



# **Astra Trident for Docker**

## **Astra Trident**

NetApp  
May 03, 2022

This PDF was generated from <https://docs.netapp.com/us-en/trident/trident-docker/prereqs-docker.html> on May 03, 2022. Always check docs.netapp.com for the latest.

# Table of Contents

- Astra Trident for Docker . . . . . 1
  - Prerequisites for deployment . . . . . 1
  - Deploy Astra Trident . . . . . 3
  - Upgrade or uninstall Astra Trident . . . . . 8
  - Work with volumes . . . . . 10
  - Collect logs . . . . . 19
  - Manage multiple Astra Trident instances . . . . . 20
  - Storage configuration options . . . . . 21
  - Known issues and limitations . . . . . 36

# Astra Trident for Docker

## Prerequisites for deployment

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

- 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 If your Docker version is out of date, [install or update it](#).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:

| Protocol | Operating system | Commands  |
|----------|------------------|---|
| NFS      | RHEL/CentOS      | <code>sudo yum install -y nfs-utils</code>      |
| NFS      | Ubuntu/Debian    | <code>sudo apt-get install -y nfs-common</code> |

| Protocol | Operating system | Commands  |
|----------|------------------|---|
| iSCSI    | RHEL/CentOS      | <ol style="list-style-type: none"> <li>1. Install the following system packages:<br/><br/> <pre>sudo yum install -y lsscsi iscsi-initiator- utils sg3_utils device- mapper-multipath</pre> </li> <li>2. Start the multipathing daemon:<br/><br/> <pre>sudo mpathconf --enable --with_multipathd y</pre> </li> <li>3. Ensure that <code>iscsid</code> and <code>multipathd</code> are enabled and running:<br/><br/> <pre>sudo systemctl enable iscsid multipathd sudo systemctl start iscsid multipathd</pre> </li> <li>4. Discover the iSCSI targets:<br/><br/> <pre>sudo iscsiadm -m discoverydb -t st -p &lt;DATA_LIF_IP&gt; --discover</pre> </li> <li>5. Log in to the discovered iSCSI targets:<br/><br/> <pre>sudo iscsiadm -m node -p &lt;DATA_LIF_IP&gt; --login</pre> </li> <li>6. Enable and start <code>iscsi</code>:<br/><br/> <pre>sudo systemctl enable iscsi sudo systemctl start iscsi</pre> </li> </ol> |

| Protocol | Operating system | Commands   |
|----------|------------------|--|
| iSCSI    | Ubuntu/Debian    | <ol style="list-style-type: none"> <li>1. Install the following system packages:<br/><br/> <pre>sudo apt-get install -y open-iscsi lsscsi sg3- utils multipath-tools scsitools</pre> </li> <li>2. Enable multipathing:<br/><br/> <pre>sudo tee /etc/multipath.conf &lt; ←'EOF' defaults { user_friendly_names yes find_multipaths yes } EOF sudo service multipath- tools restart</pre> </li> <li>3. Ensure that <code>iscsid</code> and <code>multipathd</code> are running:<br/><br/> <pre>sudo service open-iscsi start sudo service multipath- tools start</pre> </li> <li>4. Discover the iSCSI targets:<br/><br/> <pre>sudo iscsiadm -m discoverydb -t st -p &lt;DATA_LIF_IP&gt; --discover</pre> </li> <li>5. Log in to the discovered iSCSI targets:<br/><br/> <pre>sudo iscsiadm -m node -p &lt;DATA_LIF_IP&gt; --login</pre> </li> </ol> |

## Deploy Astra Trident

Astra Trident for Docker provides direct integration with the Docker ecosystem for NetApp's 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 Astra Trident can run concurrently on the same host. This allows simultaneous connections to multiple storage systems and storage types, with the ability 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 Astra Trident.

