



适用于 **Docker** 的 Trident

Trident

NetApp
July 01, 2026

目录

| | |
|--|----|
| 适用于 Docker 的 Trident | 1 |
| 部署的前提条件 | 1 |
| 验证要求 | 1 |
| NVMe 工具 | 3 |
| FC 工具 | 4 |
| 部署 Trident | 6 |
| Docker 托管插件方法（版本 1.13/17.03 及更高版本） | 6 |
| 传统方法（1.12 版或更早版本） | 8 |
| 在系统启动时启动 Trident | 9 |
| 升级或卸载 Trident | 10 |
| 升级 | 10 |
| 卸载 | 12 |
| 使用卷 | 12 |
| 创建卷 | 12 |
| 删除卷 | 13 |
| 克隆卷 | 13 |
| 访问外部创建的卷 | 14 |
| 特定于驱动程序的卷选项 | 14 |
| 收集日志 | 19 |
| 收集日志以进行故障排除 | 19 |
| 常规故障排除提示 | 19 |
| 管理多个 Trident 实例 | 20 |
| Docker 托管插件步骤（版本 1.13/17.03 或更高版本） | 20 |
| 传统（1.12 版或更早版本）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，并显示不存在此文件或目录错误。 | 31 |
| 卷名长度必须至少为 2 个字符。 | 32 |
| Docker Swarm 的某些行为会阻止 Trident 支持每种存储和驱动程序组合。 | 32 |
| 如果正在调配 FlexGroup，则当第二个 FlexGroup 与正在调配的 FlexGroup 具有一个或多个相同聚合时，ONTAP 不会调配第二个 FlexGroup。 | 32 |

适用于 Docker 的 Trident

部署的前提条件

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。

验证要求

- 验证您的部署是否符合所有 ["要求"](#)。
- 验证您是否安装了受支持的 Docker 版本。如果您的 Docker 版本已过期，["安装或更新它"](#)。

```
docker --version
```

- 验证协议先决条件是否已在主机上安装并配置。

NFS 工具

使用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 工具后重新启动工作节点，以防止将卷附加到容器时出现故障。

iSCSI 工具

使用操作系统命令安装 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. 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
```



请确保 `etc/multipath.conf` 包含 `find_multipaths no` 下的 `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. 检查 open-iscsi 版本是否为 2.0.874-5ubuntu2.10 或更高版本（用于 bionic）或 2.0.874-7.1ubuntu6.1 或更高版本（用于 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
```



请确保 `etc/multipath.conf` 包含 `find_multipaths no` 下的 `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

使用操作系统命令安装 NVMe 工具。



- NVMe 需要 RHEL 9 或更高版本。
- 如果 Kubernetes 节点的内核版本太旧，或者 NVMe 包不适用于您的内核版本，则可能需要将节点的内核版本更新为使用 NVMe 包的内核版本。

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

使用操作系统的命令安装 FC 工具。

- 当使用运行 RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) 的工作节点并配合 FC PVs 时，请在 `discard StorageClass` 中指定 `mountOption` 以执行内联空间回收。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
```



请确保 `etc/multipath.conf` 包含 `find_multipaths no` 下的 `defaults`。

3. 请确保 `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
```



请确保 `etc/multipath.conf` 包含 `find_multipaths no` 下的 `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 存储平台提供了与 Docker 生态系统的直接集成。它支持从存储平台到 Docker 主机的存储资源调配和管理，并具有将来添加其他平台的框架。

Trident 可以在同一主机上并发运行多个实例。这允许同时连接到多个存储系统和存储类型，并可以自定义用于 Docker 卷的存储。

您需要什么

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

Docker 托管插件方法（版本 1.13/17.03 及更高版本）



开始之前

如果您在传统的守护进程方法中使用了 Docker 1.13/17.03 之前的 Trident，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. 停止所有运行的实例：

