



Trident for Docker

Trident

NetApp
January 15, 2026

目录

Trident for Docker	1
部署先决条件	1
核实要求	1
NVMe 工具	3
FC工具	4
部署Trident	6
Docker 管理的插件方法（版本 1.13/17.03 及更高版本）	6
传统方法（版本 1.12 或更早版本）	8
系统启动时启动Trident	9
升级或卸载Trident	10
升级	10
卸载	12
使用卷	12
创建卷	12
移除音量	13
克隆卷	13
访问外部创建的卷	14
驱动程序特定音量选项	14
收集日志	19
收集日志以进行故障排除	19
一般故障排除技巧	19
管理多个Trident实例	20
Docker 管理插件（版本 1.13/17.03 或更高版本）的步骤	20
传统方法（版本 1.12 或更早版本）的步骤	20
存储配置选项	21
全局配置选项	21
ONTAP 配置	22
Element软件配置	29
已知问题和限制	31
将Trident Docker Volume Plugin 从旧版本升级到 20.10 及更高版本会导致升级失败，并出现“没有这样的文件或目录”错误。	31
卷名长度至少为 2 个字符。	32
Docker Swarm 的某些行为导致Trident无法支持它与所有存储和驱动程序的组合。	32
如果正在配置FlexGroup，而第二个FlexGroup与正在配置的FlexGroup有一个或多个共同的聚合，则ONTAP不会配置第二个FlexGroup。	32

Trident for Docker

部署先决条件

在部署Trident之前，您必须在主机上安装和配置必要的协议先决条件。

核实要求

- 确认您的部署满足所有要求["要求"](#)。
- 请确认您已安装受支持的 Docker 版本。如果您的 Docker 版本已过时， ["安装或更新它"](#) 。

```
docker --version
```

- 请确认您的主机上已安装并配置协议所需的先决条件。

NFS 工具

使用适用于您操作系统的命令安装 NFS 工具。

RHEL 8+

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

Ubuntu

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



安装 NFS 工具后重启工作节点，以防止将卷附加到容器时发生故障。

iSCSI 工具

使用适用于您操作系统的命令安装 iSCSI 工具。

RHEL 8+

1. 安装以下系统软件包：

```
sudo yum install -y lsscsi iscsi-initiator-utils sg3_utils device-  
mapper-multipath
```

2. 请检查 iscsi-initiator-utils 版本是否为 6.2.0.874-2.el7 或更高版本：

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

3. 将扫描方式设置为手动：

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

4. 启用多路径：

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



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

5. 确保 `iscsid` 和 `multipathd` 正在运行：

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

6. 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. 请检查 open-iscsi 版本是否为 2.0.874-5ubuntu2.10 或更高版本（适用于 bionic）或 2.0.874-7.1ubuntu6.1 或更高版本（适用于 focal）：

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

3. 将扫描方式设置为手动：

```
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` 在下面 `defaults`。

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

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

NVMe 工具

使用适用于您操作系统的命令安装 NVMe 工具。



- NVMe 需要 RHEL 9 或更高版本。
- 如果您的 Kubernetes 节点的内核版本太旧，或者您的内核版本没有 NVMe 软件包，则您可能需要将节点的内核版本更新为包含 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
```

FC工具

使用适用于您操作系统的命令安装 FC 工具。

- 当使用运行 RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) 且带有 FC PV 的工作节点时，请指定 `discard` StorageClass 中的 mountOption 用于执行内联空间回收。参考 ["红帽文档"](#)。

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` 在下面 `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` 在下面 `defaults`。

3. 确保 `multipath-tools` 已启用并正在运行：

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

部署Trident

Trident for Docker 为NetApp存储平台提供与 Docker 生态系统的直接集成。它支持从存储平台到 Docker 主机的存储资源的配置和管理，并提供了一个框架，以便将来添加其他平台。

同一主机上可以同时运行多个Trident实例。这样就可以同时连接到多个存储系统和存储类型，并且可以自定义 Docker 卷使用的存储。

你需要什么

查看["部署的先决条件"](#)。确保满足先决条件后，即可部署Trident。

Docker 管理的插件方法（版本 1.13/17.03 及更高版本）



开始之前

如果您之前在 Docker 1.13/17.03 之前的版本中使用过Trident 的传统守护进程方法，请确保在使用托管插件方法之前停止Trident进程并重新启动 Docker 守护进程。

1. 停止所有正在运行的实例：

```
pkill /usr/local/bin/netappdvp
pkill /usr/local/bin/trident
```

2. 重启 Docker。

```
systemctl restart docker
```

3. 请确保您已安装 Docker Engine 17.03（新版本 1.13）或更高版本。

```
docker --version
```

如果您的版本已过时，["安装或更新您的安装程序"](#)。

步骤

1. 创建配置文件并按如下方式指定选项：

- `config`默认文件名是 `config.json`不过，您可以通过指定名称来使用您选择的任何名称。`config`文件名选项。配置文件必须位于 `/etc/netappdvp`主机系统上的目录。
- log-level：指定日志级别(debug, info, warn, error, fatal)。默认值为 info。
- debug：指定是否启用调试日志记录。默认值为 false。如果为真，则覆盖日志级别。

