# Trident for Docker

Trident

NetApp
February 02, 2026

# Table of Contents

# Trident for Docker

## Prerequisites for deployment

You have to install and configure the necessary protocol prerequisites on your host before you can deploy Trident.

### Verify the requirements

- Verify that your deployment meets all of the requirements.
- Verify that you have a supported version of Docker installed. If your Docker version is out of date, install or update it.

  ```
  docker --version
  ```

- Verify that the protocol prerequisites are installed and configured on your host.

#### NFS tools

Install the NFS tools using the commands for your operating system.

**RHEL 8+**

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

**Ubuntu**

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

Reboot your worker nodes after installing the NFS tools to prevent failure when attaching volumes to containers.

#### iSCSI tools

Install the iSCSI tools using the commands for your operating system.

### RHEL 8+

1. Install the following system packages:

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

2. Check that iscsi-initiator-utils version is 6.2.0.874-2.el7 or later:

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

3. Set scanning to manual:

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

4. Enable multipathing:

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

> ⓘ Ensure `etc/multipath.conf` contains `find_multipaths no` under `defaults`.

5. Ensure that `iscsid` and `multipathd` are running:

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

6. Enable and start `iscsi`:

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

### Ubuntu

1. Install the following system packages:

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

2. Check that open-iscsi version is 2.0.874-5ubuntu2.10 or later (for bionic) or 2.0.874-7.1ubuntu6.1 or later (for focal):

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

3. Set scanning to manual:

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

4. Enable multipathing:

   ```
   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
   ```

   Ensure `etc/multipath.conf` contains `find_multipaths no` under `defaults`.

5. Ensure that `open-iscsi` and `multipath-tools` are enabled and running:

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

## NVMe tools

Install the NVMe tools using the commands for your operating system.

- NVMe requires RHEL 9 or later.
- If the kernel version of your Kubernetes node is too old or if the NVMe package is not available for your kernel version, you might have to update the kernel version of your node to one with the NVMe package.

**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 tools

Install the FC tools using the commands for your operating system.

- When using worker nodes that run RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) with FC PVs, specify the `discard` mountOption in the StorageClass to perform inline space reclamation. Refer to Red Hat documentation.

**RHEL 8+**

1. Install the following system packages:

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

2. Enable multipathing:

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

   Ensure `etc/multipath.conf` contains `find_multipaths no` under `defaults`.

3. Ensure that `multipathd` is running:

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

**Ubuntu**

1. Install the following system packages:

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

2. Enable multipathing:

   ```
   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
   ```

   Ensure `etc/multipath.conf` contains `find_multipaths no` under `defaults`.

3. Ensure that `multipath-tools` is enabled and running:

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

# Deploy Trident

Trident for Docker provides direct integration with the Docker ecosystem for NetApp storage platforms. It supports the provisioning and management of storage resources from the storage platform to Docker hosts, with a framework for adding additional platforms in the future.

Multiple instances of Trident can run concurrently on the same host. This allows simultaneous connections to multiple storage systems and storage types, with the ablity to customize the storage used for the Docker volumes.

**What you'll need**

See the prerequisites for deployment. After you ensure the prerequisites are met, you are ready to deploy Trident.

## Docker managed plugin method (version 1.13/17.03 and later)

> **Before you begin**
>
> If you have used Trident pre Docker 1.13/17.03 in the traditional daemon method, ensure that you stop the Trident process and restart your Docker daemon before using the managed plugin method.

1. Stop all running instances:

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

2. Restart Docker.

   ```
   systemctl restart docker
   ```

3. Ensure that you have Docker Engine 17.03 (new 1.13) or later installed.

   ```
   docker --version
   ```

   If your version is out of date, install or update your installation.

