适用于 Docker 的 Astra Trident

Astra Trident

NetApp
January 14, 2026

目录

适用于 Docker 的 Astra Trident

部署的前提条件

在部署 Astra Trident 之前，您必须在主机上安装和配置必要的协议前提条件。

验证要求

- 验证您的部署是否满足所有要求 "要求"。
- 验证您是否安装了受支持的 Docker 版本。如果您的 Docker 版本已过时， "安装或更新它"。

```
docker --version
```

- 验证是否已在主机上安装和配置协议前提条件。

NFS工具

使用适用于您的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、以防止在将卷连接到容器时失败。

iSCSI工具

使用适用于您的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. 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. 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
```

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

5. 请确保 `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 2.0.877-5ubuntu2.10 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` 下 `defaults`。

5. 请确保 `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
```

NVMe工具

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
```

部署 Astra Trident

适用于Docker的A作用 是直接与适用于NetApp存储平台的Docker生态系统集成。它支持从存储平台到 Docker 主机的存储资源配置和管理，并提供一个框架，用于在未来添加其他平台。

Astra Trident 的多个实例可以同时在同一主机上运行。这样可以同时连接到多个存储系统和存储类型，并能够自定义用于 Docker 卷的存储。

您需要的内容

请参见 "部署的前提条件"。确保满足前提条件后，即可部署 Astra Trident 。

Docker 托管插件方法（ 1.13/17.03 及更高版本）

开始之前

如果您在传统守护进程方法中使用了 Astra Trident 之前的 Docker 1.13/ 17.03 ，请确保在使用受管插件方法之前停止 Astra 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. 重新启动 Docker 。

```
systemctl restart docker
```

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

```
docker --version
```

如果您的版本已过期， "安装或更新安装"。

步骤

1. 创建配置文件并按如下所示指定选项：
   - `config`：默认文件名为 `config.json`但是、您可以通过指定来使用所选的任何名称 `config` 选项和文件名。配置文件必须位于中 `/etc/netappdvp` 主机系统上的目录。
   - `log-level`：指定日志记录级别 (`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. 为配置文件创建一个位置：

     ```
     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. 使用受管插件系统启动 Astra Trident 。替换 `<version>` 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 Astra Trident 消耗已配置系统中的存储。
   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** 或更早版本）

开始之前

1. 确保您已安装 Docker 版本 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 "按照适用于您的分发版本的说明进行操作"。

2. 确保已为您的系统配置 NFS 和 / 或 iSCSI 。

步骤

1. 安装和配置 NetApp Docker 卷插件：

a. 下载并解压缩应用程序：

```
wget
https://github.com/NetApp/trident/releases/download/v24.10.0/trident-
installer-24.02.0.tar.gz
tar zxf trident-installer-24.02.0.tar.gz
```

b. 移动到托箱路径中的某个位置：

```
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. 为配置文件创建一个位置：

```
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三叉进制守护进程。

```
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。

启动守护进程后，您可以使用 Docker 命令行界面创建和管理卷

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
```

在系统启动时启动 Astra Trident

有关基于系统d的系统的示例单元文件、请参见 `contrib/trident.service.example` 在Git repo.要对RHEL使用此文件、请执行以下操作：

1. 将文件复制到正确的位置。

   如果正在运行多个实例，则单元文件应使用唯一名称。

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

2. 编辑文件，更改问题描述（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 以载入更改：

   ```
   systemctl daemon-reload
   ```

4. 启用服务。

   此名称因您在中命名文件而异 `/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
   ```

每当您修改单元文件时、请运行 `systemctl daemon-reload` 命令以使其能够识别所做的更改。

升级或卸载 Astra Trident

您可以安全地升级适用于 Docker 的 Astra Trident ，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 `docker volume` 定向到插件的命令将不会成功、应用程序将无法挂载卷、直到插件重新运行为止。在大多数情况下，这只需要几秒钟。

升级

执行以下步骤以升级适用于 Docker 的 Astra 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
```

> Astra Trident 18.01 版取代了 nDVP 。您应直接从升级 `netapp/ndvp-plugin` 以图像形式显示到 `netapp/trident-plugin` 图像。

4. 启用插件：

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

5. 验证是否已启用此插件：

