# Table of Contents

ONTAP NAS drivers ........................................................................................................ 1
  ONTAP NAS driver overview ..................................................................................... 1
Prepare to configure a backend with ONTAP NAS drivers ........................................ 2
ONTAP NAS configuration options and examples ....................................................... 10
ONTAP NAS drivers

ONTAP NAS driver overview

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

ONTAP NAS driver details

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

If you are using Astra Control for protection, recovery, and mobility, read Astra Control driver compatibility.

<table>
<thead>
<tr>
<th>Driver</th>
<th>Protocol</th>
<th>volumeMode</th>
<th>Access modes supported</th>
<th>File systems supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>ontap-nas</td>
<td>NFS</td>
<td>Filesystem</td>
<td>RWO, ROX, RWX, RWOP</td>
<td>&quot;&quot;, nfs, smb</td>
</tr>
<tr>
<td>ontap-nas-economy</td>
<td>NFS</td>
<td>Filesystem</td>
<td>RWO, ROX, RWX, RWOP</td>
<td>&quot;&quot;, nfs, smb</td>
</tr>
<tr>
<td>ontap-nas-flexgroup</td>
<td>NFS</td>
<td>Filesystem</td>
<td>RWO, ROX, RWX, RWOP</td>
<td>&quot;&quot;, nfs, smb</td>
</tr>
</tbody>
</table>

Astra Control driver compatibility

Astra Control provides seamless protection, disaster recovery, and mobility (moving volumes between Kubernetes clusters) for volumes created with the ontap-nas, ontap-nas-flexgroup, and ontap-san drivers. Refer to Astra Control replication prerequisites for details.

- 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 use ontap-nas-economy if you anticipate the need for data protection, disaster recovery, or mobility.

User permissions

Astra 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, Astra 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 Astra 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.

**Prepare to configure a backend with ONTAP NAS drivers**

Understand the requirements, authentication options, and export policies for configuring an ONTAP backend with ONTAP NAS drivers.

**Requirements**

- For all ONTAP backends, Astra Trident requires at least one aggregate assigned to the SVM.
- You can run more than one driver, and create storage classes that point to one or the other. For example, you could configure a Gold class that uses the ontap-nas driver and a Bronze class that uses the ontap-nas-economy one.
- All your Kubernetes worker nodes must have the appropriate NFS tools installed. Refer to [here](#) for more details.
- Astra Trident supports SMB volumes mounted to pods running on Windows nodes only. Refer to [Prepare to provision SMB volumes](#) for details.

**Authenticate the ONTAP backend**

Astra Trident offers two modes of authenticating an ONTAP backend.

- **Credential-based:** This mode requires sufficient permissions to the ONTAP backend. It is recommended to use an account associated with a pre-defined security login role, such as admin or vsadmin to ensure maximum compatibility with ONTAP versions.
- **Certificate-based:** This mode requires a certificate installed on the backend for Astra Trident to communicate with an ONTAP cluster. 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**

Astra 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 Astra Trident releases. A custom security login role can be created and used with Astra Trident, but is not recommended.

A sample backend definition will look like this:

**YAML**

```yaml
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

**JSON**

```json
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "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/updation 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=vsadmin"
```

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
```

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

```
security login create -user-or-group-name vsadmin -application ontapi -authentication-method cert -vserver <vserver-name>
security login create -user-or-group-name vsadmin -application http -authentication-method cert -vserver <vserver-name>
```

5. Test authentication using certificate generated. Replace <ONTAP Management LIF> and <vserver name> with Management LIF IP and SVM name. You must ensure the LIF has its service policy set to default-data-management.

```
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>"/></netapp>'
```

6. Encode certificate, key and trusted CA certificate with Base64.
7. Create backend using the values obtained from the previous step.

```
cat cert-backend-updated.json
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "NasBackend",
    "managementLIF": "1.2.3.4",
    "dataLIF": "1.2.3.8",
    "svm": "vserver_test",
    "clientCertificate": "Faaaakkkkeeee...Vaaalllluuuuueeee",
    "clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
    "storagePrefix": "myPrefix_"
}
```

#Update backend with tridentctl
tridentctl update backend NasBackend -f cert-backend-updated.json -n trident

```
+-----------------+--------------------------------------+
|    NAME    | STORAGE DRIVER |                 UUID                 |
| STATE  | VOLUMES | +-----------------+--------------------------------------+
| NasBackend | ontap-nas   | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
| online | 9 | +-----------------+--------------------------------------+
```

### 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 update backend`. 
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 Astra Trident can communicate with the ONTAP backend and handle future volume operations.