**Steps**

1. Create a configuration file and specify the options as follows:
   - `config`: The default filename is `config.json`, however you can use any name you choose by specifying the `config` option with the filename. The configuration file must be located in the `/etc/netappdvp` directory on the host system.
   - `log-level`: Specify the logging level (`debug`, `info`, `warn`, `error`, `fatal`). The default is `info`.
   - `debug`: Specify whether debug logging is enabled. Default is false. Overrides log-level if true.

a. Create a location for the configuration file:

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

b. Create the configuration file:

```
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. Start Trident using the managed plugin system. Replace `<version>` with the plugin version (xxx.xx.x) you are using.

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

3. Begin using Trident to consume storage from the configured system.

a. Create a volume named "firstVolume":

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

b. Create a default volume when the container starts:

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

c. Remove the volume "firstVolume":

```
docker volume rm firstVolume
```

## Traditional method (version 1.12 or earlier)

### Before you begin

1. Ensure that you have Docker version 1.10 or later.

   ```
   docker --version
   ```

   If your version is out of date, update your installation.

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

   Or, follow the instructions for your distribution.

2. Ensure that NFS and/or iSCSI is configured for your system.

### Steps

1. Install and configure the NetApp Docker Volume Plugin:

   a. Download and unpack the application:

   ```
   wget
   https://github.com/NetApp/trident/releases/download/10.0/trident-
   installer-25.10.0.tar.gz
   tar zxf trident-installer-25.10.0.tar.gz
   ```

   b. Move to a location in the bin path:

   ```
   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. Create a location for the configuration file:

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

   d. Create the configuration file:

   ```
   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. After placing the binary and creating the configuration file, start the Trident daemon using the desired configuration file.

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

> ⓘ Unless specified, the default name for the volume driver is "netapp".

After the daemon is started, you can create and manage volumes by using the Docker CLI interface.

3. Create a volume:

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

4. Provision a Docker volume when starting a container:

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

5. Remove a Docker volume:

```
docker volume rm trident_1
```

```
docker volume rm trident_2
```

## Start Trident at system startup

A sample unit file for systemd based systems can be found at `contrib/trident.service.example` in the Git repo. To use the file with RHEL, do the following:

1. Copy the file to the correct location.

   You should use unique names for the unit files if you have more than one instance running.

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

2. Edit the file, change the description (line 2) to match the driver name and the configuration file path (line 9) to reflect your environment.
3. Reload systemd for it to ingest changes:

   ```
   systemctl daemon-reload
   ```

4. Enable the service.

   This name varies depending on what you named the file in the `/usr/lib/systemd/system` directory.

   ```
   systemctl enable trident
   ```

5. Start the service.

   ```
   systemctl start trident
   ```

6. View the status.

   ```
   systemctl status trident
   ```

Any time you modify the unit file, run the `systemctl daemon-reload` command for it to be aware of the changes.

# Upgrade or uninstall Trident

You can safely upgrade Trident for Docker without any impact to volumes that are in use. During the upgrade process there will be a brief period where `docker volume` commands directed at the plugin will not succeed, and applications will be unable to mount volumes until the plugin is running again. Under most circumstances, this is a matter of seconds.

## Upgrade

Perform the steps below to upgrade Trident for Docker.

**Steps**

1. List the existing volumes:

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

2. Disable the plugin:

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

3. Upgrade the plugin:

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

> The 18.01 release of Trident replaces the nDVP. You should upgrade directly from the `netapp/ndvp-plugin` image to the `netapp/trident-plugin` image.

4. Enable the plugin:

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

5. Verify that the plugin is enabled:

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

6. Verify that the volumes are visible:

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

If you are upgrading from an old version of Trident (pre-20.10) to Trident 20.10 or later, you might run into an error. For more information, refer to Known Issues. If you run into the error, you should first disable the plugin, then remove the plugin, and then install the required Trident version by passing an extra config parameter: `docker plugin install netapp/trident-plugin:20.10 --alias netapp --grant-all-permissions config=config.json`

## Uninstall

Perform the steps below to uninstall Trident for Docker.

**Steps**

1. Remove any volumes that the plugin created.
2. Disable the plugin:

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

3. Remove the plugin:

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

# Work with volumes

You can easily create, clone, and remove volumes using the standard `docker volume` commands with the Trident driver name specified when needed.

## Create a volume

- Create a volume with a driver using the default name:

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

- Create a volume with a specific Trident instance:

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

If you do not specify any options, the defaults for the driver are used.

- Override the default volume size. See the following example to create a 20 GiB volume with a driver:

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

Volume sizes are expressed as strings containing an integer value with optional units (example: 10G, 20GB, 3TiB). If no units are specified, the default is G. Size units can be expressed either as powers of 2 (B, KiB, MiB, GiB, TiB) or powers of 10 (B, KB, MB, GB, TB). Shorthand units use powers of 2 (G = GiB, T = TiB, …).

## Remove a volume

- Remove the volume just like any other Docker volume:

```
docker volume rm firstVolume
```

When using the `solidfire-san` driver, the above example deletes and purges the volume.

Perform the steps below to upgrade Trident for Docker.

## Clone a volume

When using the `ontap-nas`, `ontap-san`, and `solidfire-san` storage drivers, Trident can clone volumes. When using the `ontap-nas-flexgroup` or `ontap-nas-economy` drivers, cloning is not supported. Creating a new volume from an existing volume will result in a new snapshot being created.

- Inspect the volume to enumerate snapshots:

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

- Create a new volume from an existing volume. This will result in a new snapshot being created:

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

- Create a new volume from an existing snapshot on a volume. This will not create a new snapshot:

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

### Example

```
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
```

## Access externally created volumes

You can access externally created block devices (or their clones) by containers using Trident **only** if they have no partitions and if their filesystem is supported by Trident (for example: an `ext4`-formatted `/dev/sdc1` will not be accessible via Trident).

## Driver-specific volume options

Each storage driver has a different set of options, which you can specify at volume creation time to customize the outcome. See below for options that apply to your configured storage system.

Using these options during the volume create operation is simple. Provide the option and the value using the `-o` operator during the CLI operation. These override any equivalent values from the JSON configuration file.

### ONTAP volume options

Volume create options for both NFS, iSCSI, and FC include the following:

| Option | Description |
|---|---|
| `size` | The size of the volume, defaults to 1 GiB. |
| `spaceReserve` | Thin or thick provision the volume, defaults to thin. Valid values are `none` (thin provisioned) and `volume` (thick provisioned). |
| `snapshotPolicy` | This will set the snapshot policy to the desired value. The default is `none`, meaning no snapshots will automatically be created for the volume. Unless modified by your storage administrator, a policy named "default" exists on all ONTAP systems which creates and retains six hourly, two daily, and two weekly snapshots. The data preserved in a snapshot can be recovered by browsing to the `.snapshot` directory in any directory in the volume. |
| `snapshotReserve` | This will set the snapshot reserve to the desired percentage. The default is no value, meaning ONTAP will select the snapshotReserve (usually 5%) if you have selected a snapshotPolicy, or 0% if the snapshotPolicy is none. You can set the default snapshotReserve value in the config file for all ONTAP backends, and you can use it as a volume creation option for all ONTAP backends except ontap-nas-economy. |
| `splitOnClone` | When cloning a volume, this will cause ONTAP to immediately split the clone from its parent. The default is `false`. Some use cases for cloning volumes are best served by splitting the clone from its parent immediately upon creation, because there is unlikely to be any opportunity for storage efficiencies. For example, cloning an empty database can offer large time savings but little storage savings, so it’s best to split the clone immediately. |

| Option | Description |
|---|---|
| encryption | Enable NetApp Volume Encryption (NVE) on the new volume; defaults to false. NVE must be licensed and enabled on the cluster to use this option.<br>If NAE is enabled on the backend, any volume provisioned in Trident will be NAE enabled.<br>For more information, refer to: How Trident works with NVE and NAE. |
| tieringPolicy | Sets the tiering policy to be used for the volume. This decides whether data is moved to the cloud tier when it becomes inactive (cold). |

The following additional options are for NFS **only**:

| Option | Description |
|---|---|
| unixPermissions | This controls the permission set for the volume itself. By default the permissions will be set to `---rwxr-xr-x, or in numerical notation 0755, and root will be the owner. Either the text or numerical format will work. |
| snapshotDir | Setting this to true will make the .snapshot directory visible to clients accessing the volume. The default value is false, meaning that visibility of the .snapshot directory is disabled by default. Some images, for example the official MySQL image, don't function as expected when the .snapshot directory is visible. |
| exportPolicy | Sets the export policy to be used for the volume. The default is default. |
| securityStyle | Sets the security style to be used for access to the volume. The default is unix. Valid values are unix and mixed. |

The following additional options are for iSCSI **only**:

| Option | Description |
|---|---|