- i. 创建配置文件存放位置：


```
sudo mkdir -p /etc/netappdvp
```

ii. 创建配置文件:

```
cat << EOF > /etc/netappdvp/config.json
```

```
{  
  "version": 1,  
  "storageDriverName": "ontap-nas",  
  "managementLIF": "10.0.0.1",  
  "dataLIF": "10.0.0.2",  
  "svm": "svm_nfs",  
  "username": "vsadmin",  
  "password": "password",  
  "aggregate": "aggr1"  
}  
EOF
```

2. 使用托管插件系统启动Trident。代替`<version>`使用您正在使用的插件版本（xxx.xx.x）。

```
docker plugin install --grant-all-permissions --alias netapp  
netapp/trident-plugin:<version> config=myConfigFile.json
```

3. 开始使用Trident从已配置的系统获取存储空间。

a. 创建一个名为“firstVolume”的卷:

```
docker volume create -d netapp --name firstVolume
```

b. 容器启动时创建默认卷:

```
docker run --rm -it --volume-driver netapp --volume  
secondVolume:/my_vol alpine ash
```

c. 移除音量“firstVolume”:

```
docker volume rm firstVolume
```

传统方法（版本 1.12 或更早版本）

开始之前

1. 请确保您使用的是 Docker 版本 1.10 或更高版本。

```
docker --version
```

如果您的版本过旧，请更新您的安装。

```
curl -fsSL https://get.docker.com/ | sh
```

或者，["请按照您的分发说明进行操作"](#)。

2. 请确保您的系统已配置 NFS 和/或 iSCSI。

步骤

1. 安装并配置 NetApp Docker 卷插件：

- a. 下载并解压应用程序：

```
wget  
https://github.com/NetApp/trident/releases/download/v25.06.0/trident-  
installer-25.06.0.tar.gz  
tar xzf trident-installer-25.06.0.tar.gz
```

- b. 移动到回收站路径中的某个位置：

```
sudo mv trident-installer/extras/bin/trident /usr/local/bin/  
sudo chown root:root /usr/local/bin/trident  
sudo chmod 755 /usr/local/bin/trident
```

- c. 创建配置文件存放位置：

```
sudo mkdir -p /etc/netappdvp
```

- d. 创建配置文件：

```
cat << EOF > /etc/netappdvp/ontap-nas.json
```

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1"
}
EOF
```

2. 放置好二进制文件并创建配置文件后，使用所需的配置文件启动Trident守护进程。

```
sudo trident --config=/etc/netappdvp/ontap-nas.json
```



除非另有指定，卷驱动程序的默认名称为“netapp”。

守护进程启动后，您可以使用 Docker CLI 界面创建和管理卷。

3. 创建卷：

```
docker volume create -d netapp --name trident_1
```

4. 启动容器时配置 Docker 卷：

```
docker run --rm -it --volume-driver netapp --volume trident_2:/my_vol
alpine ash
```

5. 删除 Docker 卷：

```
docker volume rm trident_1
```

```
docker volume rm trident_2
```

系统启动时启动Trident

systemd 系统的示例单元文件可在以下位置找到：`contrib/trident.service.example`在 Git 仓库中。要在 RHEL 系统中使用该文件，请执行以下操作：

1. 将文件复制到正确位置。

如果运行多个实例，则应为单元文件使用唯一的名称。

```
cp contrib/trident.service.example
/usr/lib/systemd/system/trident.service
```

2. 编辑该文件，将描述（第 2 行）更改为与驱动程序名称匹配，并将配置文件路径（第 9 行）更改为反映您的环境。
3. 重新加载 systemd 以使其吸收更改：

```
systemctl daemon-reload
```

4. 启用该服务。

这个名称会根据你给文件命名的方式而有所不同。`/usr/lib/systemd/system` 目录。

```
systemctl enable trident
```

5. 启动服务。

```
systemctl start trident
```

6. 查看状态。

```
systemctl status trident
```



每次修改单元文件后，请运行以下命令：`systemctl daemon-reload` 命令使其能够感知这些变化。

升级或卸载Trident

您可以安全地升级Trident for Docker，而不会对正在使用的卷产生任何影响。升级过程中会有一段短暂的时间，`docker volume` 针对该插件的命令将不会成功，应用程序将无法挂载卷，直到该插件再次运行。大多数情况下，这只需要几秒钟。

升级

按照以下步骤升级Trident for Docker。

步骤

1. 列出现有卷册：

```
docker volume ls
DRIVER          VOLUME NAME
netapp:latest   my_volume
```

2. 禁用插件：

```
docker plugin disable -f netapp:latest
docker plugin ls
ID                NAME                DESCRIPTION
ENABLED
7067f39a5df5     netapp:latest       nDVP - NetApp Docker Volume
Plugin    false
```

3. 升级插件：

```
docker plugin upgrade --skip-remote-check --grant-all-permissions
netapp:latest netapp/trident-plugin:21.07
```



Trident 18.01 版本取代了 nDVP。你应该直接从以下位置升级：`netapp/ndvp-plugin` 图片到 `netapp/trident-plugin` 图像。

4. 启用插件：