### Manage NFS export policies

Astra Trident uses NFS export policies to control access to the volumes that it provisions.

Astra Trident provides two options when working with export policies:

- Astra Trident can dynamically manage the export policy itself; in this mode of operation, the storage administrator specifies a list of CIDR blocks that represent admissible IP addresses. Astra Trident adds node IPs that fall in these ranges to the export policy automatically. Alternatively, when no CIDRs are specified, any global-scoped unicast IP found on the nodes will be added to the export policy.
- Storage administrators can create an export policy and add rules manually. Astra Trident uses the default
export policy unless a different export policy name is specified in the configuration.

Dynamically manage export policies

Astra Trident provides the ability to dynamically manage export policies for ONTAP backends. This provides the storage administrator the ability to specify a permissible address space for worker node IPs, rather than defining explicit rules manually. It greatly simplifies export policy management; modifications to the export policy no longer require manual intervention on the storage cluster. Moreover, this helps restrict access to the storage cluster only to worker nodes that have IPs in the range specified, supporting a fine-grained and automated management.

Do not use Network Address Translation (NAT) when using dynamic export policies. With NAT, the storage controller sees the frontend NAT address and not the actual IP host address, so access will be denied when no match is found in the export rules.

Example

There are two configuration options that must be used. Here’s an example backend definition:

```yaml
---
version: 1
storageDriverName: ontap-nas
backendName: ontap_nas_auto_export
managementLIF: 192.168.0.135
svm: svm1
username: vsadmin
password: password
autoExportCIDRs:
- 192.168.0.0/24
autoExportPolicy: true
```

When using this feature, you must ensure that the root junction in your SVM has a previously created export policy with an export rule that permits the node CIDR block (such as the default export policy). Always follow NetApp recommended best practice to dedicate an SVM for Astra Trident.

Here is an explanation of how this feature works using the example above:

- **autoExportPolicy** is set to true. This indicates that Astra Trident will create an export policy for the svm1 SVM and handle the addition and deletion of rules using autoExportCIDRs address blocks. For example, a backend with UUID 403b5326-8482-40db-96d0-d83fb3f4daec and autoExportPolicy set to true creates an export policy named trident-403b5326-8482-40db-96d0-d83fb3f4daec on the SVM.

- **autoExportCIDRs** contains a list of address blocks. This field is optional and it defaults to ["0.0.0.0/0", ":/0"]. If not defined, Astra Trident adds all globally-scoped unicast addresses found on the worker nodes.

In this example, the 192.168.0.0/24 address space is provided. This indicates that Kubernetes node IPs that fall within this address range will be added to the export policy that Astra Trident creates. When Astra Trident registers a node it runs on, it retrieves the IP addresses of the node and checks them against the
address blocks provided in autoExportCIDRs. After filtering the IPs, Astra Trident creates export policy rules for the client IPs it discovers, with one rule for each node it identifies.

You can update autoExportPolicy and autoExportCIDRs for backends after you create them. You can append new CIDRs for a backend that is automatically managed or delete existing CIDRs. Exercise care when deleting CIDRs to ensure that existing connections are not dropped. You can also choose to disable autoExportPolicy for a backend and fall back to a manually created export policy. This will require setting the exportPolicy parameter in your backend config.