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

2. 重新启动 Docker。

```
systemctl restart docker
```

3. 确保已安装 Docker Engine 17.03（新 1.13）或更高版本。

```
docker --version
```

如果您的版本已过期，["安装或更新您的安装"](#)。

步骤

1. 创建配置文件并指定选项，如下所示：

- config: 默认文件名为 config.json，但是您可以通过指定 config 选项和文件名来使用您选择的任何名称。配置文件必须位于主机系统上的 /etc/netappdvp 目录中。
- 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 true，则覆盖日志级别。
 - i. 创建配置文件的位置：

```
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。将 <version> 替换为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 从已配置的系统使用存储。

a. 创建名为"firstVolume"的卷:

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

b. 在容器启动时创建默认卷:

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

c. 删除卷 "firstVolume":

```
docker volume rm firstVolume
```

传统方法（1.12 版或更早版本）

开始之前

1. 确保您拥有 Docker 1.10 或更高版本。

```
docker --version
```

如果您的版本已过期，请更新您的安装。

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

或者 ["按照您的发行版说明进行操作"](#)。

2. 请确保为您的系统配置 NFS 和/或 iSCSI。

步骤

1. 安装并配置 NetApp Docker Volume 插件：

- a. 下载并解压缩应用程序：

```
wget  
https://github.com/NetApp/trident/releases/download/10.0/trident-  
installer-26.02.0.tar.gz  
tar xzf trident-installer-26.02.0.tar.gz
```

- b. 移动到 bin 路径中的位置：

```
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. 创建配置文件的位置：

```
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 守护程序。

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



除非指定，否则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. 启动容器时预配 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
```

在系统启动时启动 Trident

可以在 Git 存储库中的 `contrib/trident.service.example` 找到基于 systemd 的系统的示例单元文件。要将文件与 RHEL 一起使用，请执行以下操作：

1. 将文件复制到正确的位置。

如果有多个实例正在运行，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. 编辑文件，更改描述（第 2 行）以匹配驱动程序名称和配置文件路径（第 9 行）以反映您的环境。
3. 重新加载 `systemd` 以使其获取更改：

```
systemctl daemon-reload
```

4. 启用服务。

此名称因您在 `/usr/lib/systemd/system` 目录中为文件命名的名称而异。

```
systemctl enable trident
```

5. 启动服务。

```
systemctl start trident
```

6. 查看状态。

```
systemctl status trident
```



每当您修改单元文件时，请运行 `systemctl daemon-reload` 命令以了解更改。

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。在升级过程中，会有一段短暂的时间，针对插件的 ``docker volume`` 命令将不会成功，并且在插件再次运行之前，应用程序将无法挂载卷。在大多数情况下，这只是几秒钟的问题。

升级

执行以下步骤以升级 Docker 的 Trident。

步骤

1. 列出现有卷：

```
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。您应该直接从 `netapp/ndvp-plugin` 镜像升级到 `netapp/trident-plugin` 镜像。

4. 启用插件:

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

5. 验证插件是否已启用:

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

6. 验证卷是否可见:

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



如果要从旧版 Trident (20.10 之前) 升级到 Trident 20.10 或更高版本, 可能会遇到错误。有关详细信息, 请参阅 "[已知问题](#)"。如果遇到错误, 应先禁用插件, 然后删除插件, 然后通过传递额外的配置参数安装所需的 Trident 版本: `docker plugin install netapp/trident-plugin:20.10 --alias netapp --grant-all-permissions config=config.json`

卸载

请执行以下步骤卸载 Docker 的 Trident。

步骤

1. 删除插件创建的任何卷。
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. 删除插件:

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

使用卷

您可以在需要使用指定 Trident 驱动程序名称的标准 `docker volume` 命令轻松创建、克隆和删除卷。

创建卷

- 使用默认名称的驱动程序创建卷:

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

- 使用特定 Trident 实例创建卷:

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



如果未指定任何 "[options](#)", 则使用驱动程序的默认值。

- 覆盖默认卷大小。请参见以下示例，使用驱动程序创建 20 GiB 的卷：

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



卷大小表示为包含整数值和可选单位的字符串（例如：10G、20GB、3TiB）。如果未指定单位，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）。速记单位使用 2 的幂（G = GiB、T = TiB、...）。

删除卷

- 像任何其他 Docker 卷一样删除该卷：

