Concepts

Astra Trident

NetApp
March 18, 2022

Table of Contents

Concepts

Intro to Astra Trident

Astra Trident is a fully supported open source project maintained by NetApp as part of the Astra product family. It has been designed to help you meet your containerized applications' persistence demands using industry-standard interfaces, such as the Container Storage Interface (CSI).

Astra Trident deploys in Kubernetes clusters as pods and provides dynamic storage orchestration services for your Kubernetes workloads. It enables your containerized applications to quickly and easily consume persistent storage from NetApp's broad portfolio that includes ONTAP (AFF/FAS/Select/Cloud/Amazon FSx for NetApp ONTAP), Element software (NetApp HCI/SolidFire), Astra Data Store, as well as the Azure NetApp Files service, Cloud Volumes Service on Google Cloud, and the Cloud Volumes Service on AWS.

Astra Trident is also a foundational technology for NetApp's Astra, which addresses your data protection, disaster recovery, portability, and migration use cases for Kubernetes workloads leveraging NetApp's industry-leading data management technology for snapshots, backups, replication, and cloning.

Supported Kubernetes cluster architectures

Astra Trident is supported with the following Kubernetes architectures:

Kubernetes cluster architectures	Supported	Default install
Single master, compute	Yes	Yes
Multiple master, compute	Yes	Yes
Master, `etcd`, compute	Yes	Yes
Master, infrastructure, compute	Yes	Yes

What is Astra?

Astra makes it easier for enterprises to manage, protect, and move their data-rich containerized workloads running on Kubernetes within and across public clouds and on-premises. Astra provisions and provides persistent container storage using Astra Trident from NetApp's proven and expansive storage portfolio in the public cloud and on-premises. It also offers a rich set of advanced application-aware data management functionality, such as snapshot, backup and restore, activity logs, and active cloning for data protection, disaster/data recovery, data audit, and migration use-cases for Kubernetes workloads.

You can sign up for a free trial on the Astra page.

For more information

- NetApp Astra product family
- Astra Control Service documentation
- Astra Control Center documentation
- Astra Data Store documentation

* Astra API documentation

ONTAP drivers

Astra Trident provides five unique ONTAP storage drivers for communicating with ONTAP clusters. Learn more about how each driver handles the creation of volumes and access control and their capabilities.

Driver	Protocol	VolumeMode	Access modes supported	File systems supported
`ontap-nas`	NFS	Filesystem	RWO,RWX,ROX	"", nfs
`ontap-nas-economy`	NFS	Filesystem	RWO,RWX,ROX	"", nfs
`ontap-nas-flexgroup`	NFS	Filesystem	RWO,RWX,ROX	"", nfs
`ontap-san`	iSCSI	Block	RWO,ROX,RWX	No Filesystem. Raw block device
`ontap-san`	iSCSI	Filesystem	RWO,ROX	`xfs`, `ext3`, `ext4`
`ontap-san-economy`	iSCSI	Block	RWO,ROX,RWX	No Filesystem. Raw block device
`ontap-san-economy`	iSCSI	Filesystem	RWO,ROX	`xfs`, `ext3`, `ext4`

ONTAP backends can be authenticated by using login credentials for a security role (username/password) or using the private key and the certificate that is installed on the ONTAP cluster. You can update existing backends to move from one authentication mode to the other with `tridentctl update backend`.

Provisioning

Provisioning in Astra Trident has two primary phases. The first phase associates a storage class with the set of suitable backend storage pools and occurs as a necessary preparation before provisioning. The second phase includes the volume creation itself and requires choosing a storage pool from those associated with the pending volume’s storage class.