After Astra Trident creates or updates a backend, you can check the backend using tridentctl or the corresponding tridentbackend CRD:

```
./tridentctl get backends ontap_nas_auto_export -n trident -o yaml
```

```
items:
- backendUUID: 403b5326-8482-40db-96d0-d83fb3f4daec
  config:
    aggregate: ""
    autoExportCIDRs:
    - 192.168.0.0/24
    autoExportPolicy: true
    backendName: ontap_nas_auto_export
    chapInitiatorSecret: ""
    chapTargetInitiatorSecret: ""
    chapTargetUsername: ""
    chapUsername: ""
    dataLIF: 192.168.0.135
    debug: false
    debugTraceFlags: null
    defaults:
      encryption: "false"
      exportPolicy: <automatic>
     FileSystemType: ext4
```

As nodes are added to a Kubernetes cluster and registered with the Astra Trident controller, export policies of existing backends are updated (provided they fall in the address range specified in autoExportCIDRs for the backend).

When a node is removed, Astra Trident checks all backends that are online to remove the access rule for the node. By removing this node IP from the export policies of managed backends, Astra Trident prevents rogue mounts, unless this IP is reused by a new node in the cluster.

For previously existing backends, updating the backend with tridentctl update backend will ensure that Astra Trident manages the export policies automatically. This will create a new export policy named after the backend’s UUID and volumes that are present on the backend will use the newly created export policy when they are mounted again.
Deleting a backend with auto-managed export policies will delete the dynamically created export policy. If the backend is re-created, it is treated as a new backend and will result in the creation of a new export policy.

If the IP address of a live node is updated, you must restart the Astra Trident pod on the node. Astra Trident will then update the export policy for backends it manages to reflect this IP change.

**Prepare to provision SMB volumes**

With a little additional preparation, you can provision SMB volumes using `ontap-nas` drivers.

You must configure both NFS and SMB/CIFS protocols on the SVM to create an `ontap-nas-economy` SMB volume for ONTAP on-premises. Failure to configure either of these protocols will cause SMB volume creation to fail.

**Before you begin**

Before you can provision SMB volumes, you must have the following.

- A Kubernetes cluster with a Linux controller node and at least one Windows worker node running Windows Server 2019. Astra Trident supports SMB volumes mounted to pods running on Windows nodes only.
- At least one Astra Trident secret containing your Active Directory credentials. To generate secret `smbcreds`:

  ```bash
  kubectl create secret generic smbcreds --from-literal username=user
  --from-literal password='password'
  ```

- A CSI proxy configured as a Windows service. To configure a `csi-proxy`, refer to [GitHub: CSI Proxy](https://github.com/csiprotocoldevelopers/csi-proxy) or [GitHub: CSI Proxy for Windows](https://github.com/csiprotocoldevelopers/csi-proxy-for-windows) for Kubernetes nodes running on Windows.

**Steps**

1. For on-premises ONTAP, you can optionally create an SMB share or Astra Trident can create one for you.

   - **SMB shares are required for Amazon FSx for ONTAP.**

   You can create the SMB admin shares in one of two ways either using the Microsoft Management Console Shared Folders snap-in or using the ONTAP CLI. To create the SMB shares using the ONTAP CLI:

   a. If necessary, create the directory path structure for the share.

      The `vserver cifs share create` command checks the path specified in the `-path` option during share creation. If the specified path does not exist, the command fails.

   b. Create an SMB share associated with the specified SVM:

      ```bash
      vserver cifs share create -vserver vserver_name -share-name share_name -path path [-share-properties share_properties,...]
      [other_attributes] [-comment text]
      ```
c. Verify that the share was created:

```bash
vserver cifs share show -share-name share_name
```

Refer to Create an SMB share for full details.