| fileSystemType | Sets the file system used to format iSCSI volumes. The default is ext4. Valid values are ext3, ext4, and xfs. |

| Option | Description |
|---|---|
| `spaceAllocation` | Setting this to `false` will turn off the LUN’s space-allocation feature. The default value is `true`, meaning ONTAP notifies the host when the volume has run out of space and the LUN in the volume cannot accept writes. This option also enables ONTAP to reclaim space automatically when your host deletes data. |

#### Examples

See the examples below:

- Create a 10 GiB volume:

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

- Create a 100 GiB volume with snapshots:

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

- Create a volume which has the setUID bit enabled:

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

The minimum volume size is 20 MiB.

If the snapshot reserve is not specified and the snapshot policy is `none`, Trident use a snapshot reserve of 0%.

- Create a volume with no snapshot policy and no snapshot reserve:

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

- Create a volume with no snapshot policy and a custom snapshot reserve of 10%:

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

- Create a volume with a snapshot policy and a custom snapshot reserve of 10%:

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

- Create a volume with a snapshot policy, and accept the ONTAP’s default snapshot reserve (usually 5%):

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

### Element software volume options

The Element software options expose the size and quality of service (QoS) policies associated with the volume. When the volume is created, the QoS policy associated with it is specified using the `-o type=service_level` nomenclature.

The first step to defining a QoS service level with the Element driver is to create at least one type and specify the minimum, maximum, and burst IOPS associated with a name in the configuration file.

Other Element software volume create options include the following:

| Option | Description |
|---|---|
| `size` | The size of the volume, defaults to 1 GiB or config entry … "defaults": {"size": "5G"}. |
| `blocksize` | Use either 512 or 4096, defaults to 512 or config entry DefaultBlockSize. |

#### Example

See the following sample configuration file with QoS definitions:

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

In the above configuration, we have three policy definitions: Bronze, Silver, and Gold. These names are arbitrary.

- Create a 10 GiB Gold volume:

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

- Create a 100 GiB Bronze volume:

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

# Collect logs

You can collect logs for help with troubleshooting. The method you use to collect the logs varies based on how you are running the Docker plugin.

## Collect logs for troubleshooting

**Steps**

1. If you are running Trident using the recommended managed plugin method (i.e., using `docker plugin` commands), view them as follows:

   ```
   docker plugin ls
   ```

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

   The standard logging level should allow you to diagnose most issues. If you find that's not enough, you can enable debug logging.

2. To enable debug logging, install the plugin with debug logging enabled:

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

   Or, enable debug logging when the plugin is already installed:

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

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

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

3. If you are running the binary itself on the host, logs are available in the host's `/var/log/netappdvp` directory. To enable debug logging, specify `-debug` when you run the plugin.

## General troubleshooting tips

- The most common problem new users run into is a misconfiguration that prevents the plugin from initializing. When this happens you will likely see a message such as this when you try to install or enable the plugin:

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

This means that the plugin failed to start. Luckily, the plugin has been built with a comprehensive logging capability that should help you diagnose most of the issues you are likely to come across.

- If there are problems with mounting a PV to a container, ensure that `rpcbind` is installed and running. Use the required package manager for the host OS and check if `rpcbind` is running. You can check the status of the rpcbind service by running a `systemctl status rpcbind` or its equivalent.

# Manage multiple Trident instances

Multiple instances of Trident are needed when you desire to have multiple storage configurations available simultaneously. The key to multiple instances is to give them different names using the `--alias` option with the containerized plugin, or `--volume-driver` option when instantiating Trident on the host.