```
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
```

如果要从旧版本的 Astra Trident （ 20.10 之前的版本）升级到 Astra Trident 20.10 或更高版本，则可能会遇到错误。有关详细信息，请参见 "已知问题"。如果遇到此错误、则应先禁用此插件、然后删除此插件、再通过传递一个额外的配置参数来安装所需的Astra Trident版本： `docker plugin install netapp/trident-plugin:20.10 --alias netapp --grant-all-permissions config=config.json`

卸载

执行以下步骤卸载适用于 Docker 的 Astra Trident 。

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. 删除插件：

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

使用卷

您可以使用标准轻松创建、克隆和删除卷 `docker volume` 根据需要指定了Astra Trident驱动程序名称的命令。

创建卷

- 使用默认名称创建包含驱动程序的卷：

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

- 使用特定的 Astra Trident 实例创建卷：

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

> 如果未指定任何 "选项"，将使用驱动程序的默认值。

- 覆盖默认卷大小。要使用驱动程序创建 20GiB 卷，请参见以下示例：

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

> 卷大小以字符串表示，该字符串包含一个包含可选单元的整数值（例如： 10 G ， 20 GB ， 3 TiB ）。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 ）。速率单位使用 2 的电流（ G = GiB ， T = TiB ，…）。

删除卷

- 像删除任何其他 Docker 卷一样删除此卷：

```
docker volume rm firstVolume
```

> 使用时 `solidfire-san` 驱动程序中、上述示例将删除并清除卷。

执行以下步骤以升级适用于 Docker 的 Astra Trident 。

克隆卷

使用时 `ontap-nas`， `ontap-san`， `solidfire-san`，和 `gcp-cvs storage drivers`、Astra Trident可以克隆卷。使用时 `ontap-nas-flexgroup` 或 `ontap-nas-economy` 驱动程序、不支持克隆。从现有卷创建新卷将创建新快照。

- 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。此操作不会创建新快照：

```
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

如果容器没有分区、并且Astra Trident支持其文件系统(例如：an)、则可以使用Trident * only *通过容器访问外部创建的块设备(或其克隆) `ext4`格式化 `/dev/sdc1` 无法通过Astra Trident访问)。

驱动程序专用的卷选项

每个存储驱动程序都有一组不同的选项，您可以在创建卷时指定这些选项来自定义结果。有关适用于您配置的存储系统的选项，请参见以下内容。

在卷创建操作期间使用这些选项非常简单。使用提供选项和值 `-o` 在命令行界面操作期间执行此操作。这些参数将覆盖 JSON 配置文件中的任何等效值。

ONTAP 卷选项

NFS 和 iSCSI 的卷创建选项包括以下内容：

选项	Description
`size`	卷的大小默认为 1 GiB 。
`spaceReserve`	精简或厚配置卷，默认为精简。有效值为 `none` (精简配置)和 `volume` (厚配置)。
`snapshotPolicy`	此操作会将 Snapshot 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。除非存储管理员修改，否则所有 ONTAP 系统上都存在一个名为 "defaultion" 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。通过浏览到、可以恢复快照中保留的数据 `.snapshot` 卷中任意目录中的目录。
`snapshotReserve`	此操作会将快照预留设置为所需百分比。默认值为 no 值，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 ，则选择 0% 。您可以在配置文件中为所有 ONTAP 后端设置默认 snapshotReserve 值，并可将其用作除 ontap-nas-economy. 以外的所有 ONTAP 后端的卷创建选项。
`splitOnClone`	克隆卷时，此操作将使发生原因 ONTAP 立即从其父卷拆分克隆。默认值为 `false`。在克隆卷的某些使用情形中，最好在创建后立即将克隆从其父卷中拆分，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、但只能节省很少的存储空间、因此最好立即拆分克隆。

选项	Description
encryption	在新卷上启用NetApp卷加密(NVE)；默认为 false。要使用此选项，必须在集群上获得 NVE 的许可并启用 NVE 。 如果在后端启用了NAE、则在Astra Trident中配置的任何卷都将启用NAE。 有关详细信息、请参见： "Astra Trident如何与NVE和NAE配合使用"。
tieringPolicy	设置要用于卷的分层策略。这将决定数据在变为非活动状态（冷）时是否移至云层。