2. When creating the backend, you must configure the following to specify SMB volumes. For all FSx for ONTAP backend configuration options, refer to FSx for ONTAP configuration options and examples.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>smbShare</td>
<td>You can specify one of the following: the name of an SMB share created using the Microsoft Management Console or ONTAP CLI; a name to allow Astra Trident to create the SMB share; or you can leave the parameter blank to prevent common share access to volumes. This parameter is optional for on-premises ONTAP. This parameter is required for Amazon FSx for ONTAP backends and cannot be blank.</td>
<td>smb-share</td>
</tr>
<tr>
<td>nasType</td>
<td>Must set to smb. If null, defaults to nfs.</td>
<td>smb</td>
</tr>
<tr>
<td>securityStyle</td>
<td>Security style for new volumes. Must be set to ntfs or mixed for SMB volumes.</td>
<td>ntfs or mixed</td>
</tr>
<tr>
<td>unixPermissions</td>
<td>Mode for new volumes. Must be left empty for SMB volumes.</td>
<td>&quot;&quot;</td>
</tr>
</tbody>
</table>

ONTAP NAS configuration options and examples

Learn to create and use ONTAP NAS drivers with your Astra Trident installation. This section provides backend configuration examples and details for mapping backends to StorageClasses.

Backend configuration options

See the following table for the backend configuration options:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td></td>
<td>Always 1</td>
</tr>
<tr>
<td>storageDrive rName</td>
<td>Name of the storage driver</td>
<td>&quot;ontap-nas&quot;, &quot;ontap-nas-economy&quot;, &quot;ontap-nas-flexgroup&quot;, &quot;ontap-san&quot;, &quot;ontap-san-economy&quot;</td>
</tr>
<tr>
<td>backendName</td>
<td>Custom name or the storage backend</td>
<td>Driver name + &quot;_&quot; + dataLIF</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>managementLIF</td>
<td>IP address of a cluster or SVM management LIF</td>
<td>“10.0.0.1&quot;, [2001:1234:abcd::fefe]</td>
</tr>
<tr>
<td></td>
<td>A fully-qualified domain name (FQDN) can be specified.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Can be set to use IPv6 addresses if Astra Trident was installed using the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IPv6 flag. IPv6 addresses must be defined in square brackets, such as</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For seamless MetroCluster switchover, see the MetroCluster example.</td>
<td></td>
</tr>
<tr>
<td>dataLIF</td>
<td>IP address of protocol LIF.</td>
<td>Specified address or derived from SVM, if not specified (not recommended)</td>
</tr>
<tr>
<td></td>
<td>We recommend specifying dataLIF. If not provided, Astra Trident fetches</td>
<td></td>
</tr>
<tr>
<td></td>
<td>data LIFs from the SVM. You can specify a fully-qualified domain name (FQDN)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>to be used for the NFS mount operations, allowing you to create a round-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>robin DNS to load-balance across multiple data LIFs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Can be changed after initial setting. Refer to Update dataLIF after initial</td>
<td></td>
</tr>
<tr>
<td></td>
<td>configuration.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Can be set to use IPv6 addresses if Astra Trident was installed using the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IPv6 flag. IPv6 addresses must be defined in square brackets, such as</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Omit for Metrocluster. See the MetroCluster example.</td>
<td></td>
</tr>
<tr>
<td>svm</td>
<td>Storage virtual machine to use</td>
<td>Derived if an SVM managementLIF is specified</td>
</tr>
<tr>
<td></td>
<td>Omit for Metrocluster. See the MetroCluster example.</td>
<td></td>
</tr>
<tr>
<td>autoExportPolicy</td>
<td>Enable automatic export policy creation and updating [Boolean].</td>
<td>false</td>
</tr>
<tr>
<td></td>
<td>Using the autoExportPolicy and autoExportCIDRs options, Astra Trident can</td>
<td></td>
</tr>
<tr>
