



ONTAP SAN drivers

Trident

NetApp
October 30, 2026

Table of Contents

- ONTAP SAN drivers 1
 - ONTAP SAN driver overview 1
 - ONTAP SAN driver details 1
 - User permissions 2
 - Additional considerations for NVMe/TCP 2
 - Prepare to configure backend with ONTAP SAN drivers 2
 - Requirements 3
 - Authenticate the ONTAP backend 3
 - Authenticate connections with bidirectional CHAP 8
- ONTAP SAN configuration options and examples 10
 - Backend configuration options 10
 - Backend configuration options for provisioning volumes 15
 - Minimal configuration examples 17
 - Examples of backends with virtual pools 21
 - Map backends to StorageClasses 24

ONTAP SAN drivers

ONTAP SAN driver overview

Learn about configuring an ONTAP backend with ONTAP and Cloud Volumes ONTAP SAN drivers.

ONTAP SAN driver details

Trident provides the following SAN storage drivers to communicate with the ONTAP cluster. Supported access modes are: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Driver	Protocol	volumeMode	Access modes supported	File systems supported
ontap-san	iSCSI SCSI over FC	Block	RWO, ROX, RWX, RWOP	No filesystem; raw block device
ontap-san	iSCSI SCSI over FC	Filesystem	RWO, RWOP ROX and RWX are not available in Filesystem volume mode.	xfs, ext3, ext4
ontap-san	NVMe/TCP Refer to Additional considerations for NVMe/TCP .	Block	RWO, ROX, RWX, RWOP	No filesystem; raw block device
ontap-san	NVMe/TCP Refer to Additional considerations for NVMe/TCP .	Filesystem	RWO, RWOP ROX and RWX are not available in Filesystem volume mode.	xfs, ext3, ext4
ontap-san-economy	iSCSI	Block	RWO, ROX, RWX, RWOP	No filesystem; raw block device
ontap-san-economy	iSCSI	Filesystem	RWO, RWOP ROX and RWX are not available in Filesystem volume mode.	xfs, ext3, ext4



- Use `ontap-san-economy` only if persistent volume usage count is expected to be higher than [supported ONTAP volume limits](#).
- Use `ontap-nas-economy` only if persistent volume usage count is expected to be higher

than [supported ONTAP volume limits](#) and the `ontap-san-economy` driver cannot be used.

- Do not use `ontap-nas-economy` if you anticipate the need for data protection, disaster recovery, or mobility.
- NetApp does not recommend using Flexvol autogrow in all ONTAP drivers, except `ontap-san`. As a workaround, Trident supports the use of snapshot reserve and scales Flexvol volumes accordingly.

User permissions

Trident expects to be run as either an ONTAP or SVM administrator, typically using the `admin` cluster user or a `vsadmin` SVM user, or a user with a different name that has the same role. For Amazon FSx for NetApp ONTAP deployments, Trident expects to be run as either an ONTAP or SVM administrator, using the cluster `fsxadmin` user or a `vsadmin` SVM user, or a user with a different name that has the same role. The `fsxadmin` user is a limited replacement for the cluster admin user.



If you use the `limitAggregateUsage` parameter, cluster admin permissions are required. When using Amazon FSx for NetApp ONTAP with Trident, the `limitAggregateUsage` parameter will not work with the `vsadmin` and `fsxadmin` user accounts. The configuration operation will fail if you specify this parameter.

While it is possible to create a more restrictive role within ONTAP that a Trident driver can use, we don't recommend it. Most new releases of Trident will call additional APIs that would have to be accounted for, making upgrades difficult and error-prone.

Additional considerations for NVMe/TCP

Trident supports the non-volatile memory express (NVMe) protocol using the `ontap-san` driver including:

- IPv6
- Snapshots and clones of NVMe volumes
- Resizing an NVMe volume
- Importing an NVMe volume that was created outside of Trident so that its lifecycle can be managed by Trident
- NVMe-native multipathing
- Graceful or ungraceful shutdown of the K8s nodes (24.06)

Trident does not support:

- DH-HMAC-CHAP that is natively supported by NVMe
- Device mapper (DM) multipathing
- LUKS encryption



NVMe is supported only with ONTAP REST APIs and not supported with ONTAPI (ZAPI).

Prepare to configure backend with ONTAP SAN drivers

Understand the requirements and authentication options for configuring an ONTAP

backend with ONTAP SAN drivers.