## Steps for Docker managed plugin (version 1.13/17.03 or later)

1. Launch the first instance specifying an alias and configuration file.

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

2. Launch the second instance, specifying a different alias and configuration file.

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

3. Create volumes specifying the alias as the driver name.

   For example, for gold volume:

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

   For example, for silver volume:

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

### Steps for traditional (version 1.12 or earlier)

1. Launch the plugin with an NFS configuration using a custom driver ID:

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

2. Launch the plugin with an iSCSI configuration using a custom driver ID:

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

3. Provision Docker volumes for each driver instance:

   For example, for NFS:

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

   For example, for iSCSI:

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

# Storage configuration options

See the configuration options available for your Trident configurations.

## Global configuration options

These configuration options apply to all Trident configurations, regardless of the storage platform being used.

| Option | Description | Example |
|---|---|---|
| `version` | Config file version number | `1` |
| `storageDriverName` | Name of storage driver | `ontap-nas`, `ontap-san`, `ontap-nas-economy`, `ontap-nas-flexgroup`, `solidfire-san` |
| `storagePrefix` | Optional prefix for volume names. Default: `netappdvp_`. | `staging_` |

| Option | Description | Example |
|---|---|---|
| `limitVolumeSize` | Optional restriction on volume sizes. Default: "" (not enforced) | `10g` |

Do not use `storagePrefix` (including the default) for Element backends. By default, the `solidfire-san` driver will ignore this setting and not use a prefix. NetApp recommends using either a specific tenantID for Docker volume mapping or using the attribute data which is populated with the Docker version, driver info, and raw name from Docker in cases where any name munging may have been used.

Default options are available to avoid having to specify them on every volume you create. The `size` option is available for all the controller types. See the ONTAP configuration section for an example of how to set the default volume size.

| Option | Description | Example |
|---|---|---|
| `size` | Optional default size for new volumes. Default: `1G` | `10G` |

## ONTAP configuration

In addition to the global configuration values above, when using ONTAP, the following top-level options are available.

| Option | Description | Example |
|---|---|---|
| `managementLIF` | IP address of ONTAP management LIF. You can specify a fully-qualified domain name (FQDN). | `10.0.0.1` |

| Option | Description | Example |
|---|---|---|
| `dataLIF` | IP address of protocol LIF.<br>**ONTAP NAS drivers**: NetApp recommends specifying `dataLIF`. If not provided, Trident fetches dataLIFs from the SVM. You can specify a fully-qualified domain name (FQDN) to be used for the NFS mount operations, allowing you to create a round-robin DNS to load-balance across multiple dataLIFs.<br>**ONTAP SAN drivers**: Do not specify for iSCSI or FC. Trident uses ONTAP Selective LUN Map to discover the iSCSI or FC LIFs needed to establish a multi path session. A warning is generated if `dataLIF` is explicitly defined. | `10.0.0.2` |
| `svm` | Storage virtual machine to use (required, if management LIF is a cluster LIF) | `svm_nfs` |
| `username` | Username to connect to the storage device | `vsadmin` |
| `password` | Password to connect to the storage device | `secret` |
| `aggregate` | Aggregate for provisioning (optional; if set, must be assigned to the SVM). For the `ontap-nas-flexgroup` driver, this option is ignored. All aggregates assigned to the SVM are used to provision a FlexGroup volume. | `aggr1` |
| `limitAggregateUsage` | Optional, fail provisioning if usage is above this percentage | `75%` |
| `nfsMountOptions` | Fine grained control of NFS mount options; defaults to "-o nfsvers=3". **Available only for the `ontap-nas` and `ontap-nas-economy` drivers**. See NFS host configuration information here. | `-o nfsvers=4` |

| Option | Description | Example |
|---|---|---|
| igroupName | Trident creates and manages per-node igroups as netappdvp.<br>This value cannot be changed or omitted.<br>**Available only for the ontap-san driver**. | netappdvp |
| limitVolumeSize | Maximum requestable volume size. | 300g |
| qtreesPerFlexvol | Maximum qtrees per FlexVol, must be in range [50, 300], default is 200.<br>**For the ontap-nas-economy driver, this option allows customizing the maximum number of qtrees per FlexVol**. | 300 |
| sanType | **Supported for ontap-san driver only.**<br>Use to select iscsi for iSCSI, nvme for NVMe/TCP or fcp for SCSI over Fibre Channel (FC). | iscsi if blank |