<td></td>
<td>manage export policies automatically.</td>
<td></td>
</tr>
<tr>
<td>autoExportCIDRs</td>
<td>List of CIDRs to filter Kubernetes’ node IPs against when autoExportPolicy</td>
<td>[&quot;0.0.0.0/0&quot;, &quot;:/0&quot;]</td>
</tr>
<tr>
<td></td>
<td>is enabled.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Using the autoExportPolicy and autoExportCIDRs options, Astra Trident can</td>
<td></td>
</tr>
<tr>
<td></td>
<td>manage export policies automatically.</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>labels</td>
<td>Set of arbitrary JSON-formatted labels to apply on volumes</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>clientCertificate</td>
<td>Base64-encoded value of client certificate. Used for certificate-based auth</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>clientPrivateKey</td>
<td>Base64-encoded value of client private key. Used for certificate-based auth</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>trustedCACertificate</td>
<td>Base64-encoded value of trusted CA certificate. Optional. Used for certificate-based auth</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>username</td>
<td>Username to connect to the cluster/SVM. Used for credential-based auth</td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>Password to connect to the cluster/SVM. Used for credential-based auth</td>
<td></td>
</tr>
<tr>
<td>storagePrefix</td>
<td>Prefix used when provisioning new volumes in the SVM. Cannot be updated after you set it</td>
<td>&quot;trident&quot;</td>
</tr>
<tr>
<td>limitAggregateUsage</td>
<td>Fail provisioning if usage is above this percentage.</td>
<td>&quot;&quot; (not enforced by default)</td>
</tr>
<tr>
<td></td>
<td><strong>Does not apply to Amazon FSx for ONTAP</strong></td>
<td></td>
</tr>
<tr>
<td>limitVolumeSize</td>
<td>Fail provisioning if requested volume size is above this value.</td>
<td>&quot;&quot; (not enforced by default)</td>
</tr>
<tr>
<td></td>
<td>Also restricts the maximum size of the volumes it manages for qtrees and LUNs, and the qtreesPerFlexvol option allows customizing the maximum number of qtrees per FlexVol.</td>
<td></td>
</tr>
<tr>
<td>lunsPerFlexvol</td>
<td>Maximum LUNs per Flexvol, must be in range [50, 200]</td>
<td>&quot;100&quot;</td>
</tr>
<tr>
<td>debugTraceFlags</td>
<td>Debug flags to use when troubleshooting. Example, {&quot;api&quot;:false, &quot;method&quot;:true}</td>
<td>null</td>
</tr>
<tr>
<td></td>
<td>Do not use debugTraceFlags unless you are troubleshooting and require a detailed log dump.</td>
<td></td>
</tr>
<tr>
<td>nasType</td>
<td>Configure NFS or SMB volumes creation.</td>
<td>nfs</td>
</tr>
<tr>
<td></td>
<td>Options are nfs, smb or null. Setting to null defaults to NFS volumes.</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>nfsMountOptions</td>
<td>Comma-separated list of NFS mount options.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The mount options for Kubernetes-persistent volumes are normally specified in</td>
<td></td>
</tr>
<tr>
<td></td>
<td>storage classes, but if no mount options are specified in a storage class,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Astra Trident will fall back to using the mount options specified in the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>storage backend's configuration file.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If no mount options are specified in the storage class or the configuration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>file, Astra Trident will not set any mount options on an associated persistent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>volume.</td>
<td></td>
</tr>
<tr>
<td>qtreesPerFlexvol</td>
<td>Maximum Qtrees per FlexVol, must be in range [50, 300]</td>
<td>&quot;200&quot;</td>
</tr>
<tr>
<td>smbShare</td>
<td>You can specify one of the following: the name of an SMB share created</td>
<td>smb-share</td>
</tr>
<tr>
<td></td>
<td>using the Microsoft Management Console or ONTAP CLI; a name to allow Astra</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trident to create the SMB share; or you can leave the parameter blank to</td>