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



### Before you begin

If you have used Astra Trident pre Docker 1.13/17.03 in the traditional daemon method, ensure that you stop the Astra 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": "secret",
    "aggregate": "aggr1"
}
EOF
```

2. Start Astra Trident using the managed plugin system.

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

3. Begin using Astra 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/v21.04.0/trident-
installer-21.07.0.tar.gz
tar xzf trident-installer-21.07.0.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": "secret",
    "aggregate": "aggr1"
}
EOF
```

2. After placing the binary and creating the configuration file(s), 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 Astra 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 CentOS/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 Astra Trident

You can safely upgrade Astra 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 Astra 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 Astra 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                    |
|--------------|---------------|--------------------------------|
| 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 Astra Trident (pre-20.10) to Astra Trident 20.10 or later, you might run into an error. For more information, see [Known Issues](#). If you run into the error, you should first disable the plugin, then remove the plugin, and then install the required Astra 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 Astra 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 Astra 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 Astra 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 20GiB 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 Astra Trident for Docker.

## Clone a volume

When using the `ontap-nas`, `ontap-san`, `solidfire-san`, and `gcp-cvs` storage drivers, Astra 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

```
[me@host ~]$ 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"
        }
      ]
    }
  }
]

[me@host ~]$ docker volume create -d ontap-nas --name clonedVolume -o
from=firstVolume
clonedVolume

[me@host ~]$ docker volume rm clonedVolume
[me@host ~]$ docker volume create -d ontap-nas --name volFromSnap -o
from=firstVolume -o fromSnapshot=hourly.2017-02-10_1505
volFromSnap

[me@host ~]$ 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 Astra Trident (for example: an `ext4`-formatted `/dev/sdc1` will not be accessible via Astra 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 and iSCSI 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 <code>none</code> (thin provisioned) and <code>volume</code> (thick provisioned).   |
| snapshotPolicy  | This will set the snapshot policy to the desired value. The default is <code>none</code> , 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 <code>.snapshot</code> 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 <code>ontap-nas-economy</code> .   |
| splitOnClone    | When cloning a volume, this will cause ONTAP to immediately split the clone from its parent. The default is <code>false</code> . 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.       |
| encryption      | <p>This will enable NetApp Volume Encryption (NVE) on the new volume, defaults to <code>false</code>. NVE must be licensed and enabled on the cluster to use this option.</p> <div> NetApp Aggregate Encryption (NAE) is not currently supported in Trident.</div>  |

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

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

| Option                       | Description   |
|------------------------------|---|
| <code>fileSystemType</code>  | Sets the file system used to format iSCSI volumes. The default is <code>ext4</code> . Valid values are <code>ext3</code> , <code>ext4</code> , and <code>xfs</code> .   |
| <code>spaceAllocation</code> | Setting this to <code>false</code> will turn off the LUN's space-allocation feature. The default value is <code>true</code> , 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 10GiB volume:



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

- Create a 100GiB 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 20MiB.

If the snapshot reserve is not specified and the snapshot policy is `none`, Trident will 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 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  |
|------------------------|--|
| <code>size</code>      | The size of the volume, defaults to 1GiB or config entry ... "defaults": {"size": "5G"}. |
| <code>blocksize</code> | Use either 512 or 4096, defaults to 512 or config entry <code>DefaultBlockSize</code> .  |

### 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 10GiB Gold volume:

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

- Create a 100GiB Bronze volume:

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

## CVS on GCP volume options

Volume create options for the CVS on GCP driver include the following:

| Option          | Description   |
|-----------------|---|
| size            | The size of the volume, defaults to 100 GiB for CVS-Performance volumes or 300 GiB for CVS volumes.   |
| serviceLevel    | The CVS service level of the volume, defaults to standard. Valid values are standard, premium, and extreme.                                       |
| snapshotReserve | This will set the snapshot reserve to the desired percentage. The default is no value, meaning CVS will select the snapshot reserve (usually 0%). |

### Examples

- Create a 2TiB volume:

```
docker volume create -d netapp --name demo -o size=2T
```

- Create a 5TiB premium volume:

```
docker volume create -d netapp --name demo -o size=5T -o  
serviceLevel=premium
```

The minimum volume size is 100 GiB for CVS-Performance volumes, or 300 GiB for CVS volumes.

## Azure NetApp Files volume options

Volume create options for the Azure NetApp Files driver include the following:

| Option | Description                                 |
|--------|---|
| size   | The size of the volume, defaults to 100 GB. |

### Examples

- Create a 200GiB volume:

```
docker volume create -d netapp --name demo -o size=200G
```

The minimum volume size is 100 GB.

# 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.

## Steps

1. If you are running Astra 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 Astra 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 Astra Trident configurations.

### Global configuration options

These configuration options apply to all Astra 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, azure-netapp-files, or gcp-cvs |
| storagePrefix     | Optional prefix for volume names. Default: "netappdvp_".         | staging_  |
| 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. We recommend 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 |
| dataLIF             | IP address of protocol LIF; will be derived if not specified. For the <code>ontap-nas</code> drivers <b>only</b> , you can specify an FQDN, in which case the FQDN will be used for the NFS mount operations. For the <code>ontap-san</code> drivers, the default is to use all data LIF IPs from the SVM and to use iSCSI multipath. Specifying an IP address for <code>dataLIF</code> for the <code>ontap-san</code> drivers forces the driver to disable multipath and use only the specified address. | 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 <code>ontap-nas-flexgroup</code> 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%      |



| Option           | Description   | Example      |
|------------------|---|--------------|
| nfsMountOptions  | Fine grained control of NFS mount options; defaults to “-o nfsvers=3”. <b>Available only for the <code>ontap-nas</code> and <code>ontap-nas-economy</code> drivers.</b> See <a href="#">NFS host configuration information here</a> . | -o nfsvers=4 |
| igroupName       | The igroup used by the plugin; defaults to “netappdvp”. <b>Available only for the <code>ontap-san</code> driver.</b>  | myigroup     |
| limitVolumeSize  | Maximum requestable volume size and qtree parent volume size. <b>For the <code>ontap-nas-economy</code> driver, this option additionally limits the size of the FlexVols that it creates.</b>   | 300g         |
| qtreesPerFlexvol | Maximum qtrees per FlexVol, must be in range [50, 300], default is 200. <b>For the <code>ontap-nas-economy</code> driver, this option allows customizing the maximum number of qtrees per FlexVol.</b>                                | 300          |

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 ONTAP’s default  | 10      |
| splitOnClone    | Split a clone from its parent upon creation, defaults to “false”      | false   |
| encryption      | Enable NetApp Volume Encryption, defaults to “false”                  | true    |

| Option                       | Description  | Example |
|------------------------------|--|---------|
| <code>unixPermissions</code> | NAS option for provisioned NFS volumes, defaults to “777”  | 777     |
| <code>snapshotDir</code>     | NAS option for access to the <code>.snapshot</code> directory, defaults to “false”               | true    |
| <code>exportPolicy</code>    | NAS option for the NFS export policy to use, defaults to “default”                               | default |
| <code>securityStyle</code>   | NAS option for access to the provisioned NFS volume, defaults to “unix”                          | mixed   |
| <code>fileSystemType</code>  | SAN option to select the file system type, defaults to “ext4”                                    | xfs     |
| <code>tieringPolicy</code>   | Tiering policy to use, default is “none”; “snapshot-only” for pre-ONTAP 9.5 SVM-DR configuration | none    |

## 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 FlexVols. 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 FlexVols. 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 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 FlexVols. 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 size is 100GB.
- Cloning is not supported for FlexGroup Volumes.

For information about FlexGroups and workloads that are appropriate for FlexGroups see the [NetApp FlexGroup Volume Best Practices and Implementation Guide](#).

To get advanced features and huge 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`.

### 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": "secret",
  "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": "secret",
  "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": "secret",
  "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": "secret",
  "aggregate": "aggr1",
  "igroupName": "myigroup"
}
```