```
docker volume rm firstVolume
```



使用 `solidfire-san` 驱动程序时，上面的示例会删除并清除该卷。

执行以下步骤以升级 Docker 的 Trident。

克隆卷

使用 `ontap-nas`、`ontap-san``和 ``solidfire-san`` 存储驱动程序时，Trident 可以克隆卷。使用 ``ontap-nas-flexgroup` 或 `ontap-nas-economy` 驱动程序时，不支持克隆。从现有卷创建新卷将导致创建新的快照。

- 检查卷以枚举快照：

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

- 从现有卷创建新卷。这将导致创建新的快照：

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

- 从卷上的现有快照创建新卷。这不会创建新快照：

```
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

您可以通过使用 Trident 访问容器创建的外部块设备（或其克隆），*仅当*它们没有分区并且其文件系统受 Trident 支持时（例如：`ext4` 格式的 `/dev/sdc1` 将无法通过 Trident 访问）。

特定于驱动程序的卷选项

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

在卷创建操作期间使用这些选项非常简单。在 CLI 操作期间使用 `-o` 操作符提供选项和值。这些会覆盖 JSON 配置文件中的任何等效值。

ONTAP 卷选项

NFS、iSCSI 和 FC 的卷创建选项包括：

| 选项 | 说明 |
|-----------------|---|
| size | 卷的大小，默认为 1 GiB。 |
| spaceReserve | 薄型或厚型配置卷，默认为薄型。有效值为 none（精简配置）和 volume（厚配置）。 |
| snapshotPolicy | 这会将快照策略设置为所需的值。默认值为 none，这意味着不会自动为卷创建快照。除非您的存储管理员修改，否则所有 ONTAP 系统上都存在名为"default"的策略，该策略会创建并保留六个每小时快照、两个每日快照和两个每周快照。可以通过浏览到卷中任何目录中的`.snapshot`目录来恢复快照中保留的数据。 |
| snapshotReserve | 这会将快照预留设置为所需的百分比。默认情况下没有值，这意味着如果你选择了 snapshotPolicy，ONTAP 会选择 snapshotReserve（通常为 5%）；如果 snapshotPolicy 为 none，则为 0%。你可以在配置文件中为所有 ONTAP 后端设置默认的 snapshotReserve 值，并且除了 ontap-nas-economy 以外，你可以将其作为所有 ONTAP 后端的卷创建选项。 |
| splitOnClone | 克隆卷时，这将导致 ONTAP 立即将克隆从其父级拆分出来。默认值为 false。克隆卷的某些用例最好在创建时立即将克隆从其父级拆分出来，因为不太可能有任何提高存储效率的机会。例如，克隆空数据库可以节省大量时间，但节省的存储空间很少，因此最好立即拆分克隆。 |
| encryption | <p>在新卷上启用 NetApp Volume Encryption (NVE)；默认为 false。必须在群集上许可并启用 NVE 才能使用此选项。</p> <p>如果在后端启用了 NAE，则在 Trident 中配置的任何卷都将启用 NAE。</p> <p>有关更多信息，请参阅："Trident 如何与 NVE 和 NAE 配合使用"。</p> |
| tieringPolicy | 设置要用于卷的分层策略。这决定了数据在变为非活动（冷）时是否移动到云层。 |