<td></td>
</tr>
<tr>
<td></td>
<td>prevent common share access to volumes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This parameter is optional for on-premises ONTAP.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This parameter is required for Amazon FSx for ONTAP backends and cannot be</td>
<td></td>
</tr>
<tr>
<td></td>
<td>blank.</td>
<td></td>
</tr>
<tr>
<td>useREST</td>
<td>Boolean parameter to use ONTAP REST APIs. <strong>Tech preview</strong></td>
<td>false</td>
</tr>
<tr>
<td></td>
<td>useREST is provided as a <strong>tech preview</strong> that is recommended for test</td>
<td></td>
</tr>
<tr>
<td></td>
<td>environments and not for production workloads. When set to true, Astra</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trident will use ONTAP REST APIs to communicate with the backend. This</td>
<td></td>
</tr>
<tr>
<td></td>
<td>feature requires ONTAP 9.11.1 and later. In addition, the ONTAP login role</td>
<td></td>
</tr>
<tr>
<td></td>
<td>used must have access to the <strong>ontap</strong> application. This is satisfied by the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pre-defined vsadmin and cluster-admin roles.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>useREST is not supported with MetroCluster.</td>
<td></td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>spaceAllocation</td>
<td>Space-allocation for LUNs</td>
<td>&quot;true&quot;</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>spaceReserve</td>
<td>Space reservation mode; &quot;none&quot; (thin) or &quot;volume&quot; (thick)</td>
<td>&quot;none&quot;</td>
</tr>
<tr>
<td>snapshotPolicy</td>
<td>Snapshot policy to use</td>
<td>&quot;none&quot;</td>
</tr>
<tr>
<td>qosPolicy</td>
<td>QoS policy group to assign for volumes created. Choose one of qosPolicy or</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td></td>
<td>adaptiveQosPolicy per storage pool/backend</td>
<td></td>
</tr>
<tr>
<td>adaptiveQosPolicy</td>
<td>Adaptive QoS policy group to assign for volumes created. Choose one of</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td></td>
<td>qosPolicy or adaptiveQosPolicy per storage pool/backend. Not supported by</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ontap-nas-economy.</td>
<td></td>
</tr>
<tr>
<td>snapshotReserve</td>
<td>Percentage of volume reserved for snapshots</td>
<td>&quot;0&quot; if snapshotPolicy is &quot;none&quot;, otherwise &quot;&quot;</td>
</tr>
<tr>
<td>splitOnClone</td>
<td>Split a clone from its parent upon creation</td>
<td>&quot;false&quot;</td>
</tr>
<tr>
<td>encryption</td>
<td>Enable NetApp Volume Encryption (NVE) on the new volume; defaults to false.</td>
<td>&quot;false&quot;</td>
</tr>
<tr>
<td></td>
<td>NVE must be licensed and enabled on the cluster to use this option.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If NAE is enabled on the backend, any volume provisioned in Astra Trident</td>
<td></td>
</tr>
<tr>
<td></td>
<td>will be NAE enabled.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For more information, refer to: How Astra Trident works with NVE and NAE.</td>
<td></td>
</tr>
<tr>
<td>tieringPolicy</td>
<td>Tiering policy to use &quot;none&quot;</td>
<td>&quot;snapshot-only&quot; for pre-ONTAP 9.5 SVM-DR</td>
</tr>
<tr>
<td>unixPermissions</td>
<td>Mode for new volumes</td>
<td>&quot;777&quot; for NFS volumes; empty (not applicable)</td>
</tr>
<tr>
<td></td>
<td>for SMB volumes</td>
<td></td>
</tr>
<tr>
<td>snapshotDir</td>
<td>Controls access to the .snapshot directory</td>
<td>&quot;false&quot;</td>
</tr>
<tr>
<td>exportPolicy</td>
<td>Export policy to use</td>
<td>&quot;default&quot;</td>
</tr>
<tr>
<td>securityStyle</td>
<td>Security style for new volumes.</td>
<td>NFS default is unix.</td>
</tr>
<tr>
<td></td>
<td>NFS supports mixed and unix security styles.</td>
<td>SMB default is ntfs.</td>
</tr>
<tr>
<td></td>
<td>SMB supports mixed and ntfs security styles.</td>
<td></td>
</tr>
</tbody>
</table>