```
docker plugin enable netapp:latest
```

5. 请确认插件已启用：

```
docker plugin ls
ID                NAME                DESCRIPTION
ENABLED
7067f39a5df5     netapp:latest       Trident - NetApp Docker Volume
Plugin    true
```

6. 确认卷可见：

```
docker volume ls
DRIVER          VOLUME NAME
netapp:latest   my_volume
```



如果您从旧版本的Trident（20.10 之前）升级到Trident 20.10 或更高版本，则可能会遇到错误。更多信息，请参阅["已知问题"](#)。如果遇到此错误，您应该先禁用该插件，然后删除该插件，最后通过传递额外的配置参数来安装所需的Trident版本：`docker plugin install netapp/trident-plugin:20.10 --alias netapp --grant-all-permissions config=config.json`

卸载

按照以下步骤卸载Trident for Docker。

步骤

1. 删除插件创建的所有卷。
2. 禁用插件：

```
docker plugin disable netapp:latest
docker plugin ls
```

ID	NAME	DESCRIPTION
ENABLED		
7067f39a5df5	netapp:latest	nDVP - NetApp Docker Volume
Plugin	false	

3. 移除插件：

```
docker plugin rm netapp:latest
```

使用卷

您可以使用标准方法轻松创建、克隆和删除卷。`docker volume` 需要时，请使用指定Trident驱动程序名称的命令。

创建卷

- 使用默认名称创建带有驱动程序的卷：

```
docker volume create -d netapp --name firstVolume
```

- 创建包含特定Trident实例的卷：

```
docker volume create -d ntap_bronze --name bronzeVolume
```



如果您没有指定任何内容"options"使用的是驱动程序的默认设置。

- 覆盖默认卷大小。请参阅以下示例，使用驱动程序创建 20 GiB 的卷：

```
docker volume create -d netapp --name my_vol --opt size=20G
```



卷大小以字符串形式表示，其中包含一个整数值，单位可选（例如：10G、20GB、3TiB）。如果没有指定单位，则默认值为 G。大小单位可以表示为 2 的幂（B、KiB、MiB、GiB、TiB）或 10 的幂（B、KB、MB、GB、TB）。简写单位使用 2 的幂（G = GiB, T = TiB, ...）。

移除音量

- 删除该卷的方式与其他 Docker 卷一样：

```
docker volume rm firstVolume
```



使用时 `solidfire-san` 上述示例驱动程序删除并清除卷。

按照以下步骤升级Trident for Docker。

克隆卷

使用时 `ontap-nas`，`ontap-san`，`solidfire-san`，和 ``gcp-cvs storage drivers`Trident` 可以克隆卷。使用时 ``ontap-nas-flexgroup`` 或者 ``ontap-nas-economy`` 驱动程序不支持克隆。从现有卷创建新卷将创建一个新的快照。

- 检查卷以枚举快照：

```
docker volume inspect <volume_name>
```

- 从现有卷创建新卷。这将导致创建一个新的快照：

```
docker volume create -d <driver_name> --name <new_name> -o from  
=<source_docker_volume>
```

- 从现有卷上的快照创建新卷。这不会创建新的快照：

```
docker volume create -d <driver_name> --name <new_name> -o from  
=<source_docker_volume> -o fromSnapshot=<source_snap_name>
```

示例

```
docker volume inspect firstVolume

[
  {
    "Driver": "ontap-nas",
    "Labels": null,
    "Mountpoint": "/var/lib/docker-volumes/ontap-nas/netappdvp_firstVolume",
    "Name": "firstVolume",
    "Options": {},
    "Scope": "global",
    "Status": {
      "Snapshots": [
        {
          "Created": "2017-02-10T19:05:00Z",
          "Name": "hourly.2017-02-10_1505"
        }
      ]
    }
  }
]

docker volume create -d ontap-nas --name clonedVolume -o from=firstVolume clonedVolume

docker volume rm clonedVolume
docker volume create -d ontap-nas --name volFromSnap -o from=firstVolume -o fromSnapshot=hourly.2017-02-10_1505 volFromSnap