Requirements

For all ONTAP backends, Trident requires at least one aggregate be assigned to the SVM.



[ASA r2 systems](#) differ from other ONTAP systems (ASA, AFF, and FAS) in the implementation of their storage layer. In ASA r2 systems, storage availability zones are used instead of aggregates. Refer to [this](#) Knowledge Base article on how to assign aggregates to SVMs in ASA r2 systems.

Remember that you can also run more than one driver, and create storage classes that point to one or the other. For example, you could configure a `san-dev` class that uses the `ontap-san` driver and a `san-default` class that uses the `ontap-san-economy` one.

All your Kubernetes worker nodes must have the appropriate iSCSI tools installed. Refer to [Prepare the worker node](#) for details.

Authenticate the ONTAP backend

Trident offers two modes of authenticating an ONTAP backend.

- **Credential-based:** The username and password to an ONTAP user with the required permissions. It is recommended to use a pre-defined security login role, such as `admin` or `vsadmin` to ensure maximum compatibility with ONTAP versions.
- **Certificate-based:** Trident can also communicate with an ONTAP cluster using a certificate installed on the backend. Here, the backend definition must contain Base64-encoded values of the client certificate, key, and the trusted CA certificate if used (recommended).

You can update existing backends to move between credential-based and certificate-based methods. However, only one authentication method is supported at a time. To switch to a different authentication method, you must remove the existing method from the backend configuration.



If you attempt to provide **both credentials and certificates**, backend creation will fail with an error that more than one authentication method was provided in the configuration file.

Enable credential-based authentication

Trident requires the credentials to an SVM-scoped/cluster-scoped admin to communicate with the ONTAP backend. It is recommended to make use of standard, pre-defined roles such as `admin` or `vsadmin`. This ensures forward compatibility with future ONTAP releases that might expose feature APIs to be used by future Trident releases. A custom security login role can be created and used with Trident, but is not recommended.

A sample backend definition will look like this:

YAML

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-san
```

```
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: password
```

JSON

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password"
}
```

Keep in mind that the backend definition is the only place the credentials are stored in plain text. After the backend is created, usernames/passwords are encoded with Base64 and stored as Kubernetes secrets. The creation or update of a backend is the only step that requires knowledge of the credentials. As such, it is an admin-only operation, to be performed by the Kubernetes/storage administrator.

Enable certificate-based authentication

New and existing backends can use a certificate and communicate with the ONTAP backend. Three parameters are required in the backend definition.

- `clientCertificate`: Base64-encoded value of client certificate.
- `clientPrivateKey`: Base64-encoded value of associated private key.
- `trustedCACertificate`: Base64-encoded value of trusted CA certificate. If using a trusted CA, this parameter must be provided. This can be ignored if no trusted CA is used.

A typical workflow involves the following steps.

Steps

1. Generate a client certificate and key. When generating, set Common Name (CN) to the ONTAP user to authenticate as.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=admin"
```

2. Add trusted CA certificate to the ONTAP cluster. This might be already handled by the storage administrator. Ignore if no trusted CA is used.

```
security certificate install -type server -cert-name <trusted-ca-cert-
```

```
name> -vserver <vserver-name>
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled
true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca
<cert-authority>
```

3. Install the client certificate and key (from step 1) on the ONTAP cluster.

```
security certificate install -type client-ca -cert-name <certificate-
name> -vserver <vserver-name>
security ssl modify -vserver <vserver-name> -client-enabled true
```



After running this command, ONTAP prompts for certificate input. Paste the contents of the `k8senv.pem` file generated in step 1, then enter `END` to complete the installation.

4. Confirm the ONTAP security login role supports `cert` authentication method.

```
security login create -user-or-group-name admin -application ontapi
-authentication-method cert
security login create -user-or-group-name admin -application http
-authentication-method cert
```

5. Test authentication using certificate generated. Replace `<ONTAP Management LIF>` and `<vserver name>` with Management LIF IP and SVM name.

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns="http://www.netapp.com/filer/admin" version="1.21"
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. Encode certificate, key and trusted CA certificate with Base64.

```
base64 -w 0 k8senv.pem >> cert_base64
base64 -w 0 k8senv.key >> key_base64
base64 -w 0 trustedca.pem >> trustedca_base64
```

7. Create backend using the values obtained from the previous step.