Using QoS policy groups with Astra Trident requires ONTAP 9.8 or later. It is recommended to use a non-shared QoS policy group and ensure the policy group is applied to each constituent individually. A shared QoS policy group will enforce the ceiling for the total throughput of all workloads.

Volume provisioning examples

Here's an example with defaults defined:
For ontap-nas and ontap-nas-flexgroups, Astra Trident now uses a new calculation to ensure that the FlexVol is sized correctly with the snapshotReserve percentage and PVC. When the user requests a PVC, Astra Trident creates the original FlexVol with more space by using the new calculation. This calculation ensures that the user receives the writable space they requested for in the PVC, and not lesser space than what they requested. Before v21.07, when the user requests a PVC (for example, 5GiB), with the snapshotReserve to 50 percent, they get only 2.5GiB of writable space. This is because what the user requested for is the whole volume and snapshotReserve is a percentage of that. With Trident 21.07, what the user requests for is the writeable space and Astra Trident defines the snapshotReserve number as the percentage of the whole volume. This does not apply to ontap-nas-economy. See the following example to see how this works:

The calculation is as follows:

\[
\text{Total volume size} = \frac{(\text{PVC requested size})}{1 - \left(\frac{\text{snapshotReserve \%}}{100}\right)}
\]

For snapshotReserve = 50%, and PVC request = 5GiB, the total volume size is \(\frac{5}{1 - (0.5)} = 10\text{GiB}\) and the available size is 5GiB, which is what the user requested in the PVC request. The `volume show` command should show results similar to this example:
Existing backends from previous installs will provision volumes as explained above when upgrading Astra Trident. For volumes that you created before upgrading, you should resize their volumes for the change to be observed. For example, a 2GiB PVC with `snapshotReserve=50` earlier resulted in a volume that provides 1GiB of writable space. Resizing the volume to 3GiB, for example, provides the application with 3GiB of writable space on a 6 GiB 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, the recommendation is to specify DNS names for LIFs instead of IP addresses.

**ONTAP NAS economy example**

```yaml
---
version: 1
storageDriverName: ontap-nas-economy
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

**ONTAP NAS Flexgroup example**

```yaml
---
version: 1
storageDriverName: ontap-nas-flexgroup
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
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 `dataLIF` and `svm` parameters. For example:

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

**SMB volumes example**

```yaml
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
nasType: smb
securityStyle: ntfs
unixPermissions: ""
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
---
```
Certificate-based authentication example

This is a minimal backend 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.

```json
---
version: 1
backendName: DefaultNASBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.15
svm: nfs_svm
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
---
```

Auto export policy example

This example shows you how you can instruct Astra Trident to use dynamic export policies to create and manage the export policy automatically. This works the same for the `ontap-nas-economy` and `ontap-nas-flexgroup` drivers.

```json
---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
labels:
  k8scluster: test-cluster-east-1a
  backend: test1-nasbackend
autoExportPolicy: true
autoExportCIDRs:
- 10.0.0.0/24
username: admin
password: password
nfsMountOptions: nfsvers=4
```
IPv6 addresses example

This example shows managementLIF using an IPv6 address.

```yaml
---
version: 1
storageDriverName: ontap-nas
backendName: nas_ipv6_backend
managementLIF: "[5c5d:5edf:8f:7657:bef8:109b:1b41:d491]"
labels:
  k8scluster: test-cluster-east-1a
  backend: test1-ontap-ipv6
svm: nas_ipv6_svm
username: vsadmin
password: password
```

Amazon FSx for ONTAP using SMB volumes example

The smbShare parameter is required for FSx for ONTAP using SMB volumes.

```yaml
---
version: 1
backendName: SMBBackend
storageDriverName: ontap-nas
managementLIF: example.mgmt.fqdn.aws.com
nasType: smb
dataLIF: 10.0.0.15
svm: nfs_svm
smbShare: smb-share
clientCertificate: ZXR0ZXJwYXJhbD...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
```

Examples of backends with virtual pools

In the sample backend definition files shown below, 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.