#### **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": "secret",
  "aggregate": "aggr1",
  "igroupName": "myigroup"
}
```

## 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         | <code>&lt;a href="https://&amp;lt;login&amp;gt;:&amp;lt;password&amp;gt;@&amp;lt;mvip&amp;gt;/json-rpc/&amp;lt;element-version&amp;gt;" class="bare"&gt;https://&amp;lt;login&amp;gt;:&amp;lt;password&amp;gt;@&amp;lt;mvip&amp;gt;/json-rpc/&amp;lt;element-version&amp;gt;&lt;/a&gt;</code> | <a href="https://admin:admin@192.168.160.3/json-rpc/8.0">https://admin:admin@192.168.160.3/json-rpc/8.0</a> |
| SVIP             | iSCSI IP address and port   | 10.0.0.7:3260   |
| TenantName       | SolidFireF Tenant to use (created if not found)   | "docker"  |
| 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
      }
    }
  ]
}
```

## Cloud Volumes Service (CVS) on GCP configuration

Trident now includes support for smaller volumes with the default CVS service type on [GCP](#). For backends created with `storageClass=software`, volumes will now have a minimum provisioning size of 300 GiB. **NetApp recommends customers consume sub-1TiB volumes for non-production workloads.** CVS currently provides this feature under Controlled Availability and does not provide technical support.



Sign up for access to sub-1TiB volumes [here](#).



When deploying backends using the default CVS service type `storageClass=software`, you should obtain access to the sub-1TiB volumes feature on GCP for the Project Number(s) and Project ID(s) in question. This is necessary for Trident to provision sub-1TiB volumes. If not, volume creations **will fail** for PVCs that are <600 GiB. Obtain access to sub-1TiB volumes using [this form](#).

Volumes created by Trident for the default CVS service level will be provisioned as follows:

- PVCs that are smaller than 300 GiB will result in Trident creating a 300 GiB CVS volume.
- PVCs that are between 300 GiB to 600 GiB will result in Trident creating a CVS volume of the requested size.
- PVCs that are between 600 GiB and 1 TiB will result in Trident creating a 1TiB CVS volume.
- PVCs that are greater than 1 TiB will result in Trident creating a CVS volume of the requested size.

In addition to the global configuration values, when using CVS on GCP, these options are available.

| Option                         | Description  | Example                            |
|--------------------------------|--|------------------------------------|
| <code>apiRegion</code>         | CVS account region (required). Is the GCP region where this backend will provision volumes.  | "us-west2"                         |
| <code>projectNumber</code>     | GCP project number (required). Can be found in the GCP web portal's Home screen.   | "123456789012"                     |
| <code>hostProjectNumber</code> | GCP shared VPC host project number (required if using a shared VPC)  | "098765432109"                     |
| <code>apiKey</code>            | API key for GCP service account with CVS admin role (required). Is the JSON-formatted contents of a GCP service account's private key file (copied verbatim into the backend config file). The service account must have the <code>netappcloudvolumes.admin</code> role. | (contents of the private key file) |
| <code>secretKey</code>         | CVS account secret key (required). Can be found in the CVS web portal in Account settings > API access.  | "default"                          |

| Option          | Description  | Example                         |
|-----------------|--|---------------------------------|
| proxyURL        | Proxy URL if proxy server required to connect to the CVS account. The proxy server can either be an HTTP proxy or an HTTPS proxy. In case of an HTTPS proxy, certificate validation is skipped to allow the usage of self-signed certificates in the proxy server. <b>Proxy servers with authentication enabled are not supported.</b> | "http://proxy-server-hostname/" |
| nfsMountOptions | NFS mount options; defaults to "-o nfsvers=3"  | "nfsvers=3,proto=tcp,timeo=600" |
| serviceLevel    | Performance level (standard, premium, extreme), defaults to "standard"   | "premium"                       |
| network         | GCP network used for CVS volumes, defaults to "default"  | "default"                       |



If using a shared VPC network, you should specify both `projectNumber` and `hostProjectNumber`. In that case, `projectNumber` is the service project and `hostProjectNumber` is the host project.



The NetApp Cloud Volumes Service for GCP does not support CVS-Performance volumes less than 100 GiB in size, or CVS volumes less than 300 GiB in size. To make it easier to deploy applications, Trident automatically creates volumes of the minimum size if a too-small volume is requested.