```
cat cert-backend.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
```

```

"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"svm": "vserver_test",
"clientCertificate": "Faaaakkkkeeee...Vaaalllluuuuueeee",
"clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
"trustedCACertificate": "QNFinfO...SiqOyN",
"storagePrefix": "myPrefix_"
}

```

```
tridentctl create backend -f cert-backend.json -n trident
```

```

+-----+-----+-----+-----+
+-----+-----+
|      NAME      | STORAGE DRIVER |                      UUID                      |
STATE | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |         0 |
+-----+-----+-----+-----+
+-----+-----+

```

Update authentication methods or rotate credentials

You can update an existing backend to use a different authentication method or to rotate their credentials. This works both ways: backends that make use of username/password can be updated to use certificates; backends that utilize certificates can be updated to username/password based. To do this, you must remove the existing authentication method and add the new authentication method. Then use the updated backend.json file containing the required parameters to execute `tridentctl backend update`.

```

cat cert-backend-updated.json
{
"version": 1,
"storageDriverName": "ontap-san",
"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"svm": "vserver_test",
"username": "vsadmin",
"password": "password",
"storagePrefix": "myPrefix_"
}

```

```
#Update backend with tridentctl
```

```
tridentctl update backend SanBackend -f cert-backend-updated.json -n
trident
```

```

+-----+-----+-----+-----+
+-----+-----+

```

NAME	STORAGE DRIVER	UUID
SanBackend	ontap-san	586b1cd5-8cf8-428d-a76c-2872713612c1
online	9	



When rotating passwords, the storage administrator must first update the password for the user on ONTAP. This is followed by a backend update. When rotating certificates, multiple certificates can be added to the user. The backend is then updated to use the new certificate, following which the old certificate can be deleted from the ONTAP cluster.

Updating a backend does not disrupt access to volumes that have already been created, nor impact volume connections made after. A successful backend update indicates that Trident can communicate with the ONTAP backend and handle future volume operations.

Create 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 username 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"
```

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](#)

Authenticate connections with bidirectional CHAP

Trident can authenticate iSCSI sessions with bidirectional CHAP for the `ontap-san` and `ontap-san-economy` drivers. This requires enabling the `useCHAP` option in your backend definition. When set to `true`, Trident configures the SVM's default initiator security to bidirectional CHAP and set the username and secrets from the backend file. NetApp recommends using bidirectional CHAP to authenticate connections. See the following sample configuration:

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap_san_chap
managementLIF: 192.168.0.135
svm: ontap_iscsi_svm
useCHAP: true
username: vsadmin
password: password
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
```



The `useCHAP` parameter is a Boolean option that can be configured only once. It is set to `false` by default. After you set it to `true`, you cannot set it to `false`.

In addition to `useCHAP=true`, the `chapInitiatorSecret`, `chapTargetInitiatorSecret`, `chapTargetUsername`, and `chapUsername` fields must be included in the backend definition. The secrets can be changed after a backend is created by running `tridentctl update`.

How it works

By setting `useCHAP` to `true`, the storage administrator instructs Trident to configure CHAP on the storage backend. This includes the following:

- Setting up CHAP on the SVM:
 - If the SVM's default initiator security type is `none` (set by default) **and** there are no pre-existing LUNs already present in the volume, Trident will set the default security type to `CHAP` and proceed to configuring the CHAP initiator and target username and secrets.
 - If the SVM contains LUNs, Trident will not enable CHAP on the SVM. This ensures that access to LUNs that are already present on the SVM isn't restricted.
- Configuring the CHAP initiator and target username and secrets; these options must be specified in the backend configuration (as shown above).

After the backend is created, Trident creates a corresponding `tridentbackend` CRD and stores the CHAP secrets and usernames as Kubernetes secrets. All PVs that are created by Trident on this backend will be mounted and attached over CHAP.

Rotate credentials and update backends

You can update the CHAP credentials by updating the CHAP parameters in the `backend.json` file. This will require updating the CHAP secrets and using the `tridentctl update` command to reflect these changes.



When updating the CHAP secrets for a backend, you must use `tridentctl` to update the backend. Do not update the credentials on the storage cluster using the ONTAP CLI or ONTAP System Manager as Trident will not be able to pick up these changes.