以下其他选项*仅适用于* NFS：

| 选项 | 说明 |
|-----------------|--|
| unixPermissions | 这会控制卷本身的权限集。默认情况下，权限将设置为 <code>---rwxr-xr-x</code> ，或使用数字表示法 <code>0755</code> ， <code>root</code> 将成为所有者。文本或数字格式都可以使用。 |
| snapshotDir | 将此设置为 <code>true</code> 将使 <code>.snapshot</code> 目录对访问卷的客户端可见。默认值为 <code>false</code> ，这意味着默认情况下会禁用 <code>.snapshot</code> 目录的可见性。某些镜像（例如官方 MySQL 镜像）在 <code>.snapshot</code> 目录可见时无法按预期运行。 |
| exportPolicy | 设置要用于该卷的导出策略。默认值为 <code>default</code> 。 |
| securityStyle | 设置用于访问卷的安全样式。默认值为 <code>unix</code> 。有效值为 <code>unix</code> 和 <code>mixed</code> 。 |

以下附加选项*仅*适用于 iSCSI:

| 选项 | 说明 |
|-----------------|---|
| fileSystemType | 设置用于格式化 iSCSI 卷的文件系统。默认值为 <code>ext4</code> 。有效值为 <code>ext3</code> 、 <code>ext4</code> 和 <code>xfso</code> 。 |
| spaceAllocation | 将此设置为 <code>false</code> 将关闭 LUN 的空间分配功能。默认值为 <code>true</code> ，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。此选项还使 ONTAP 能够在主机删除数据时自动回收空间。 |

示例

请参见以下示例:

- 创建 10 GiB 卷:

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

- 使用快照创建 100 GiB 卷:

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

- 创建启用了 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，并且快照策略为 none，则 Trident 使用 0% 的快照保留。

- 创建没有快照策略和快照预留的卷：

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

- 创建没有快照策略且自定义快照保留为 10% 的卷：

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

- 创建具有快照策略和 10% 自定义快照预留的卷：

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

- 使用快照策略创建卷，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 软件卷选项

Element 软件选项显示与卷关联的大小和服务质量 (QoS) 策略。创建卷时，与其关联的 QoS 策略使用 `-o type=service_level` 命名法指定。

使用 Element 驱动程序定义 QoS 服务级别的第一步是至少创建一种类型，并指定与配置文件中的名称关联的最小、最大和突发 IOPS。

其他 Element 软件卷创建选项包括：

| 选项 | 说明 |
|-----------|--|
| size | 卷的大小，默认为 1 GiB 或配置项....."defaults": {"size": "5G"}。 |
| blocksize | 使用 512 或 4096，默认值为 512 或 config 条目 DefaultBlockSize。 |

示例

请参见以下具有 QoS 定义的配置文件示例：

```
{
  "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
      }
    }
  ]
}
```

在以上配置中，我们三个策略定义：Bronze、Silver 和 Gold。这些名称是任意的。

- 创建 10 GiB Gold 卷：

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

- 创建 100 GiB Bronze 卷：

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

收集日志

您可以收集日志以获得故障排除方面的帮助。用于收集日志的方法因运行 Docker 插件的方式而异。

收集日志以进行故障排除

步骤

1. 如果使用推荐的托管插件方法（即使用 `docker plugin` 命令）运行 Trident，请按如下方式查看：

```
docker plugin ls
```

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

标准日志记录级别应允许您诊断大多数问题。如果您发现这还不够，则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的插件：

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

或者，在已安装插件时启用调试日志记录：

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

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

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

3. 如果您在主机上运行二进制文件本身，则可在主机的 `/var/log/netappdvp` 目录中找到日志。要启用调试日志记录，请在运行插件时指定 `-debug`。

常规故障排除提示

- 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。当发生这种情况时，您尝试安装或启用插件时可能会看到这样的消息：

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

这意味着插件无法启动。幸运的是，该插件具有全面的日志记录功能，可以帮助您诊断可能遇到的大多数问题。

- 如果将 PV 挂载到容器时出现问题，请确保 `rpcbind` 已安装并正在运行。请使用主机操作系统所需的软件包管理器，并检查 `rpcbind` 是否正在运行。可以通过运行 `systemctl status rpcbind` 或其等效项来检查 `rpcbind` 服务的状态。