以下附加选项适用于 NFS * 仅 * ：

选项	Description
unixPermissions	此选项用于控制为卷本身设置的权限。默认情况下、权限将设置为 `---rwxr-xr-x`或以数字表示法0755、和 `root 将成为所有者。文本或数字格式均可使用。
snapshotDir	将其设置为 true 将创建 .snapshot 访问卷的客户端可以看到的目录。默认值为 false、表示的可见性 .snapshot 默认情况下、目录处于禁用状态。某些映像(例如官方MySQL映像)在出现时无法按预期运行 .snapshot 目录可见。
exportPolicy	设置要用于卷的导出策略。默认值为 default。
securityStyle	设置用于访问卷的安全模式。默认值为 unix。有效值为 unix 和 mixed。

以下附加选项适用于 iSCSI * 仅 * ：

选项	Description
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、表示当卷空间用尽且卷中的LUN无法接受写入时、ONTAP 会向主机发出通知。此选项还允许 ONTAP 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
```

- 创建具有快照的 100GiB 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 20MiB 。

如果未指定快照预留且快照策略为 `none`、Trident将使用0%的快照预留。

- 创建无快照策略且无快照预留的卷：

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

- 创建一个无快照策略且自定义快照预留为 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 的默认快照预留（通常为 5% ）：

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

Element 软件卷选项

Element 软件选项会显示与卷关联的大小和服务质量（ QoS ）策略。创建卷时、将使用指定与其关联的QoS策略 `-o type=service_level` 术语。

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	**Description**
`size`	卷的大小，默认为 1GiB 或配置条目 … " 默认值 " ：｛ "size" ： "5c" ｝ 。
`blocksize`	使用 512 或 4096 ，默认为 512 或配置条目 DefaultBlockSize 。

示例

请参见以下包含 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，我们有三个策略定义：铜牌，银牌和金牌。这些名称是任意的。

- 创建 10 GiB 黄金卷：

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

- 创建 100GiB 铜牌卷：

```
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. 如果您使用建议的托管插件方法(例如、使用 `docker plugin` 命令)、请按如下所示查看它们：

```
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，请安装启用了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二进制文件本身、则主机的中会显示日志 `/var/log/netappdvp` 目录。要启用调试日志记录、请指定 `-debug` 运行插件时。

一般故障排除提示

- 新用户遇到的最常见问题是配置不当，导致插件无法初始化。如果发生这种情况，在尝试安装或启用插件时，您可能会看到如下消息：

  ```
  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，该插件已构建了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。

- 如果在将PV挂载到容器时出现问题、请确保这样 `rpcbind` 已安装且正在运行。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。您可以通过运行来检查rpcbind服务的状态 `systemctl status rpcbind` 或其等效项。

管理多个 Astra Trident 实例

如果希望同时提供多个存储配置，则需要多个 Trident 实例。多个实例的关键是使用为其提供不同的名称 `--alias` 选项、或者 `--volume-driver` 在主机上实例化Trident时的选项。

Docker 托管插件（ 1.13/17.03 或更高版本）的步骤

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，指定其他别名和配置文件。

```
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，对于银牌卷：

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

传统（ 1.12 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 启动具有 iSCSI 配置的插件：

```
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

请参见适用于您的 Astra 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 Astra Trident 配置，而不考虑所使用的存储平台。

选项	**Description**	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`	卷名称的可选前缀。默认值：`netappdvp_`。	`staging_`
`limitVolumeSize`	卷大小的可选限制。默认值：""(未强制实施)	`10g`

请勿使用 `storagePrefix` 元素后端的(包括默认值)。默认情况下、 `solidfire-san` 驱动程序将忽略此设置、而不使用前缀。我们建议使用特定的租户 ID 进行 Docker 卷映射，或者在可能已使用任何名称的情况下使用 Docker 中填充的 Docker 版本，驱动程序信息和原始名称的属性数据。

您可以使用默认选项来避免在创建的每个卷上指定这些选项。。 `size` 选项可用于所有控制器类型。有关如何设置默认卷大小的示例，请参见 ONTAP 配置一节。

选项	Description	示例
size	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，在使用 ONTAP 时，还可以使用以下顶级选项。