```
cat backend-san.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "ontap_san_chap",
  "managementLIF": "192.168.0.135",
  "svm": "ontap_iscsi_svm",
  "useCHAP": true,
  "username": "vsadmin",
  "password": "password",
  "chapInitiatorSecret": "cl9qxUpDaTeD",
  "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
}

./tridentctl update backend ontap_san_chap -f backend-san.json -n trident
+-----+-----+-----+-----+
+-----+-----+
|  NAME           | STORAGE DRIVER |                               UUID                               |
STATE | VOLUMES |
```

```

+-----+-----+-----+
+-----+-----+
| ontap_san_chap | ontap-san      | aa458f3b-ad2d-4378-8a33-1a472ffbeb5c |
online |          7 |
+-----+-----+-----+
+-----+-----+

```

Existing connections will remain unaffected; they will continue to remain active if the credentials are updated by Trident on the SVM. New connections use the updated credentials and existing connections continue to remain active. Disconnecting and reconnecting old PVs will result in them using the updated credentials.

ONTAP SAN configuration options and examples

Create and use ONTAP SAN drivers with your Trident installation. This section provides backend configuration examples and details for mapping backends to StorageClasses. [ASA r2 systems](#) differ from other ONTAP systems (ASA, AFF, and FAS) in the implementation of their storage layer. These variations impact the usage of certain parameters as notated. [Learn more about the differences between ASA r2 systems and other ONTAP systems](#). In the Trident backend configuration, you need not specify that your system is ASA r2. When you select `ontap-san` as the `storageDriverName`, Trident detects automatically the ASA r2 or other ONTAP systems. Some backend configuration parameters are not applicable to ASA r2 systems as noted in the table below.



Only the `ontap-san` driver (with iSCSI, NVMe/TCP, and FC protocols) is supported for ASA r2 systems.

Backend configuration options

Beginning with the 26.06 release, you can enable ONTAP balanced placement so that ONTAP selects the aggregate for new volumes. Refer to [Use balanced placement with ONTAP backends](#) for requirements and examples.

See the following table for the backend configuration options:

Parameter	Description	Default
<code>version</code>		Always 1
<code>storageDriverName</code>	Name of the storage driver. Cannot be modified after creation. To update this parameter, you must create a new backend.	<code>ontap-san</code> or <code>ontap-san-economy</code>
<code>backendName</code>	Custom name or the storage backend	Driver name + "_" + dataLIF

Parameter	Description	Default
managementLIF	<p>IP address of a cluster or SVM management LIF.</p> <p>A fully-qualified domain name (FQDN) can be specified.</p> <p>Can be set to use IPv6 addresses if Trident was installed using the IPv6 flag. IPv6 addresses must be defined in square brackets, such as [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].</p> <p>For seamless MetroCluster switchover, see the MetroCluster example.</p> <div>  <p>If you are using "vsadmin" credentials, managementLIF must be that of the SVM; if using "admin" credentials, managementLIF must be that of the cluster.</p> </div>	"10.0.0.1", "[2001:1234:abcd::fefe]"
dataLIF	<p>IP address of protocol LIF.</p> <p>Can be set to use IPv6 addresses if Trident was installed using the IPv6 flag. IPv6 addresses must be defined in square brackets, such as [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].</p> <p>Do not specify for iSCSI. Trident uses ONTAP Selective LUN Map to discover the iSCSI LIFs needed to establish a multi path session. A warning is generated if dataLIF is explicitly defined.</p> <p>Omit for Metrocluster. See the MetroCluster example.</p>	Derived by the SVM
svm	<p>Storage virtual machine to use</p> <p>Omit for Metrocluster. See the MetroCluster example.</p>	Derived if an SVM managementLIF is specified
useCHAP	<p>Use CHAP to authenticate iSCSI for ONTAP SAN drivers [Boolean].</p> <p>Set to true for Trident to configure and use bidirectional CHAP as the default authentication for the SVM given in the backend. Refer to Prepare to configure backend with ONTAP SAN drivers for details.</p> <p>Not supported for FCP or NVMe/TCP.</p>	false

Parameter	Description	Default
chapInitiatorSecret	CHAP initiator secret. Required if useCHAP=true	""
labels	Set of arbitrary JSON-formatted labels to apply on volumes	""
chapTargetInitiatorSecret	CHAP target initiator secret. Required if useCHAP=true	""
chapUsername	Inbound username. Required if useCHAP=true	""
chapTargetUsername	Target username. Required if useCHAP=true	""
clientCertificate	Base64-encoded value of client certificate. Used for certificate-based auth	""
clientPrivateKey	Base64-encoded value of client private key. Used for certificate-based auth	""
trustedCACertificate	Base64-encoded value of trusted CA certificate. Optional. Used for certificate-based authentication.	""
username	Username needed to communicate with the ONTAP cluster. Used for credential-based authentication. For Active Directory authentication, see Authenticate Trident to a backend SVM using Active Directory credentials .	""
password	Password needed to communicate with the ONTAP cluster. Used for credential-based authentication. For Active Directory authentication, see Authenticate Trident to a backend SVM using Active Directory credentials .	""
svm	Storage virtual machine to use	Derived if an SVM managementLIF is specified
