于您的操作系统的命令安装FC工具。

- 如果将运行RHE/RedHat Core-OS的工作节点与FC PV结合使用、请在StorageClass中指定 `discard`mountOption以执行实时空间回收。请参阅 "Red Hat 文档"。

**RHEL 8+**

1. 安装以下系统软件包：

```
sudo yum install -y lsscsi device-mapper-multipath
```

2. 启用多路径：

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```

> 确保 `etc/multipath.conf`包含 `find_multipaths no`在下 `defaults`。

3. 确保 `multipathd`正在运行：

```
sudo systemctl enable --now multipathd
```

**Ubuntu**

1. 安装以下系统软件包：

```
sudo apt-get install -y lsscsi sg3-utils multipath-tools scsitools
```

2. 启用多路径：

```
sudo tee /etc/multipath.conf <<-EOF
defaults {
    user_friendly_names yes
    find_multipaths no
}
EOF
sudo systemctl enable --now multipath-tools.service
sudo service multipath-tools restart
```

> 确保 `etc/multipath.conf`包含 `find_multipaths no`在下 `defaults`。

3. 确保 `multipath-tools`已启用且正在运行：

```
sudo systemctl status multipath-tools
```

# 光纤通道(FC)支持

## 现在、您可以在Trident中使用光纤通道(Fibre Channel、FC)协议来配置和管理ONTAP系统上的存储资源。

基于光纤通道的**SCSI (FC)**是**Trident 24.10**版本中的一项技术预览功能。

光纤通道因其高性能、可靠性和可扩展性而成为企业存储环境中广泛采用的协议。它为存储设备提供了一个强大而高效的通信通道、可实现快速而安全的数据传输。通过使用基于光纤通道的SCSI、您可以利用现有基于SCSI的存储基础架构、同时享受光纤通道的高性能和远距离功能。它可以整合存储资源、创建可扩展的高效存储区域网络(SAN)、从而以低延迟处理大量数据。

将FC功能与Trident结合使用、您可以执行以下操作：

- 使用部署规范动态配置PVC。
- 创建卷快照并从此快照创建新卷。
- 克隆现有FC-PVC。
- 调整已部署卷的大小。

## 前提条件

为FC配置所需的网络和节点设置。

### 网络设置

1. 获取目标接口的WWPN。有关详细信息、请参见 "network interface show" 。
2. 获取启动程序(主机)上接口的WWPN。

   请参阅相应的主机操作系统实用程序。

3. 使用主机和目标的WWPN在FC交换机上配置分区。

   有关信息、请参见相应的交换机供应商文档。

   有关详细信息、请参见以下ONTAP文档：

   - "光纤通道和 FCoE 分区概述"
   - "配置FC和FC-NVMe SAN主机的方式()"

### 准备工作节点

Kubernetes集群中的所有工作节点都必须能够挂载为Pod配置的卷。要为FC准备工作节点、必须安装所需的工具。

#### 安装**FC**工具

使用适用于您的操作系统的命令安装FC工具。

- 如果将运行RHE/RedHat Core-OS的工作节点与FC PV结合使用、请在StorageClass中指定

`discard`mountOption以执行实时空间回收。请参阅 "Red Hat 文档"。

**RHEL 8+**

1. 安装以下系统软件包：

```
sudo yum install -y lsscsi device-mapper-multipath
```

2. 启用多路径：

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```

> 确保 `etc/multipath.conf`包含 `find_multipaths no`在下 `defaults`。

3. 确保 `multipathd`正在运行：

```
sudo systemctl enable --now multipathd
```

**Ubuntu**

1. 安装以下系统软件包：

```
sudo apt-get install -y lsscsi sg3-utils multipath-tools scsitools
```

2. 启用多路径：

```
sudo tee /etc/multipath.conf <<-EOF
defaults {
    user_friendly_names yes
    find_multipaths no
}
EOF
sudo systemctl enable --now multipath-tools.service
sudo service multipath-tools restart
```

> 确保 `etc/multipath.conf`包含 `find_multipaths no`在下 `defaults`。

3. 确保 `multipath-tools`已启用且正在运行：

```
sudo systemctl status multipath-tools
```

## 创建后端配置

为驱动程序和 `fcp`创建一个Trident后端 `ontap-san`作为sanType。

请参阅：

- "准备使用ONTAP SAN驱动程序配置后端"
- "ONTAP SAN配置选项和示例"

使用**FC**的后端配置示例

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  sanType: fcp
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

## 创建存储类。

有关详细信息、请参见：

- "存储配置选项"

存储类示例

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: fcp-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  storagePools: "ontap-san-backend:.*"
  fsType: "ext4"
allowVolumeExpansion: True
```

版权信息

版权所有 © 2025 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。