选项	Description	示例
managementLIF	ONTAP 管理 LIF 的 IP 地址。您可以指定完全限定域名（ FQDN ）。	10.0.0.1
dataLIF	协议 LIF 的 IP 地址。 • ONTAP NAS驱动程序*：建议指定 dataLIF。如果未提供此参数、则Astra Trident会从SVM提取数据LIF。您可以指定用于NFS挂载操作的完全限定域名(FQDN)、从而可以创建循环DNS、以便在多个数据LIF之间实现负载平衡。 • ONTAP SAN驱动程序*：不为iSCSI指定。 Astra Trident使用 "ONTAP 选择性LUN映射" 发现建立多路径会话所需的iSCI LIF。如果出现、则会生成警告 dataLIF 已明确定义。	10.0.0.2
svm	要使用的 Storage Virtual Machine （如果管理 LIF 为集群 LIF ，则为必填项）	svm_nfs
username	用于连接到存储设备的用户名	vsadmin
password	用于连接到存储设备的密码	secret
aggregate	要配置的聚合（可选；如果设置了聚合，则必须将其分配给 SVM ）。。 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，如果使用量超过此百分比，则配置失败	75%

选项	Description	示例
nfsMountOptions	对 NFS 挂载选项进行精细控制；默认为 -o nfsver=3 。仅适用于 **ontap-nas** 和 **ontap-nas-economy** 驱动程序。 "请参见此处的 NFS 主机配置信息"。	-o nfsvers=4
igroupName	Asta三元数据可按节点创建和管理 igroups 作为 netappdvp。 此值不能更改或省略。 仅适用于 **ontap-san** 驱动程序。	netappdvp
limitVolumeSize	可请求的最大卷大小和 qtree 父卷大小。用于 **ontap-nas-economy** 驱动程序、此选项还会限制其创建的**FlexVol**的大小。	300g
qtreesPerFlexvol	每个 FlexVol 的最大 qtree 数必须在 50 ， 300 范围内，默认值为 200 。 *用于 ontap-nas-economy 驱动程序、此选项允许自定义每个qtree*的最大FlexVol 数。	300
sanType	支持 **ontap-san** 仅限驱动程序。 使用选择 iscsi 对于iSCSI或 nvme 适用于NVMe/TCP。	iscsi 如果为空

您可以使用默认选项来避免在创建的每个卷上指定这些选项：

选项	Description	示例
spaceReserve	空间预留模式； none (精简配置)或 volume (厚)	none
snapshotPolicy	要使用的Snapshot策略、默认为 none	none
snapshotReserve	Snapshot预留百分比、默认值为""以接受ONTAP 默认值	10
splitOnClone	创建克隆时将其从父级拆分、默认为 false	false

选项	Description	示例
encryption	在新卷上启用NetApp卷加密(NVE)；默认为 false。要使用此选项，必须在集群上获得 NVE 的许可并启用 NVE 。 如果在后端启用了NAE、则在Astra Trident中配置的任何卷都将启用NAE。 有关详细信息、请参见： "Astra Trident如何与NVE和NAE配合使用"。	true
unixPermissions	对于已配置的NFS卷、NAS选项默认为 777	777
snapshotDir	用于访问的NAS选项 .snapshot 目录、默认为 false	true
exportPolicy	要使用的NFS导出策略的NAS选项、默认为 default	default
securityStyle	用于访问已配置NFS卷的NAS选项。 NFS支持 mixed 和 unix 安全模式。默认值为 unix。	unix
fileSystemType	SAN选项要选择文件系统类型、默认为 ext4	xfs
tieringPolicy	要使用的分层策略、默认为 none； snapshot-only 适用于ONTAP 9.5 SVM-DR之前的配置	none

扩展选项

。 ontap-nas 和 ontap-san 驱动程序会为每个Docker卷创建一个ONTAP FlexVol。对于每个集群节点，ONTAP 最多支持 1000 个 FlexVol ，而集群最多支持 12 ， 000 个 FlexVol 。如果您的Docker卷要求符合此限制、则会显示 ontap-nas 由于FlexVol提供了其他功能、例如Docker卷粒度快照和克隆、因此驱动程序是首选NAS解决方案。