docker volume rm volFromSnap
```

访问外部创建的卷

只有当外部创建的块设备（或其克隆）没有分区且其文件系统受Trident支持时（例如：一个），容器才能使用Trident访问这些块设备（或其克隆）。`ext4` 格式化 `/dev/sdc1` 无法通过Trident访问）。

驱动程序特定音量选项

每个存储驱动程序都有一组不同的选项，您可以在创建卷时指定这些选项以自定义结果。请参阅下方适用于您配置的存储系统的选项。

在创建卷操作期间使用这些选项非常简单。提供选项和值 `o` CLI 操作期间的运算符。这些值会覆盖 JSON 配置文件中的任何等效值。

ONTAP容量选项

NFS、iSCSI 和 FC 的卷创建选项包括以下几种：

选项	描述
size	卷大小默认为 1 GiB。
spaceReserve	容量配置方式可以是精简配置或厚配置，默认为精简配置。有效值为 none（精简配置）和 volume（厚配置）。
snapshotPolicy	这将把快照策略设置为所需值。默认值为 `none` 这意味着不会自动为该卷创建快照。除非存储管理员修改，否则所有ONTAP系统上都存在一个名为“default”的策略，该策略会创建并保留 6 个每小时快照、2 个每日快照和 2 个每周快照。可以通过浏览来恢复快照中保存的数据。`.snapshot` 卷中任意目录中的目录。
snapshotReserve	这将把快照预留量设置为所需的百分比。默认值为空，这意味着如果您选择了 snapshotPolicy，ONTAP 将选择 snapshotReserve（通常为 5%）；如果 snapshotPolicy 为 none，则 ONTAP 将选择 snapshotReserve（通常为 0%）。您可以在配置文件中为所有ONTAP后端设置默认 snapshotReserve 值，并且可以将其用作除 ontap-nas-economy 之外的所有ONTAP后端的卷创建选项。
splitOnClone	克隆卷时，这将导致ONTAP立即将克隆卷与其父卷分离。默认值为 false。有些克隆卷的用例最好在创建后立即将克隆卷与其父卷分离，因为不太可能有机会提高存储效率。例如，克隆一个空数据库可以节省大量时间，但节省的存储空间却很少，因此最好立即拆分克隆。
encryption	在新卷上启用NetApp卷加密 (NVE)；默认设置为 false。要使用此选项，必须在集群上获得 NVE 许可并启用 NVE。 如果后端启用了 NAE，则在Trident中配置的任何卷都将启用 NAE。 更多信息，请参阅："Trident如何与 NVE 和 NAE 协同工作"。
tieringPolicy	设置卷要使用的分层策略。这决定了当数据变为不活动状态（冷数据）时，是否将其迁移到云层。

以下附加选项仅适用于 NFS：

选项	描述
unixPermissions	这控制卷本身的权限集。默认情况下，权限将设置为 `---rwxr-xr-x` 或者用数字表示法表示为 0755，并且 `root` 将成为所有者。文本格式或数字格式均可。
snapshotDir	将此设置为 `true` 将使 `.snapshot` 客户端访问该卷时可见的目录。默认值为 `false` 这意味着可见性 `.snapshot` 目录默认处于禁用状态。某些镜像，例如官方的 MySQL 镜像，在以下情况下无法按预期工作：`.snapshot` 目录可见。
exportPolicy	设置该卷要使用的导出策略。默认值为 default。
securityStyle	设置用于访问卷的安全样式。默认值为 unix。有效值为 unix 和 mixed。

以下附加选项仅适用于 iSCSI：

选项	描述
fileSystemType	设置用于格式化 iSCSI 卷的文件系统。默认值为 ext4。有效值为 ext3，ext4，和 xfs。
spaceAllocation	将其设置为 `false` 将关闭 LUN 的空间分配功能。默认值为 `true` 这意味着当卷空间不足且卷中的 LUN 无法接受写入时，ONTAP 会通知主机。此选项还允许 ONTAP 在主机删除数据时自动回收空间。

示例

请看以下示例：

- 创建一个 10 GiB 卷：

```
docker volume create -d netapp --name demo -o size=10G -o
encryption=true
```

- 创建一个带有快照的 100 GiB 卷：

```
docker volume create -d netapp --name demo -o size=100G -o
snapshotPolicy=default -o snapshotReserve=10
```

- 创建一个启用了 setUID 位的卷：

```
docker volume create -d netapp --name demo -o unixPermissions=4755
```

最小卷大小为 20 MiB。

如果未指定快照保留，且快照策略为 `none` Trident 使用 0% 的快照储备。

- 创建一个没有快照策略和快照保留的卷：

```
docker volume create -d netapp --name my_vol --opt snapshotPolicy=none
```

- 创建一个没有快照策略且自定义快照保留比例为 10% 的卷：

```
docker volume create -d netapp --name my_vol --opt snapshotPolicy=none
--opt snapshotReserve=10
```

- 创建一个具有快照策略和 10% 自定义快照保留空间的卷：

```
docker volume create -d netapp --name my_vol --opt
snapshotPolicy=myPolicy --opt snapshotReserve=10
```

- 创建具有快照策略的卷，并接受 ONTAP 的默认快照保留（通常为 5%）：

```
docker volume create -d netapp --name my_vol --opt
snapshotPolicy=myPolicy
```

Element 软件音量选项

Element 软件选项会显示与卷相关的大小和服务质量 (QoS) 策略。创建卷时，使用以下方式指定与其关联的 QoS 策略：`-o type=service\_level`命名法。

使用 Element 驱动程序定义 QoS 服务级别的第一步是创建至少一个类型，并在配置文件中指定与名称关联的最小、最大和突发 IOPS。

Element软件的其他卷创建选项包括以下几种：

选项	描述
size	卷的大小，默认为 1 GiB 或配置条目...“defaults”：{“size”: “5G”}。
blocksize	可以使用 512 或 4096，默认值为 512 或配置项 DefaultBlockSize。

示例

请参阅以下包含 QoS 定义的示例配置文件：