When using CVS on GCP, these default volume option settings are available.

| Option          | Description   | Example                  |
|-----------------|---|--------------------------|
| exportRule      | NFS access list (addresses and/or CIDR subnets), defaults to "0.0.0.0/0"  | "10.0.1.0/24,10.0.2.100" |
| snapshotDir     | Controls visibility of the <code>.snapshot</code> directory               | "false"                  |
| snapshotReserve | Snapshot reserve percentage, default is "" to accept the CVS default of 0 | "10"                     |
| size            | Volume size, defaults to "100GiB"   | "10T"                    |



## Example CVS on GCP configuration file

```
{
  "version": 1,
  "storageDriverName": "gcp-cvs",
  "projectNumber": "012345678901",
  "apiRegion": "us-west2",
  "apiKey": {
    "type": "service_account",
    "project_id": "my-gcp-project",
    "private_key_id": "1234567890123456789012345678901234567890",
    "private_key": "-----BEGIN PRIVATE KEY-----
\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZ
srtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisI
sAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSa
PIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZN
chRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzll
ZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl
/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kw
s8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY
9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHc
zZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHi
sIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOgu
SaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyA
ZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz
llZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3
bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4
Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5o
jY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nzn
HczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtr
HisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbO
guSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKe
yAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRA
GzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4j
K3bl/qp8B4Kws8zX5ojY9m\nXsYg6gyxy4zq7OlwWgLwGa==\n-----END PRIVATE
KEY-----\n",
    "client_email": "cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com",
    "client_id": "123456789012345678901",
    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
    "token_uri": "https://oauth2.googleapis.com/token",
    "auth_provider_x509_cert_url":
"https://www.googleapis.com/oauth2/v1/certs",
    "client_x509_cert_url":
"https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com"
```

```

    },
    "proxyURL": "http://proxy-server-hostname/"
  }

```

## Azure NetApp Files configuration

To configure and use an [Azure NetApp Files](#) backend, you will need the following:

- subscriptionID from an Azure subscription with Azure NetApp Files enabled
- tenantID, clientID, and clientSecret from an [App Registration](#) in Azure Active Directory with sufficient permissions to the Azure NetApp Files service
- Azure location that contains at least one [delegated subnet](#)



If you're using Azure NetApp Files for the first time or in a new location, some initial configuration is required that the [quickstart guide](#) will walk you through.



Astra Trident 21.04.0 and earlier do not support Manual QoS capacity pools.

| Option            | Description   | Default                               |
|-------------------|---|---------------------------------------|
| version           | Always 1  |                                       |
| storageDriverName | "azure-netapp-files"                                      |                                       |
| backendName       | Custom name for the storage backend                       | Driver name + "_" + random characters |
| subscriptionID    | The subscription ID from your Azure subscription          |                                       |
| tenantID          | The tenant ID from an App Registration                    |                                       |
| clientID          | The client ID from an App Registration                    |                                       |
| clientSecret      | The client secret from an App Registration                |                                       |
| serviceLevel      | One of "Standard", "Premium" or "Ultra"                   | "" (random)                           |
| location          | Name of the Azure location new volumes will be created in | "" (random)                           |

| Option          | Description  | Default                      |
|-----------------|--|------------------------------|
| virtualNetwork  | Name of a virtual network with a delegated subnet              | "" (random)                  |
| subnet          | Name of a subnet delegated to Microsoft.Netapp/volumes         | "" (random)                  |
| nfsMountOptions | Fine-grained control of NFS mount options                      | "-o nfsvers=3"               |
| limitVolumeSize | Fail provisioning if requested volume size is above this value | "" (not enforced by default) |



The Azure NetApp Files service does not support volumes less than 100 GB in size. To make it easier to deploy applications, Trident automatically creates 100 GB volumes if a smaller volume is requested.