storagePrefix	Prefix used when provisioning new volumes in the SVM. Cannot be modified later. To update this parameter, you need to create a new backend.	trident

Parameter	Description	Default
aggregate	<p>Aggregate for provisioning (optional; if set, must be assigned to the SVM). For the <code>ontap-nas-flexgroup</code> driver, this option is ignored. If not assigned, any of the available aggregates can be used to provision a FlexGroup volume.</p> <div>  <p>When the aggregate is updated in SVM, it is updated in Trident automatically by polling SVM without having to restart the Trident Controller. When you have configured a specific aggregate in Trident to provision volumes, if the aggregate is renamed or moved out of the SVM, the backend moves to failed state in Trident while polling the SVM aggregate. You must either change the aggregate to one that is present on the SVM or remove it altogether to bring the backend back online.</p> </div> <p>Do not specify for ASA r2 systems.</p> <p>Do not specify when <code>useBalancedPlacement</code> is set to <code>true</code>.</p>	""
useBalancedPlacement	<p>Delegate aggregate selection for new volumes to ONTAP [Boolean]. Requires ONTAP 9.19.1 or later and <code>useREST</code> set to <code>true</code>. Cannot be combined with <code>aggregate</code>.</p> <p>Refer to Use balanced placement with ONTAP backends for requirements and examples.</p>	false
limitAggregateUsage	<p>Fail provisioning if usage is above this percentage.</p> <p>If you are using an Amazon FSx for NetApp ONTAP backend, do not specify <code>limitAggregateUsage</code>. The provided <code>fsxadmin</code> and <code>vsadmin</code> do not contain the permissions required to retrieve aggregate usage and limit it using Trident.</p> <p>Do not specify for ASA r2 systems.</p>	"" (not enforced by default)
limitVolumeSize	<p>Fail provisioning if requested volume size is above this value.</p> <p>Also restricts the maximum size of the volumes it manages for LUNs.</p>	"" (not enforced by default)
lunsPerFlexvol	Maximum LUNs per Flexvol, must be in range [50, 200]	100

Parameter	Description	Default
debugTraceFlags	<p>Debug flags to use when troubleshooting. Example, {"api":false, "method":true}</p> <p>Do not use unless you are troubleshooting and require a detailed log dump.</p>	null
useREST	<p>Boolean parameter to use ONTAP REST APIs.</p> <p>useREST When set to <code>true</code>, Trident uses ONTAP REST APIs to communicate with the backend; when set to <code>false</code>, Trident uses ONTAPI (ZAPI) calls to communicate with the backend. This feature requires ONTAP 9.11.1 and later. In addition, the ONTAP login role used must have access to the <code>ontapi</code> application. This is satisfied by the pre-defined <code>vsadmin</code> and <code>cluster-admin</code> roles. Beginning with the Trident 24.06 release and ONTAP 9.15.1 or later, <code>useREST</code> is set to <code>true</code> by default; change <code>useREST</code> to <code>false</code> to use ONTAPI (ZAPI) calls.</p> <p>useREST is fully qualified for NVMe/TCP.</p> <div>  <p>NVMe is supported only with ONTAP REST APIs and not supported with ONTAPI (ZAPI).</p> </div> <p>If specified, always set to <code>true</code> for ASA r2 systems.</p>	<code>true</code> for ONTAP 9.15.1 or later, otherwise <code>false</code> .
sanType	Use to select <code>iscsi</code> for iSCSI, <code>nvme</code> for NVMe/TCP or <code>fc</code> for SCSI over Fibre Channel (FC).	<code>iscsi</code> if blank
formatOptions	<p>Use <code>formatOptions</code> to specify command line arguments for the <code>mkfs</code> command, which is applied whenever a volume is formatted. This allows you to format the volume according to your preferences. Make sure to specify the <code>formatOptions</code> similar to that of the <code>mkfs</code> command options, excluding the device path. Example: <code>"-E nodiscard"</code></p> <p>Supported for <code>ontap-san</code> and <code>ontap-san-economy</code> drivers with iSCSI protocol. Additionally, supported for ASA r2 systems when using iSCSI and NVMe/TCP protocols.</p>	
limitVolumePoolSize	Maximum requestable FlexVol size when using LUNs in <code>ontap-san-economy</code> backend.	"" (not enforced by default)
denyNewVolumePools	Restricts <code>ontap-san-economy</code> backends from creating new FlexVol volumes to contain their LUNs. Only preexisting Flexvols are used for provisioning new PVs.	

Recommendations for using formatOptions

Trident recommends the following options to expedite the formatting process:

- **-E nodiscard (ext3, ext4):** Do not attempt to discard blocks at mkfs time (discarding blocks initially is useful on solid state devices and sparse / thin-provisioned storage). This replaces the deprecated option "-K" and it is applicable to ext3, ext4 file systems.
- **-K (xfs):** Do not attempt to discard blocks at mkfs time. This option is applicable to xfs file system.

Authenticate Trident to a backend SVM using Active Directory credentials

You can configure Trident to authenticate to a backend SVM using Active Directory (AD) credentials. Before an AD account can access the SVM, you must configure AD domain controller access to the cluster or SVM. For cluster administration with an AD account, you must create domain tunnel. Refer to [Configure Active Directory domain controller access in ONTAP](#) for details.

steps

1. Configure Domain Name System (DNS) settings for a backend SVM:

```
vserver services dns create -vserver <svm_name> -dns-servers  
<dns_server_ip1>,<dns_server_ip2>
```

2. Run the following command to create a computer account for the SVM in Active Directory:

```
vserver active-directory create -vserver DataSVM -account-name ADSERVER1  
-domain demo.netapp.com
```

3. Use this command to create an AD user or group to manage the cluster or SVM

```
security login create -vserver <svm_name> -user-or-group-name  
<ad_user_or_group> -application <application> -authentication-method domain  
-role vsadmin
```

4. In the Trident backend configuration file, set the `username` and `password` parameters to the AD user or group name and password, respectively.

Backend configuration options for provisioning volumes

You can control default provisioning using these options in the `defaults` section of the configuration. For an example, see the configuration examples below.

Parameter	Description	Default
<code>spaceAllocation</code>	Space-allocation for LUNs	"true" If specified, set to true for ASA r2 systems.
<code>spaceReserve</code>	Space reservation mode; "none" (thin) or "volume" (thick). Set to none for ASA r2 systems.	"none"

Parameter	Description	Default
snapshotPolicy	Snapshot policy to use. Set to <code>none</code> for ASA r2 systems.	"none"
qosPolicy	QoS policy group to assign for volumes created. Choose one of qosPolicy or adaptiveQosPolicy per storage pool/backend. Using QoS policy groups with Trident requires ONTAP 9.8 or later. You should use a non-shared QoS policy group and ensuring the policy group is applied to each constituent individually. A shared QoS policy group enforces the ceiling for the total throughput of all workloads.	""
adaptiveQosPolicy	Adaptive QoS policy group to assign for volumes created. Choose one of qosPolicy or adaptiveQosPolicy per storage pool/backend	""
snapshotReserved	Percentage of volume reserved for snapshots. Do not specify for ASA r2 systems.	"0" if snapshotPolicy is "none", otherwise ""
splitOnClone	Split a clone from its parent upon creation	"false"
encryption	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. If NAE is enabled on the backend, any volume provisioned in Trident is NAE enabled. For more information, refer to: How Trident works with NVE and NAE .	"false" If specified, set to <code>true</code> for ASA r2 systems.
luksEncryption	Enable LUKS encryption. Refer to Use Linux Unified Key Setup (LUKS) .	"" Set to <code>false</code> for ASA r2 systems.
tieringPolicy	Tiering policy to use "none" Do not specify for ASA r2 systems .	
nameTemplate	Template to create custom volume names.	""

Volume provisioning examples

Here's an example with defaults defined:

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: trident_svm
```

```

username: admin
password: <password>
labels:
  k8scluster: dev2
  backend: dev2-sanbackend
storagePrefix: alternate-trident
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: standard
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'

```



For all volumes created using the `ontap-san` driver, Trident adds an extra 10 percent capacity to the FlexVol to accommodate the LUN metadata. The LUN is provisioned with the exact size that the user requests in the PVC. Trident adds 10 percent to the FlexVol (shows as Available size in ONTAP). Users now get the amount of usable capacity they requested. This change also prevents LUNs from becoming read-only unless the available space is fully utilized. This does not apply to `ontap-san-economy`.