如果所需的Docker卷数超过FlexVol 限制所能容纳的数量、请选择 ontap-nas-economy 或 ontap-san-economy 驱动程序。

。 ontap-nas-economy 驱动程序会在一个自动管理的ONTAP 卷池中将Docker卷创建为FlexVol qtree。qtree的扩展能力远高于此，每个集群节点最多可扩展 100 ， 000 个，每个集群最多可扩展 2 ， 400 ， 000 个，但某些功能会受到影响。。 ontap-nas-economy 驱动程序不支持Docker卷粒度快照或克隆。

。 ontap-nas-economy 目前、Docker Swarm不支持驱动程序、因为Swarm不会跨多个节点编排卷创建。

。 ontap-san-economy 驱动程序会在一个由自动管理的FlexVol构成的共享池中将Docker卷创建为ONTAP LUN。这样，每个 FlexVol 就不会仅限于一个 LUN ，并且可以为 SAN 工作负载提供更好的可扩展性。根据存储

阵列的不同， ONTAP 每个集群最多支持 16384 个 LUN 。由于卷是下面的 LUN ，因此此驱动程序支持 Docker 卷粒度快照和克隆。

选择 `ontap-nas-flexgroup` 驱动程序、用于将并行性提高到单个卷、该卷可以扩展到包含数十亿个文件的PB范围。FlexGroup 的一些理想用例包括 AI/ML/DL ，大数据和分析，软件构建，流式传输，文件存储库等。配置 FlexGroup 卷时， Trident 会使用分配给 SVM 的所有聚合。Trident 中的 FlexGroup 支持还需要注意以下事项：

- 需要 ONTAP 9.2 或更高版本。
- 截至本文撰写时， FlexGroup 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。

有关适用于 FlexGroup 的 FlexGroup 和工作负载的信息，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、您可以运行多个Docker卷插件实例、其中一个使用 `ontap-nas` 另一种方法是使用 `ontap-nas-economy`。

ONTAP 配置文件示例

<code>ontap-nas</code> 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"
    }
}
```

<code>ontap-nas-flexgroup</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"
}
```

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

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

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	**Description**	示例
`Endpoint`	\https：<login><element-version>：<password>@<mvip>/json-rpC/RPC	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 SolidFireF 租户（如果未找到，则创建）	`docker`
`InitiatorIFace`	将 iSCSI 流量限制为非默认接口时，请指定接口	`default`
`Types`	QoS 规范	请参见以下示例
`LegacyNamePrefix`	升级后的 Trident 安装的前缀。如果您使用的是1.3.2之前的版本的Trident并对现有卷执行升级、则需要设置此值才能访问通过volume-name方法映射的旧卷。	`netappdvp-`

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

Element 软件配置文件示例

```
{
    "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 Astra Trident 与 Docker 结合使用时的已知问题和限制的信息。

将 **Trident Docker** 卷插件从旧版本升级到 **20.10** 及更高版本会导致升级失败，并且不会显示此类文件或目录错误。

临时解决策

1. 禁用插件。

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

2. 删除此插件。

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

3. 通过提供额外的插件重新安装插件 `config` 参数。

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

卷名称的长度必须至少为 **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。 "请参见错误 25773"。

Docker Swarm 的某些行为会阻止 **Astra Trident** 在每个存储和驱动程序组合中为其提供支持。

- Docker Swarm 目前使用卷名称而非卷 ID 作为其唯一卷标识符。
- 卷请求会同时发送到 Swarm 集群中的每个节点。
- 卷插件（包括 Astra Trident ）必须在 Swarm 集群中的每个节点上独立运行。
  由于ONTAP 的工作方式和的方式 `ontap-nas` 和 `ontap-san` 驱动程序正常运行、但它们恰好是唯一能够在这些限制下运行的驱动程序。

其余驱动程序可能会受到诸如争用情况等问题的影响，这些问题可能会导致为单个请求创建大量卷，而无需明确的 " 赢家 " ；例如， Element 具有一项功能，允许卷具有相同的名称，但 ID 不同。

NetApp 已向 Docker 团队提供反馈，但没有任何迹象表明将来可以采用。

如果要配置 **FlexGroup** ，则在第二个 **FlexGroup** 具有一个或多个与要配置的 **FlexGroup** 相同的聚合时， **ONTAP** 不会配置第二个 **FlexGroup** 。