You can control how each volume is provisioned by default using these options in a special section of the configuration.

| Option      | Description   | Default     |
|-------------|---|-------------|
| exportRule  | The export rule(s) for new volumes. Must be a comma-separated list of any combination of IPv4 addresses or IPv4 subnets in CIDR notation. | "0.0.0.0/0" |
| snapshotDir | Controls visibility of the .snapshot directory  | "false"     |
| size        | The default size of new volumes   | "100G"      |

## Example Azure NetApp Files configurations

### Example 1: Minimal backend configuration for azure-netapp-files

This is the absolute minimum backend configuration. With this configuration, Trident will discover all of your NetApp accounts, capacity pools, and subnets delegated to ANF in every location worldwide, and place new volumes on one of them randomly.

This configuration is useful when you're just getting started with ANF and trying things out, but in practice you're going to want to provide additional scoping for the volumes you provision to make sure that they have the characteristics you want and end up on a network that's close to the compute that's using it. See the subsequent examples for more details.

```
{
  "version": 1,
  "storageDriverName": "azure-netapp-files",
  "subscriptionID": "9f87c765-4774-fake-ae98-a721add45451",
  "tenantID": "68e4f836-edc1-fake-bff9-b2d865ee56cf",
  "clientID": "dd043f63-bf8e-fake-8076-8de91e5713aa",
  "clientSecret": "SECRET"
}
```

### Example 2: Single location and specific service level for azure-netapp-files

This backend configuration places volumes in Azure’s “eastus” location in a “Premium” capacity pool. Trident automatically discovers all of the subnets delegated to ANF in that location and will place a new volume on one of them randomly.

```
{
  "version": 1,
  "storageDriverName": "azure-netapp-files",
  "subscriptionID": "9f87c765-4774-fake-ae98-a721add45451",
  "tenantID": "68e4f836-edc1-fake-bff9-b2d865ee56cf",
  "clientID": "dd043f63-bf8e-fake-8076-8de91e5713aa",
  "clientSecret": "SECRET",
  "location": "eastus",
  "serviceLevel": "Premium"
}
```

### Example 3: Advanced configuration for azure-netapp-files

This backend configuration further reduces the scope of volume placement to a single subnet, and also modifies some volume provisioning defaults.

```
{
  "version": 1,
  "storageDriverName": "azure-netapp-files",
  "subscriptionID": "9f87c765-4774-fake-ae98-a721add45451",
  "tenantID": "68e4f836-edc1-fake-bff9-b2d865ee56cf",
  "clientID": "dd043f63-bf8e-fake-8076-8de91e5713aa",
  "clientSecret": "SECRET",
  "location": "eastus",
  "serviceLevel": "Premium",
  "virtualNetwork": "my-virtual-network",
  "subnet": "my-subnet",
  "nfsMountOptions": "nfsvers=3,proto=tcp,timeo=600",
  "limitVolumeSize": "500Gi",
  "defaults": {
    "exportRule": "10.0.0.0/24,10.0.1.0/24,10.0.2.100",
    "size": "200Gi"
  }
}
```

#### Example 4: Virtual storage pools with azure-netapp-files

This backend configuration defines multiple [pools of storage](#) in a single file. This is useful when you have multiple capacity pools supporting different service levels and you want to create storage classes in Kubernetes that represent those.

This is just scratching the surface of the power of virtual storage pools and their labels.

```
{
  "version": 1,
  "storageDriverName": "azure-netapp-files",
  "subscriptionID": "9f87c765-4774-fake-ae98-a721add45451",
  "tenantID": "68e4f836-edc1-fake-bff9-b2d865ee56cf",
  "clientID": "dd043f63-bf8e-fake-8076-8de91e5713aa",
  "clientSecret": "SECRET",
  "nfsMountOptions": "nfsvers=3,proto=tcp,timeo=600",
  "labels": {
    "cloud": "azure"
  },
  "location": "eastus",

  "storage": [
    {
      "labels": {
        "performance": "gold"
      },
      "serviceLevel": "Ultra"
    },
    {
      "labels": {
        "performance": "silver"
      },
      "serviceLevel": "Premium"
    },
    {
      "labels": {
        "performance": "bronze"
      },
      "serviceLevel": "Standard",
    }
  ]
}
```

## Known issues and limitations

Find information about known issues and limitations when using Astra 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 Astra 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 Astra 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.**

## Copyright Information

Copyright © 2022 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system-without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## Trademark Information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.