| limitVolumePoolSize | **Supported for ontap-san-economy and ontap-san-economy drivers only.**<br>Limits FlexVol sizes in ONTAP ontap-nas-economy and ontap-SAN-economy drivers. | 300g |

Default options are available to avoid having to specify them on every volume you create:

| Option | Description | Example |
|---|---|---|
| spaceReserve | Space reservation mode; none (thin provisioned) or volume (thick) | none |
| snapshotPolicy | Snapshot policy to use, default is none | none |
| snapshotReserve | Snapshot reserve percentage, default is "" to accept the ONTAP default | 10 |

| Option | Description | Example |
|---|---|---|
| `splitOnClone` | Split a clone from its parent upon creation, defaults to `false` | `false` |
| `encryption` | Enables NetApp Volume Encryption (NVE) on the new volume; defaults to `false`. NVE must be licensed and enabled on the cluster to use this option.<br>If NAE is enabled on the backend, any volume provisioned in Trident will be NAE enabled.<br>For more information, refer to: How Trident works with NVE and NAE. | true |
| `unixPermissions` | NAS option for provisioned NFS volumes, defaults to `777` | `777` |
| `snapshotDir` | NAS option for access to the `.snapshot` directory. | "true" for NFSv4<br>"false" for NFSv3 |
| `exportPolicy` | NAS option for the NFS export policy to use, defaults to `default` | `default` |
| `securityStyle` | NAS option for access to the provisioned NFS volume.<br>NFS supports `mixed` and `unix` security styles. The default is `unix`. | `unix` |
| `fileSystemType` | SAN option to select the file system type, defaults to `ext4` | `xfs` |
| `tieringPolicy` | Tiering policy to use, default is `none`. | `none` |
| `skipRecoveryQueue` | During volume deletion, bypass the recovery queue in storage and delete the volume immediately. | `` |

### Scaling options

The `ontap-nas` and `ontap-san` drivers create an ONTAP FlexVol for each Docker volume. ONTAP supports up to 1000 FlexVols per cluster node with a cluster maximum of 12,000 FlexVol volumes If your Docker volume requirements fit within that limitation, the `ontap-nas` driver is the preferred NAS solution due to the additional features offered by FlexVols, such as Docker-volume-granular snapshots and cloning.

If you need more Docker volumes than can be accommodated by the FlexVol limits, choose the `ontap-nas-economy` or the `ontap-san-economy` driver.

The `ontap-nas-economy` driver creates Docker volumes as ONTAP Qtrees within a pool of automatically managed FlexVol volumes. Qtrees offer far greater scaling, up to 100,000 per cluster node and 2,400,000 per cluster, at the expense of some features. The `ontap-nas-economy` driver does not support Docker-volume-granular snapshots or cloning.

ⓘ The `ontap-nas-economy` driver is not currently supported in Docker Swarm, because Docker Swarm does not orchestrate volume creation across multiple nodes.

The `ontap-san-economy` driver creates Docker volumes as ONTAP LUNs within a shared pool of automatically managed FlexVol volumes. This way, each FlexVol is not restricted to only one LUN and it offers better scalability for SAN workloads. Depending on the storage array, ONTAP supports up to 16384 LUNs per cluster. Because the volumes are LUNs underneath, this driver supports Docker-volume-granular snapshots and cloning.

Choose the `ontap-nas-flexgroup` driver to increase parallelism to a single volume that can grow into the petabyte range with billions of files. Some ideal use cases for FlexGroups include AI/ML/DL, big data and analytics, software builds, streaming, file repositories, and so on. Trident uses all aggregates assigned to an SVM when provisioning a FlexGroup volume. FlexGroup support in Trident also has the following considerations:

- Requires ONTAP version 9.2 or greater.
- As of this writing, FlexGroups only support NFS v3.
- Recommended to enable the 64-bit NFSv3 identifiers for the SVM.
- The minimum recommended FlexGroup member/volume size is 100 GiB.
- Cloning is not supported for FlexGroup volumes.

For information about FlexGroups and workloads that are appropriate for FlexGroups refer to the NetApp FlexGroup volume Best Practices and Implementation Guide.

To get advanced features and large scale in the same environment, you can run multiple instances of the Docker Volume Plugin, with one using `ontap-nas` and another using `ontap-nas-economy`.

### Custom ONTAP role for Trident

You can create an ONTAP cluster role with minimum privileges so that you do not have to use the ONTAP admin role to perform operations in Trident. When you include the username in a Trident backend configuration, Trident uses the ONTAP cluster role you created to perform the operations.

Refer to Trident custom-role generator for more information about creating Trident custom roles.

**Using ONTAP CLI**

1. Create a new role using the following command:

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

2. Create a usename for the Trident user:

```
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. Map the role to the user:

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

**Using System Manager**

Perform the following steps in ONTAP System Manager:

1. **Create a custom role**:
   a. To create a custom role at the cluster-level, select **Cluster > Settings**.

      (Or) To create a custom role at the SVM level, select **Storage > Storage VMs >** **`required SVM`>** **Settings > Users and Roles**.

   b. Select the arrow icon (→) next to **Users and Roles**.

   c. Select **+Add** under **Roles**.

   d. Define the rules for the role and click **Save**.

2. **Map the role to the Trident user**:
   + Perform the following steps on the **Users and Roles** page:

   a. Select Add icon **+** under **Users**.

   b. Select the required username, and select a role in the drop-down menu for **Role**.

   c. Click **Save**.

Refer to the following pages for more information:

- Custom roles for administration of ONTAP or Define custom roles
- Work with roles and users

### Example ONTAP configuration files

### NFS example for `ontap-nas` driver

```
{
  "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"
  }
}
```

### NFS example for `ontap-nas-flexgroup` driver

```
{
  "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"
  }
}
```

### NFS example for `ontap-nas-economy` driver

```
{
  "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"
}
```

### iSCSI example for `ontap-san` driver

```
{
  "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"
}
```

### NFS example for `ontap-san-economy` driver

```
{
  "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 example for `ontap-san` driver**

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

**SCSI over FC example for `ontap-san` driver**

```
{
  "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 software configuration

In addition to the global configuration values, when using Element software (NetApp HCI/SolidFire), these options are available.

| Option | Description | Example |
|---|---|---|
| `Endpoint` | https://<login>:<password>@<mvip>/json-rpc/<element-version> | https://admin:admin@192.168.160.3/json-rpc/8.0 |
| `SVIP` | iSCSI IP address and port | 10.0.0.7:3260 |
| `TenantName` | SolidFireF Tenant to use (created if not found) | `docker` |

| Option | Description | Example |
|---|---|---|
| `InitiatorIFace` | Specify interface when restricting iSCSI traffic to non-default interface | `default` |
| `Types` | QoS specifications | See example below |
| `LegacyNamePrefix` | Prefix for upgraded Trident installs. If you used a version of Trident prior to 1.3.2 and perform an upgrade with existing volumes, you’ll need to set this value to access your old volumes that were mapped via the volume-name method. | `netappdvp-` |

The `solidfire-san` driver does not support Docker Swarm.

### Example Element software configuration file

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

# Known issues and limitations

Find information about known issues and limitations when using Trident with Docker.

## Upgrading Trident Docker Volume Plugin to 20.10 and later from older versions results in upgrade failure with the no such file or directory error.

### Workaround

1. Disable the plugin.

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

2. Remove the plugin.

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

3. Reinstall the plugin by providing the extra `config` parameter.

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

## Volume names must be a minimum of 2 characters in length.

> This is a Docker client limitation. The client will interpret a single character name as being a Windows path. See bug 25773.

## Docker Swarm has certain behaviors that prevent Trident from supporting it with every storage and driver combination.

- Docker Swarm presently makes use of volume name instead of volume ID as its unique volume identifier.
- Volume requests are simultaneously sent to each node in a Swarm cluster.
- Volume plugins (including Trident) must run independently on each node in a Swarm cluster. Due to the way ONTAP works and how the `ontap-nas` and `ontap-san` drivers function, they are the only ones that happen to be able to operate within these limitations.

The rest of the drivers are subject to issues like race conditions that can result in the creation of a large number of volumes for a single request without a clear "winner"; for example, Element has a feature that allows volumes to have the same name but different IDs.

NetApp has provided feedback to the Docker team, but does not have any indication of future recourse.

## If a FlexGroup is being provisioned, ONTAP does not provision a second FlexGroup if the second FlexGroup has one or more aggregates in common with the FlexGroup being provisioned.