```
{
  "Types": [
    {
      "Type": "Bronze",
      "Qos": {
        "minIOPS": 1000,
        "maxIOPS": 2000,
        "burstIOPS": 4000
      }
    },
    {
      "Type": "Silver",
      "Qos": {
        "minIOPS": 4000,
        "maxIOPS": 6000,
        "burstIOPS": 8000
      }
    },
    {
      "Type": "Gold",
      "Qos": {
        "minIOPS": 6000,
        "maxIOPS": 8000,
        "burstIOPS": 10000
      }
    }
  ]
}
```

在上述配置中，我们三个策略定义：青铜、白银和黄金。这些名称是随意起的。

- 创建 10 GiB 黄金卷：

```
docker volume create -d solidfire --name sfGold -o type=Gold -o size=10G
```

- 创建 100 GiB 青铜卷：

```
docker volume create -d solidfire --name sfBronze -o type=Bronze -o
size=100G
```

收集日志

您可以收集日志以帮助进行故障排除。收集日志的方法取决于您运行 Docker 插件的方式。

收集日志以进行故障排除

步骤

1. 如果您使用推荐的托管插件方法运行Trident（即，使用`docker plugin`命令），可以这样理解：

```
docker plugin ls
```

ID	NAME	DESCRIPTION
ENABLED		
4fb97d2b956b	netapp:latest	nDVP - NetApp Docker Volume
Plugin	false	
journalctl -u docker grep 4fb97d2b956b		

标准日志级别应该足以诊断大多数问题。如果这还不够，您可以启用调试日志记录。

2. 要启用调试日志记录，请安装启用调试日志记录功能的插件：

```
docker plugin install netapp/trident-plugin:<version> --alias <alias>  
debug=true
```

或者，在插件已安装的情况下启用调试日志记录：

```
docker plugin disable <plugin>
```

```
docker plugin set <plugin> debug=true
```

```
docker plugin enable <plugin>
```

3. 如果您在主机上运行该二进制文件，则日志位于主机的目录中。`/var/log/netappdvp`目录。要启用调试日志记录，请指定`-debug`当您运行插件时。

一般故障排除技巧

- 新用户遇到的最常见问题是配置错误，导致插件无法初始化。发生这种情况时，当您尝试安装或启用插件时，可能会看到类似这样的消息：

```
Error response from daemon: dial unix /run/docker/plugins/<id>/netapp.sock:
connect: no such file or directory
```

这意味着插件启动失败。幸运的是，该插件内置了全面的日志记录功能，应该可以帮助您诊断可能遇到的大多数问题。

- 如果将光伏系统安装到集装箱上遇到问题，请确保：`rpcbind` 已安装并正在运行。使用主机操作系统所需的软件包管理器并检查是否 `rpcbind` 正在运行。您可以通过运行以下命令来检查 rpcbind 服务的状态：
`systemctl status rpcbind` 或其等效物。

管理多个Trident实例

当您需要同时拥有多个存储配置时，需要多个Trident实例。实现多个实例的关键在于使用不同的名称来命名它们。`--alias` 使用容器化插件的选项，或者 `--volume-driver` 在主机上实例化Trident时的选择。

Docker 管理插件（版本 1.13/17.03 或更高版本）的步骤

1. 启动第一个实例时，指定别名和配置文件。

```
docker plugin install --grant-all-permissions --alias silver
netapp/trident-plugin:21.07 config=silver.json
```

2. 启动第二个实例，指定不同的别名和配置文件。

```
docker plugin install --grant-all-permissions --alias gold
netapp/trident-plugin:21.07 config=gold.json
```

3. 创建卷时，将别名指定为驱动程序名称。

例如，黄金交易量：

```
docker volume create -d gold --name ntapGold
```

例如，对于白银交易量而言：

```
docker volume create -d silver --name ntapSilver
```

传统方法（版本 1.12 或更早版本）的步骤

1. 使用自定义驱动程序 ID 通过 NFS 配置启动插件：

```
sudo trident --volume-driver=netapp-nas --config=/path/to/config  
-nfs.json
```

2. 使用自定义驱动程序 ID 和 iSCSI 配置启动插件：

```
sudo trident --volume-driver=netapp-san --config=/path/to/config  
-iscsi.json
```

3. 为每个驱动程序实例配置 Docker 卷：

例如，对于 NFS：