管理多个 Trident 实例

当您希望同时提供多个存储配置时，需要多个 Trident 实例。多个实例的关键是使用带有容器化插件的 `--alias` 选项或在主机上实例化 Trident 时使用 `--volume-driver` 选项为它们提供不同的名称。

Docker 托管插件步骤（版本 1.13/17.03 或更高版本）

1. 启动指定别名和配置文件的第一个实例。

```
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. 创建将别名指定为驱动程序名称的卷。

例如，对于黄金卷：

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

例如，对于银卷：

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

传统（1.12 版或更早版本）的步骤

1. 使用自定义驱动程序 ID 启动具有 NFS 配置的插件：

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

2. 使用自定义驱动程序 ID 通过 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 | 说明 | 示例 |
|-------------------|------------------------|---|
| version | 配置文件版本号 | 1 |
| storageDriverName | 存储驱动程序的名称 | ontap-nas, ontap-san, ontap-nas-economy, ontap-nas-flexgroup, solidfire-san |
| storagePrefix | 卷名的可选前缀。默认值：netappdvp_ | staging_ |
| limitVolumeSize | 对卷大小的可选限制。默认值："（未强制执行） | 10g |



不要对 Element 后端使用 `storagePrefix` (包括默认值)。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 `tenantID` 进行 Docker 卷映射, 或者在可能使用任何名称 `munging` 的情况下使用 Docker 版本、驱动程序信息和 Docker 原始名称填充的属性数据。

默认选项可用于避免在创建的每个卷上指定它们。该 `size` 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 | 说明 | 示例 |
|-------------------|-------------------|-----|
| <code>size</code> | 新卷的可选默认大小。默认值: 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 | 说明 | 示例 |
|----------------------------|---|----------------------|
| <code>managementLIF</code> | ONTAP 管理 LIF 的 IP 地址。您可以指定完全限定的域名 (FQDN)。 | 10.0.0.1 |
| <code>dataLIF</code> | 协议 LIF 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> 。可以指定要用于 NFS 挂载操作的完全限定域名 (FQDN), 允许您创建轮询 DNS 以跨多个 <code>dataLIF</code> 进行负载平衡。 ONTAP SAN 驱动程序: 不为 iSCSI 或 FC 指定。Trident 使用 "ONTAP 选择性 LUN 映射" 来发现建立多路径会话所需的 iSCSI 或 FC LIF。如果明确定义了 <code>dataLIF</code> , 则会生成警告。 | 10.0.0.2 |
| <code>svm</code> | 要使用的 Storage Virtual Machine (如果 Management LIF 是集群 LIF, 则必需) | <code>svm_nfs</code> |
| <code>username</code> | 连接到存储设备的用户名 | <code>vsadmin</code> |
| <code>password</code> | 连接到存储设备的密码 | <code>secret</code> |

| 选项 | 说明 | 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 <code>ontap-nas-flexgroup</code> 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，如果使用率高于此百分比，则配置失败 | 75% |
| nfsMountOptions | NFS 挂载选项的细粒度控制；默认为 "-o nfsvers=3"。仅适用于 <code>`ontap-nas`</code> 和 <code>`ontap-nas-economy`</code> 驱动程序 " 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的 <code>igroups`</code> 作为 <code>`netappdvp`</code> 。 此值不能更改或省略。 仅适用于 <code>ontap-san</code> 驱动程序。 | netappdvp |
| limitVolumeSize | 可请求的最大卷大小。 | 300g |
| qtreesPerFlexvol | 每个 FlexVol 的最大 qtrees，必须在 [50, 300] 范围内，默认值为 200。 对于 <code>ontap-nas-economy</code> 驱动程序，此选项允许自定义每个 FlexVol 的最大 <code>qtree</code> 数。 | 300 |
| sanType | *仅支持 <code>ontap-san</code> 驱动程序。*用于选择 <code>iscsi</code> iSCSI、 <code>nvme</code> NVMe/TCP 或 <code>fcp</code> 光纤通道 (FC) 上的 SCSI。 | iscsi 如果为空 |
| limitVolumePoolSize | *仅支持 <code>`ontap-san-economy`</code> 和 <code>`ontap-san-economy`</code> 驱动程序。*限制 ONTAP <code>ontap-nas-economy</code> 和 <code>ontap-SAN-economy</code> 驱动程序中的 FlexVol 大小。 | 300g |

默认选项可用于避免在创建的每个卷上指定它们：

| 选项 | 说明 | 示例 |
|-------------------|---|--|