For backends that define `snapshotReserve`, Trident calculates the size of volumes as follows:

```

Total volume size = [(PVC requested size) / (1 - (snapshotReserve
percentage) / 100)] * 1.1

```

The 1.1 is the extra 10 percent Trident adds to the FlexVol to accommodate the LUN metadata. For `snapshotReserve` = 5%, and PVC request = 5 GiB, the total volume size is 5.79 GiB and the available size is 5.5 GiB. The `volume show` command should show results similar to this example:

Vserver	Volume	Aggregate	State	Type	Size	Available	Used%
		_pvc_89f1c156_3801_4de4_9f9d_034d54c395f4	online	RW	10GB	5.00GB	0%
		_pvc_e42ec6fe_3baa_4af6_996d_134adbbb8e6d	online	RW	5.79GB	5.50GB	0%
		_pvc_e8372153_9ad9_474a_951a_08ae15e1c0ba	online	RW	1GB	511.8MB	0%

3 entries were displayed.

Currently, resizing is the only way to use the new calculation for an existing volume.

Minimal configuration examples

The following examples show basic configurations that leave most parameters to default. This is the easiest way to define a backend.



If you are using Amazon FSx on NetApp ONTAP with Trident, NetApp recommends that you specify DNS names for LIFs instead of IP addresses.

▼ ONTAP SAN example

This is a basic configuration using the `ontap-san` driver.

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
username: vsadmin
password: <password>
```

▼ MetroCluster example

You can configure the backend to avoid having to manually update the backend definition after switchover and switchback during [SVM replication and recovery](#).

For seamless switchover and switchback, specify the SVM using `managementLIF` and omit the `svm` parameters. For example:

```
version: 1
storageDriverName: ontap-san
managementLIF: 192.168.1.66
username: vsadmin
password: password
```

▼ ONTAP SAN economy example

```
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
username: vsadmin
password: <password>
```

▼ Certificate-based authentication example

In this basic configuration example `clientCertificate`, `clientPrivateKey`, and `trustedCACertificate` (optional, if using trusted CA) are populated in `backend.json` and take the base64-encoded values of the client certificate, private key, and trusted CA certificate, respectively.

```

---
version: 1
storageDriverName: ontap-san
backendName: DefaultSANBackend
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz

```

▼ Bidirectional CHAP examples

These examples create a backend with useCHAP set to true.

ONTAP SAN CHAP example

```

---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
username: vsadmin
password: <password>

```

ONTAP SAN economy CHAP example

```

---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy

```

```
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
username: vsadmin
password: <password>
```

▼ Balanced placement example

This example enables ONTAP balanced placement so that ONTAP selects the aggregate for new volumes. Balanced placement requires ONTAP 9.19.1 or later. Refer to [Use balanced placement with ONTAP backends](#) for details.

```
---
version: 1
storageDriverName: ontap-san
backendName: san-balanced-placement
managementLIF: 10.0.0.1
svm: svm_iscsi
username: vsadmin
password: <password>
useBalancedPlacement: true
```

▼ NVMe/TCP example

You must have an SVM configured with NVMe on your ONTAP backend. This is a basic backend configuration for NVMe/TCP.

```
---
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 (FCP) example

You must have an SVM configured with FC on your ONTAP backend. This is a basic backend configuration for FC.

```
---
version: 1
backendName: fcp-backend
storageDriverName: ontap-san
```

```
managementLIF: 10.0.0.1
svm: svm_fc
username: vsadmin
password: password
sanType: fcp
useREST: true
```

▼ Backend configuration example with nameTemplate

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap-san-backend
managementLIF: <ip address>
svm: svm0
username: <admin>
password: <password>
defaults:
  nameTemplate:
    "{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.vo\
      lume.RequestName}}"
labels:
  cluster: ClusterA
  PVC: "{{.volume.Namespace}}_{{.volume.RequestName}}"
```

▼ formatOptions example for ontap-san-economy driver

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: ""
svm: svm1
username: ""
password: "!"
storagePrefix: whelk_
debugTraceFlags:
  method: true
  api: true
defaults:
  formatOptions: -E nodiscard
```

Examples of backends with virtual pools

In these sample backend definition files, specific defaults are set for all storage pools, such as `spaceReserve` at none, `spaceAllocation` at false, and `encryption` at false. The virtual pools are defined in the storage

section.