```
docker volume create -d netapp-nas --name my_nfs_vol
```

例如，对于 iSCSI：

```
docker volume create -d netapp-san --name my_iscsi_vol
```

存储配置选项

查看适用于您的Trident配置的配置选项。

全局配置选项

这些配置选项适用于所有Trident配置，无论使用何种存储平台。

选项	描述	示例
version	配置文件版本号	1
storageDriverName	存储驱动程序名称	ontap-nas, ontap-san , ontap-nas-economy , ontap-nas-flexgroup , solidfire-san
storagePrefix	卷名称的可选前缀。默认： netappdvp_。	staging_
limitVolumeSize	对容量大小有可选限制。默认值："" (不强制执行)	10g



请勿使用 `storagePrefix`（包括默认值）适用于 Element 后端。默认情况下，`solidfire-san` 驱动程序将忽略此设置，并且不使用前缀。NetApp 建议使用特定的租户 ID 进行 Docker 卷映射，或者使用属性数据，该属性数据填充了 Docker 版本、驱动程序信息和来自 Docker 的原始名称（如果可能使用了任何名称修改）。

可以使用默认选项，避免在创建的每个卷上都指定这些选项。这 `size` 此选项适用于所有控制器类型。有关如何设置默认卷大小的示例，请参阅 ONTAP 配置部分。

选项	描述	示例
<code>size</code>	新建卷的可选默认大小。默认：1G	10G

ONTAP 配置

除了上述全局配置值之外，使用 ONTAP 时，还可以使用以下顶级选项。

选项	描述	示例
<code>managementLIF</code>	ONTAP 管理 LIF 的 IP 地址。您可以指定一个完全限定域名 (FQDN)。	10.0.0.1
<code>dataLIF</code>	LIF 协议的 IP 地址。 • ONTAP NAS 驱动程序*：NetApp 建议指定 <code>dataLIF</code>。如果未提供，Trident 将从 SVM 获取 <code>dataLIF</code>。您可以指定一个完全限定域名 (FQDN) 用于 NFS 挂载操作，从而创建轮询 DNS 以在多个 <code>dataLIF</code> 之间进行负载均衡。 • ONTAP SAN 驱动程序*：请勿指定 iSCSI 或 FC。Trident 的使用 "ONTAP 选择性 LUN 地图" 发现建立多路径会话所需的 iSCSI 或 FC LIF。如果出现以下情况，则会生成警告：`dataLIF` 已明确定义。	10.0.0.2
<code>svm</code>	要使用的存储虚拟机（如果管理 LIF 是集群 LIF，则此项为必填项）	<code>svm_nfs</code>
<code>username</code>	连接到存储设备的用户名	<code>vsadmin</code>
<code>password</code>	连接存储设备的密码	<code>secret</code>

选项	描述	示例
aggregate	用于配置的聚合（可选；如果设置，则必须分配给 SVM）。对于 `ontap-nas-flexgroup` 驱动程序，此选项将被忽略。分配给 SVM 的所有聚合都用于配置 FlexGroup 卷。	aggr1
limitAggregateUsage	可选，如果使用率超过此百分比，则配置失败	75%
nfsMountOptions	对 NFS 挂载选项进行精细控制；默认值为“-o nfsvers=3”。仅限以下情况：`ontap-nas` 和 `ontap-nas-economy` 司机。 "请在此处查看 NFS 主机配置信息" 。	-o nfsvers=4
igroupName	Trident 创建和管理每个节点 igroups 作为 `netappdvp`。 此值不能更改或省略。 仅限以下情况：`ontap-san` 司机。	netappdvp
limitVolumeSize	最大可请求容量。	300g
qtreesPerFlexvol	每个 FlexVol 的最大 qtree 数量必须在 [50, 300] 范围内，默认值为 200。 对于 `ontap-nas-economy` 驱动程序，此选项允许自定义每个 FlexVol 的最大 qtree 数量。	300
sanType	*支持 `ontap-san` 仅限司机。*用于选择 `iscsi` 对于 iSCSI，`nvme` 适用于 NVMe/TCP 或 `fcp` 用于光纤通道 (FC) 上的 SCSI。	`iscsi` 如果为空
limitVolumePoolSize	*支持 `ontap-san-economy` 和 `ontap-san-economy` 仅限司机。*限制 ONTAP ontap-nas-economy 和 ontap-SAN-economy 驱动程序中的 FlexVol 大小。	300g

系统提供了默认选项，避免在创建的每个卷上都进行指定：

选项	描述	示例
spaceReserve	空间预订模式； none（精简配置）或 volume（厚的）	none
snapshotPolicy	要使用的快照策略，默认值为 none	none
snapshotReserve	快照预留百分比，默认值为空字符串，表示接受ONTAP默认值。	10
splitOnClone	创建时将克隆体与其父级分离，默认为 false	false
encryption	在新卷上启用NetApp卷加密 (NVE)；默认启用 false。要使用此选项，必须在集群上获得 NVE 许可并启用 NVE。 如果后端启用了 NAE，则在Trident中配置的任何卷都将启用 NAE。 更多信息，请参阅："Trident如何与 NVE 和 NAE 协同工作"。	true
unixPermissions	NAS 选项，用于已配置的 NFS 卷，默认值为 777	777
snapshotDir	NAS选项用于访问`.snapshot`目录。	NFSv4 为“true”，NFSv3 为“false”。
exportPolicy	NFS导出策略要使用的NAS选项，默认值为 default	default
securityStyle	NAS 选项，用于访问已配置的 NFS 卷。 NFS 支持 mixed`和`unix`安全措施。默认值为`unix。	unix
fileSystemType	SAN选项用于选择文件系统类型，默认为 ext4	xfs
tieringPolicy	要使用的分层策略，默认值为 none。	none

缩放选项

这`ontap-nas`和`ontap-san`驱动程序为每个 Docker 卷创建一个ONTAP FlexVol。ONTAP每个集群节点最多支持 1000 个 FlexVol，集群最多可支持 12,000 个FlexVol卷。如果您的 Docker 卷需求符合此限制，则`ontap-nas`由于 FlexVols 提供了 Docker 卷粒度快照和克隆等附加功能，因此驱动程序是首选的 NAS 解决方案。

如果您需要的 Docker 卷数量超过了FlexVol 的限制，请选择`ontap-nas-economy`或者`ontap-san-economy`司机。

这 `ontap-nas-economy` 驱动程序在自动管理的 FlexVol 卷池中创建作为 ONTAP Qtree 的 Docker 卷。Qtree 提供了更大的可扩展性，每个集群节点最多可达 100,000 个节点，每个集群最多可达 2,400,000 个节点，但代价是牺牲了一些功能。这 `ontap-nas-economy` 驱动程序不支持 Docker 卷粒度快照或克隆。



这 `ontap-nas-economy` Docker Swarm 目前不支持该驱动程序，因为 Docker Swarm 无法协调跨多个节点的卷创建。

这 `ontap-san-economy` 驱动程序在自动管理的 FlexVol 卷的共享池中创建 Docker 卷作为 ONTAP LUN。这样一来，每个 FlexVol 就不局限于一个 LUN，并且为 SAN 工作负载提供了更好的可扩展性。根据存储阵列的不同，ONTAP 每个集群最多支持 16384 个 LUN。由于这些卷底层是 LUN，因此该驱动程序支持 Docker 卷粒度快照和克隆。

选择 `ontap-nas-flexgroup` 驱动程序可提高单个卷的并行处理能力，该卷可以增长到 PB 级，包含数十亿个文件。FlexGroups 的一些理想用例包括 AI/ML/DL、大数据和分析、软件构建、流媒体、文件存储库等等。Trident 在配置 FlexGroup 卷时会使用分配给 SVM 的所有聚合。Trident 中的 FlexGroup 支持还需考虑以下几点：

- 需要 ONTAP 版本 9.2 或更高版本。
- 截至撰写本文时，FlexGroups 仅支持 NFS v3。
- 建议为 SVM 启用 64 位 NFSv3 标识符。
- 建议的最小 FlexGroup 成员/卷大小为 100 GiB。
- FlexGroup 卷不支持克隆。

有关 FlexGroup 以及适合 FlexGroup 的工作负载的信息，请参阅以下内容：["NetApp FlexGroup 卷最佳实践和实施指南"](#)。

为了在同一环境中获得高级功能和大规模部署，您可以运行多个 Docker Volume Plugin 实例，其中一个使用 `ontap-nas` 以及另一个使用 `ontap-nas-economy`。

为 Trident 定制 ONTAP 角色

您可以创建一个具有最低权限的 ONTAP 集群角色，这样您就不必使用 ONTAP 管理员角色在 Trident 中执行操作。在 Trident 后端配置中包含用户名时，Trident 将使用您创建的 ONTAP 集群角色来执行操作。

请参阅["Trident 自定义角色生成器"](#)有关创建 Trident 自定义角色的更多信息。

使用ONTAP CLI

1. 使用以下命令创建新角色：