| spaceReserve | 空间预留模式； none（精简配置）或 volume（厚配置） | none |
| snapshotPolicy | 要使用的 Snapshot 策略，默认为 none | none |
| snapshotReserve | Snapshot 预留百分比，默认为 "" 以接受 ONTAP 默认值 | 10 |
| splitOnClone | 创建时从其父级拆分克隆，默认为 false | false |
| encryption | <p>在新卷上启用 NetApp Volume Encryption (NVE)；默认为 false。必须在群集上许可并启用 NVE 才能使用此选项。</p> <p>如果在后端启用了 NAE，则在 Trident 中配置的任何卷都将启用 NAE。</p> <p>有关更多信息，请参阅："Trident 如何与 NVE 和 NAE 配合使用"。</p> | true |
| unixPermissions | 调配的 NFS 卷的 NAS 选项，默认为 777 | 777 |
| snapshotDir | 用于访问 <code>`.snapshot`</code> 目录的 NAS 选项。 | <p>启用或禁用对 <code>`.snapshot`</code> 目录的访问。</p> <p>有效值：true, false。默认行为可能因 ONTAP 和 Trident 版本而异。明确设置以确保行为一致。</p> |
| exportPolicy | 要使用的 NFS 导出策略的 NAS 选项，默认为 default | default |
| securityStyle | <p>用于访问配置的 NFS 卷的 NAS 选项。</p> <p>NFS 支持 <code>mixed`</code> 和 <code>`unix`</code> 安全样式。默认值为 <code>`unix`</code>。</p> | unix |
| fileSystemType | 选择文件系统类型的 SAN 选项，默认为 ext4 | xfs |
| tieringPolicy | 要使用的分层策略，默认值为 none。 | none |
| skipRecoveryQueue | 在卷删除过程中，绕过存储中的恢复队列并立即删除卷。 | `` |

扩展选项

``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 FlexVols，集群最大容量为 12,000 个 FlexVol 卷。如果您的 Docker 卷要求符合该限制，则 ``ontap-nas`` 驱动程序是首选的 NAS 解决方案，因为 FlexVols 提供了额外的功能，例如 Docker 卷粒度快照和克隆。

如果您需要超出 FlexVol 限制的 Docker 卷，请选择 `ontap-nas-economy` 或 `ontap-san-economy` 驱动程序。

``ontap-nas-economy`` 驱动程序在自动管理的 FlexVol 卷池中将 Docker 卷创建为 ONTAP Qtree。Qtree 提供了更大的扩展性，每个集群节点高达 100,000 个，每个集群高达 2,400,000 个，但牺牲了一些功能。``ontap-nas-economy`` 驱动程序不支持 `Docker-volume-granular` 快照或克隆。



Docker Swarm 当前不支持该 ``ontap-nas-economy`` 驱动程序，因为 Docker Swarm 不跨多个节点编排卷创建。

``ontap-san-economy`` 驱动程序在自动管理的 FlexVol 卷的共享池中将 Docker 卷创建为 ONTAP LUN。这样，每个 FlexVol 就不仅限于一个 LUN，还可以为 SAN 工作负载提供更好的可扩展性。根据存储阵列，ONTAP 支持每个集群最多 16384 个 LUN。因为卷是底层的 LUN，所以此驱动程序支持 `Docker-volume-granular` 快照和克隆。

选择 `ontap-nas-flexgroup` 驱动程序，以提升对单一卷的并行性，该卷可扩展到拥有数十亿文件的 PB 级别。FlexGroups 的一些理想用例包括 AI/ML/DL、大数据与分析、软件构建、流媒体、文件存储库等。Trident 在配置 FlexGroup 卷时，会使用分配给 SVM 的所有聚合体。Trident 中对 FlexGroup 的支持还需注意以下事项：

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

有关 FlexGroups 和适用于 FlexGroups 的工作负载的信息，请参见 "[NetApp FlexGroup 卷最佳实践和实施指南](#)"。

要在同一环境中获得高级功能和大规模扩展，您可以运行 Docker Volume Plugin 的多个实例，一个使用 `ontap-nas`，另一个使用 `ontap-nas-economy`。

Trident 的自定义 ONTAP 角色

您可以使用最低权限创建 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 使用您创建的 ONTAP 集群角色来执行操作。

有关创建 Trident 自定义角色的详细信息，请参见 ["Trident 自定义角色生成器"](#)。

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 **System Manager**

在 ONTAP System Manager 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 **Cluster > Settings**。

(或) 要在 SVM 级别创建自定义角色，请选择*存储 > Storage VM > required SVM> 设置 > 用户和角色*。

- b. 选择 **Users and Roles** 旁边的箭头图标 (→)。
- c. 在 **Roles** 下选择 **+Add**。
- d. 定义角色的规则并单击 **Save**。

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. 选择 **Users** 下的添加图标 +。
- b. 选择所需的用户名，然后在 **Role** 下拉菜单中选择一个角色。
- c. 单击 **Save**。

有关详细信息，请参见以下页面：

- ["用于管理 ONTAP 的自定义角色"](#) 或 ["定义自定义角色"](#)
- ["使用角色和用户"](#)

ONTAP 配置文件示例

ONTAP-nas`<code>` 驱动程序的 NFS`</code>` 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`<code>`</code> 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
}
```