Trident sets provisioning labels in the "Comments" field. Comments are set on the FlexVol volume Trident copies all labels present on a virtual pool to the storage volume at provisioning. For convenience, storage administrators can define labels per virtual pool and group volumes by label.

In these examples, some of the storage pools set their own `spaceReserve`, `spaceAllocation`, and encryption values, and some pools override the default values.

▼ ONTAP SAN example

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: "false"
  encryption: "false"
  qosPolicy: standard
labels:
  store: san_store
  kubernetes-cluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
      protection: gold
      creditpoints: "40000"
      zone: us_east_1a
      defaults:
        spaceAllocation: "true"
        encryption: "true"
        adaptiveQosPolicy: adaptive-extreme
  - labels:
      protection: silver
      creditpoints: "20000"
      zone: us_east_1b
      defaults:
        spaceAllocation: "false"
        encryption: "true"
        qosPolicy: premium
```

```

- labels:
  protection: bronze
  creditpoints: "5000"
  zone: us_east_1c
  defaults:
    spaceAllocation: "true"
    encryption: "false"

```

▼ ONTAP SAN economy example

```

---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: "false"
  encryption: "false"
labels:
  store: san_economy_store
region: us_east_1
storage:
  - labels:
    app: oracledb
    cost: "30"
    zone: us_east_1a
    defaults:
      spaceAllocation: "true"
      encryption: "true"
  - labels:
    app: postgresdb
    cost: "20"
    zone: us_east_1b
    defaults:
      spaceAllocation: "false"
      encryption: "true"
  - labels:
    app: mysqldb
    cost: "10"

```

```

zone: us_east_1c
defaults:
  spaceAllocation: "true"
  encryption: "false"
- labels:
  department: legal
  creditpoints: "5000"
zone: us_east_1c
defaults:
  spaceAllocation: "true"
  encryption: "false"

```

▼ NVMe/TCP example

```

---
version: 1
storageDriverName: ontap-san
sanType: nvme
managementLIF: 10.0.0.1
svm: nvme_svm
username: vsadmin
password: <password>
useREST: true
defaults:
  spaceAllocation: "false"
  encryption: "true"
storage:
- labels:
  app: testApp
  cost: "20"
defaults:
  spaceAllocation: "false"
  encryption: "false"

```

Map backends to StorageClasses

The following StorageClass definitions refer to the [Examples of backends with virtual pools](#). Using the `parameters.selector` field, each StorageClass calls out which virtual pools can be used to host a volume. The volume now has the aspects defined in the chosen virtual pool.

- The `protection-gold` StorageClass maps to the first virtual pool in the `ontap-san` backend. This is the only pool offering gold-level protection.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:

```

```

name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"

```

- The **protection-not-gold** StorageClass maps to the second and third virtual pool in **ontap-san** backend. These are the only pools offering a protection level other than gold.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection!=gold"
  fsType: "ext4"

```

- The **app-mysqldb** StorageClass maps to the third virtual pool in **ontap-san-economy** backend. This is the only pool offering storage pool configuration for the **mysqldb** type app.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"

```

- The **protection-silver-creditpoints-20k** StorageClass maps to the second virtual pool in **ontap-san** backend. This is the only pool offering silver-level protection and 20000 creditpoints.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"

```

- The **creditpoints-5k** StorageClass maps to the third virtual pool in **ontap-san** backend and the fourth

virtual pool in the `ontap-san-economy` backend. These are the only pool offerings with 5000 creditpoints.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: csi.trident.netapp.io
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"
```

- The `my-test-app-sc` StorageClass maps to the `testAPP` virtual pool in the `ontap-san` driver with `sanType: nvme`. This is the only pool offering `testApp`.

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: my-test-app-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=testApp"
  fsType: "ext4"
```

Trident decides which virtual pool is selected and ensures the storage requirement is met.

Copyright information

Copyright © 2026 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.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

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.