Astra Trident sets provisioning labels in the "Comments" field. Comments are set on FlexVol for ontap-nas or FlexGroup for ontap-nas-flexgroup. Astra 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.
---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
svm: svm_nfs
username: admin
password: <password>
nfsMountOptions: nfsvers=4
defaults:
  spaceReserve: none
  encryption: 'false'
  qosPolicy: standard
labels:
  store: nas_store
  k8scluster: prod-cluster-1
region: us_east_1
storage:
- labels:
  app: msoffice
  cost: '100'
  zone: us_east_1a
  defaults:
    spaceReserve: volume
    encryption: 'true'
    unixPermissions: '0755'
    adaptiveQosPolicy: adaptive-premium
- labels:
  app: slack
  cost: '75'
  zone: us_east_1b
  defaults:
    spaceReserve: none
    encryption: 'true'
    unixPermissions: '0755'
- labels:
  department: legal
  creditpoints: '5000'
  zone: us_east_1b
  defaults:
    spaceReserve: none
    encryption: 'true'
    unixPermissions: '0755'
- labels:
  app: wordpress
cost: '50'
zone: us_east_1c
defaults:
  spaceReserve: none
  encryption: 'true'
  unixPermissions: '0775'
- labels:
  app: mysql
  cost: '25'
zone: us_east_1d
defaults:
  spaceReserve: volume
  encryption: 'false'
  unixPermissions: '0775'
---
version: 1
storageDriverName: ontap-nas-flexgroup
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: <password>
defaults:
  spaceReserve: none
  encryption: 'false'
labels:
  store: flexgroup_store
  k8scluster: prod-cluster-1
  region: us_east_1
storage:
- labels:
  protection: gold
  creditpoints: '50000'
  zone: us_east_1a
  defaults:
    spaceReserve: volume
    encryption: 'true'
    unixPermissions: '0755'
- labels:
  protection: gold
  creditpoints: '30000'
  zone: us_east_1b
  defaults:
    spaceReserve: none
    encryption: 'true'
    unixPermissions: '0755'
- labels:
  protection: silver
  creditpoints: '20000'
  zone: us_east_1c
  defaults:
    spaceReserve: none
    encryption: 'true'
    unixPermissions: '0775'
- labels:
  protection: bronze
  creditpoints: '10000'
  zone: us_east_1d
  defaults:
spaceReserve: volume
encryption: 'false'
unixPermissions: '0775'
ONTAP NAS economy example

---

version: 1
storageDriverName: ontap-nas-economy
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: <password>
defaults:
  spaceReserve: none
collection: 'false'
labels:
  store: nas_economy_store
region: us_east_1
storage:
  - labels:
      department: finance
creditpoints: '6000'
zone: us_east_1a
defaults:
  spaceReserve: volume
collection: 'true'
unixPermissions: '0755'
  - labels:
      protection: bronze
creditpoints: '5000'
zone: us_east_1b
defaults:
  spaceReserve: none
collection: 'true'
unixPermissions: '0755'
  - labels:
      department: engineering
creditpoints: '3000'
zone: us_east_1c
defaults:
  spaceReserve: none
collection: 'true'
unixPermissions: '0775'
  - labels:
      department: humanresource
creditpoints: '2000'
zone: us_east_1d
defaults:
  spaceReserve: volume
Map backends to StorageClasses

The following StorageClass definitions refer to 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 will have the aspects defined in the chosen virtual pool.

- The protection-gold StorageClass will map to the first and second virtual pool in the ontap-nas-flexgroup backend. These are the only pools offering gold level protection.

```yaml
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 will map to the third and fourth virtual pool in the ontap-nas-flexgroup backend. These are the only pools offering protection level other than gold.

```yaml
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-mysql StorageClass will map to the fourth virtual pool in the ontap-nas backend. This is the only pool offering storage pool configuration for mysql type app.
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysql
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysqldb"
fsType: "ext4"

- The **protection-silver-creditpoints-20k** StorageClass will map to the third virtual pool in the **ontap-nas-flexgroup** 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 will map to the third virtual pool in the **ontap-nas** backend and the second virtual pool in the **ontap-nas-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"

Astra Trident will decide which virtual pool is selected and will ensure the storage requirement is met.

**Update dataLIF after initial configuration**

You can change the data LIF after initial configuration by running the following command to provide the new backend JSON file with updated data LIF.
tridentctl update backend <backend-name> -f <path-to-backend-json-file-with-updated-dataLIF>

- If PVCs are attached to one or multiple pods, you must bring down all corresponding pods and then bring them back up in order for the new data LIF to take effect.