用于 <code>ontap-san</code> 驱动程序的基于 FC 的 SCSI 示例

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

Element 软件配置

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

| 选项 | 说明 | 示例 |
|------------|--|--|
| Endpoint | https://<login>:<password>@<mvip>/json-rpc/<element-version> | https://admin:admin@192.168.160.3/json-rpc/8.0 |
| SVIP | iSCSI IP 地址和端口 | 10.0.0.7:3260 |
| TenantName | 要使用的 SolidFire 租户（如果未找到，则创建） | docker |

| 选项 | 说明 | 示例 |
|------------------|--|------------|
| InitiatorIFace | 在将 iSCSI 流量限制为非默认接口时指定接口 | default |
| Types | QoS 规范 | 请参见以下示例 |
| LegacyNamePrefix | 升级版 Trident 安装的前缀。如果您使用了 1.3.2 之前的 Trident 版本并对现有卷执行升级，则需要设置此值以访问通过卷名称方法映射的旧卷。 | netappdvp- |

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

示例 **Element** 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

查找有关在 Docker 中使用 Trident 的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，并显示不存在此文件或目录错误。

临时解决策

1. 禁用插件。

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

2. 删除插件。

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

3. 通过提供额外 `config` 参数重新安装插件。

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

卷名长度必须至少为 **2** 个字符。



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

Docker Swarm 的某些行为会阻止 **Trident** 支持每种存储和驱动程序组合。

- Docker Swarm 目前使用卷名而不是卷 ID 作为其唯一的卷标识符。
- 卷请求同时发送到 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` 驱动程序的功能，它们是唯一能够在这些限制范围内运行的驱动程序。

其余的驱动程序会受到诸如竞争条件之类的问题的影响，这些问题可能会导致为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不同 ID 的功能。

NetApp 已经向 Docker 团队提供了反馈，但没有任何未来追索的迹象。

如果正在调配 **FlexGroup**，则当第二个 **FlexGroup** 与正在调配的 **FlexGroup** 具有一个或多个相同聚合时，**ONTAP** 不会调配第二个 **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。