Associating backend storage pools with a storage class relies on both the storage class’s requested attributes and its `storagePools`, `additionalStoragePools`, and `excludeStoragePools` lists. When you create a storage class, Trident compares the attributes and pools offered by each of its backends to those requested by the storage class. If a storage pool’s attributes and name match all of the requested attributes and pool names, Astra Trident adds that storage pool to the set of suitable storage pools for that storage class. In addition, Astra Trident adds all storage pools listed in the `additionalStoragePools` list to that set, even if their attributes do not fulfill all or any of the storage class’s requested attributes. You should use the `excludeStoragePools` list to override and remove storage pools from use for a storage class. Astra Trident performs a similar process every time you add a new backend, checking whether its storage pools satisfy those of the existing storage classes and removing any that have been marked as excluded.

Astra Trident then uses the associations between storage classes and storage pools to determine where to provision volumes. When you create a volume, Astra Trident first gets the set of storage pools for that volume's storage class, and, if you specify a protocol for the volume, Astra Trident removes those storage pools that cannot provide the requested protocol (for example, a NetApp HCI/SolidFire backend cannot provide a file-based volume while an ONTAP NAS backend cannot provide a block-based volume). Astra Trident randomizes the order of this resulting set, to facilitate an even distribution of volumes, and then iterates through it, attempting to provision the volume on each storage pool in turn. If it succeeds on one, it returns successfully, logging any failures encountered in the process. Astra Trident returns a failure **only if** it fails to provision on **all** the storage pools available for the requested storage class and protocol.

Volume snapshots

Learn more about how Astra Trident handles the creation of volume snapshots for its drivers.

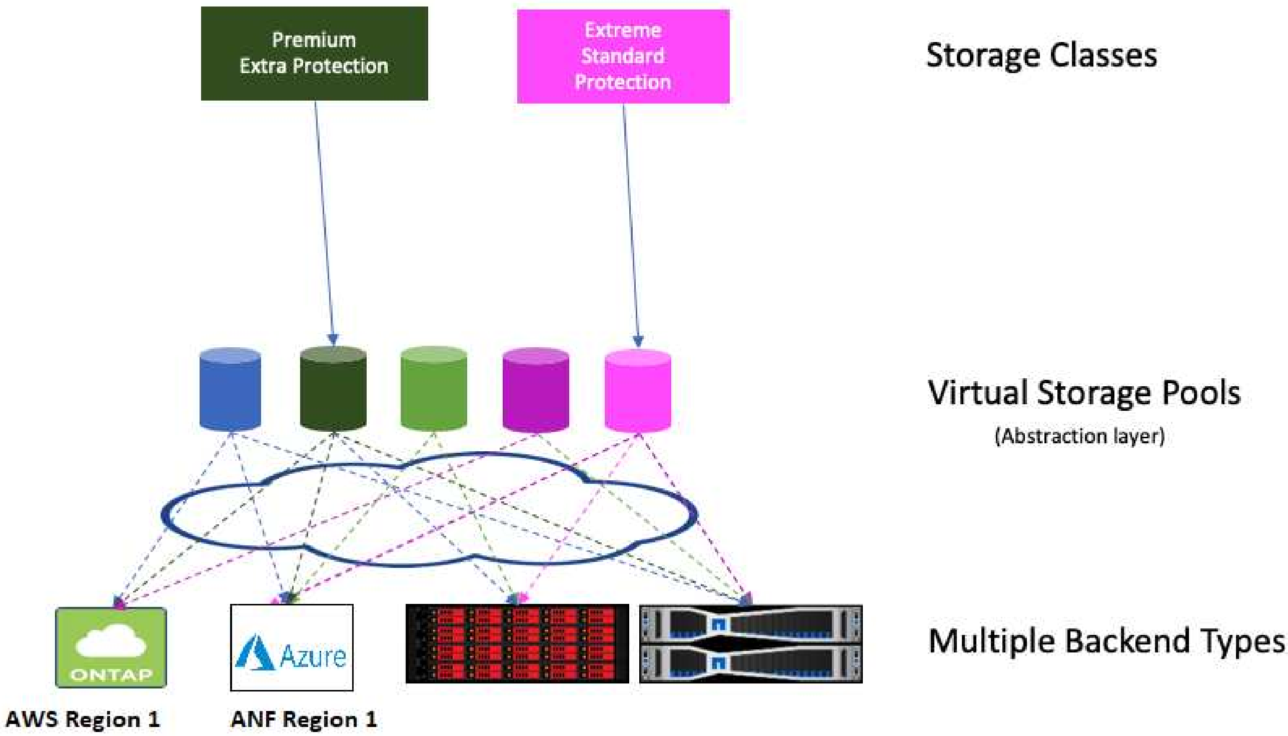
- For the `ontap-nas`, `ontap-san`, `aws-cvs`, `gcp-cvs`, and `azure-netapp-files` drivers, each Persistent Volume (PV) maps to a FlexVol. As a result, volume snapshots are created as NetApp snapshots. NetApp's snapshot technology delivers more stability, scalability, recoverability, and performance than competing snapshot technologies. These snapshot copies are extremely efficient both in the time needed to create them and in storage space.
- For the `ontap-nas-flexgroup` driver, each Persistent Volume (PV) maps to a FlexGroup. As a result, volume snapshots are created as NetApp FlexGroup snapshots. NetApp's snapshot technology delivers more stability, scalability, recoverability, and performance than competing snapshot technologies. These snapshot copies are extremely efficient both in the time needed to create them and in storage space.
- For the `ontap-san-economy` driver, PVs map to LUNs created on shared FlexVols. VolumeSnapshots of PVs are achieved by performing FlexClones of the associated LUN. ONTAP's FlexClone technology makes it possible to create copies of even the largest datasets almost instantaneously. Copies share data blocks with their parents, consuming no storage except what is required for metadata.
- For the `solidfire-san` driver, each PV maps to a LUN created on the NetApp Element software/NetApp HCI cluster. VolumeSnapshots are represented by Element snapshots of the underlying LUN. These snapshots are point-in-time copies and only take up a small amount of system resources and space.
- When working with the `ontap-nas` and `ontap-san` drivers, ONTAP snapshots are point-in-time copies of the FlexVol and consume space on the FlexVol itself. This can result in the amount of writable space in the volume to reduce with time as snapshots are created/scheduled. One simple way of addressing this is to grow the volume by resizing through Kubernetes. Another option is to delete snapshots that are no longer required. When a VolumeSnapshot created through Kubernetes is deleted, Astra Trident will delete the associated ONTAP snapshot. ONTAP snapshots that were not created through Kubernetes can also be deleted.