```
security login role create <role_name\> -cmddirname "command" -access all  
-vserver <svm_name\>
```

2. 为Trident用户创建用户名：

```
security login create -username <user_name\> -application ontapi  
-authmethod password -role <name_of_role_in_step_1\> -vserver <svm_name\>  
-comment "user_description"  
security login create -username <user_name\> -application http -authmethod  
password -role <name_of_role_in_step_1\> -vserver <svm_name\> -comment  
"user_description"
```

3. 将角色映射到用户：

```
security login modify username <user_name\> -vserver <svm_name\> -role  
<role_name\> -application ontapi -application console -authmethod  
<password\>
```

使用系统管理器

在ONTAP系统管理器中执行以下步骤：

1. 创建自定义角色：

- a. 要在集群级别创建自定义角色，请选择“集群 > 设置”。

（或者）要在 SVM 级别创建自定义角色，请选择“存储”>“存储虚拟机”> required SVM > 设置 > 用户和角色*。

- b. 选择“用户和角色”旁边的箭头图标（→）。
- c. 在“角色”下选择“+添加”。
- d. 定义角色规则，然后点击“保存”。

2. 将角色映射到Trident用户：+ 在“用户和角色”页面上执行以下步骤：

- a. 在“用户”下方选择“添加”图标 +。
- b. 选择所需的用户名，然后在“角色”下拉菜单中选择角色。
- c. 单击“保存”。

更多信息请参阅以下页面：

- ["用于管理ONTAP的自定义角色"或者"定义自定义角色"](#)
- ["与角色和用户协作"](#)

ONTAP配置文件示例

`ontap-nas`驱动程序的NFS示例

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1",
  "defaults": {
    "size": "10G",
    "spaceReserve": "none",
    "exportPolicy": "default"
  }
}
```

`ontap-nas-flexgroup`驱动程序的NFS示例

```
{
  "version": 1,
  "storageDriverName": "ontap-nas-flexgroup",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password",
  "defaults": {
    "size": "100G",
    "spaceReserve": "none",
    "exportPolicy": "default"
  }
}
```

<code>ontap-nas-economy</code> 驱动程序的 NFS 示例

```
{
  "version": 1,
  "storageDriverName": "ontap-nas-economy",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1"
}
```

<code>ontap-san</code> 驱动程序的 iSCSI 示例

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1",
  "igroupName": "netappdvp"
}
```

<code>ontap-san-economy</code> 驱动程序的 NFS 示例

```
{
  "version": 1,
  "storageDriverName": "ontap-san-economy",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi_eco",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1",
  "igroupName": "netappdvp"
}
```

<code>ontap-san</code> 驱动程序的 NVMe/TCP 示例

```
{
  "version": 1,
  "backendName": "NVMeBackend",
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "svm": "svm_nvme",
  "username": "vsadmin",
  "password": "password",
  "sanType": "nvme",
  "useREST": true
}
```

<code>ontap-san</code>驱动程序的SCSI over FC示例

```
{
  "version": 1,
  "backendName": "ontap-san-backend",
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "sanType": "fc",
  "svm": "trident_svm",
  "username": "vsadmin",
  "password": "password",
  "useREST": true
}
```

Element软件配置

除了全局配置值之外，在使用 Element 软件（NetApp HCI/ SolidFire）时，还可以使用这些选项。

选项	描述	示例
Endpoint	https://&lt;登录名&gt;:&lt;密码&gt;:@&lt;mvip&gt;/json-rpc/&lt;元素版本&gt; ;	https://admin:admin@192.168.160.3/json-rpc/8.0
SVIP	iSCSI IP 地址和端口	10.0.0.7:3260

选项	描述	示例
TenantName	要使用的 SolidFireF 租户（如果未找到则创建）	docker
InitiatorIFace	将 iSCSI 流量限制在非默认接口时，请指定接口。	default
Types	QoS规范	参见下面的示例
LegacyNamePrefix	升级版Trident安装的前缀。如果您使用的是 1.3.2 之前的Trident版本，并且使用现有卷执行升级，则需要设置此值才能访问通过卷名称方法映射的旧卷。	netappdvp-

这 `solidfire-san` 驱动程序不支持 Docker Swarm。

示例元素软件配置文件


```

{
  "version": 1,
  "storageDriverName": "solidfire-san",
  "Endpoint": "https://admin:admin@192.168.160.3/json-rpc/8.0",
  "SVIP": "10.0.0.7:3260",
  "TenantName": "docker",
  "InitiatorIFace": "default",
  "Types": [
    {
      "Type": "Bronze",
      "Qos": {
        "minIOPS": 1000,
        "maxIOPS": 2000,
        "burstIOPS": 4000
      }
    },
    {
      "Type": "Silver",
      "Qos": {
        "minIOPS": 4000,
        "maxIOPS": 6000,
        "burstIOPS": 8000
      }
    },
    {
      "Type": "Gold",
      "Qos": {
        "minIOPS": 6000,
        "maxIOPS": 8000,
        "burstIOPS": 10000
      }
    }
  ]
}

```

已知问题和限制

查找有关将Trident与 Docker 结合使用时已知问题和限制的信息。

将**Trident Docker Volume Plugin** 从旧版本升级到 **20.10** 及更高版本会导致升级失败，并出现“没有这样的文件或目录”错误。

临时解决策

1. 禁用插件。

```
docker plugin disable -f netapp:latest
```

2. 移除插件。

```
docker plugin rm -f netapp:latest
```

3. 重新安装插件，并提供额外信息 `config` 范围。

```
docker plugin install netapp/trident-plugin:20.10 --alias netapp --grant  
-all-permissions config=config.json
```

卷名长度至少为 **2** 个字符。



这是 Docker 客户端的限制。客户端会将单个字符名称解释为 Windows 路径。"[参见错误 25773](#)"。

Docker Swarm 的某些行为导致 **Trident** 无法支持它与所有存储和驱动程序的组合。

- Docker Swarm 目前使用卷名称而不是卷 ID 作为其唯一的卷标识符。
- 卷请求会同时发送到 Swarm 集群中的每个节点。
- 卷插件（包括 Trident）必须在 Swarm 集群中的每个节点上独立运行。由于 ONTAP 的工作方式以及 `ontap-nas` 和 `ontap-san` 驾驶员之所以能发挥作用，是因为他们是唯一能够在这些限制条件下操作的人。

其余驱动程序会受到诸如竞争条件之类的问题的影响，这可能会导致单个请求创建大量卷，而没有明确的“赢家”；例如，Element 具有允许卷具有相同名称但 ID 不同的功能。

NetApp 已向 Docker 团队提供了反馈，但目前没有任何迹象表明未来会采取什么措施。

如果正在配置 **FlexGroup**，而第二个 **FlexGroup** 与正在配置的 **FlexGroup** 有一个或多个共同的聚合，则 **ONTAP** 不会配置第二个 **FlexGroup**。

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。