With Astra Trident, you can use VolumeSnapshots to create new PVs from them. Creating PVs from these snapshots is performed by using the FlexClone technology for supported ONTAP and CVS backends. When creating a PV from a snapshot, the backing volume is a FlexClone of the snapshot's parent volume. The `solidfire-san` driver uses Element software volume clones to create PVs from snapshots. Here it creates a clone from the Element snapshot.

Virtual storage pools

Virtual storage pools provide a layer of abstraction between Astra Trident's storage backends and Kubernetes' `StorageClasses`. They allow an administrator to define aspects, such as location, performance, and protection for each backend in a common, backend-agnostic way without making a `StorageClass` specify which physical backend, backend pool, or backend type to use to meet desired criteria.

The storage administrator can define virtual storage pools on any of the Astra Trident backends in a JSON or YAML definition file.



Any aspect specified outside the virtual pools list is global to the backend and will apply to all the virtual pools, while each virtual pool might specify one or more aspects individually (overriding any backend-global aspects).

> When defining virtual storage pools, do not attempt to rearrange the order of existing virtual pools in a backend definition.
> It is also advisable to not edit/modify attributes for an existing virtual pool and define a new virtual pool instead.

Most aspects are specified in backend-specific terms. Crucially, the aspect values are not exposed outside the backend's driver and are not available for matching in `StorageClasses`. Instead, the administrator defines one or more labels for each virtual pool. Each label is a key:value pair, and labels might be common across unique backends. Like aspects, labels can be specified per-pool or global to the backend. Unlike aspects, which have predefined names and values, the administrator has full discretion to define label keys and values as needed.

A `StorageClass` identifies which virtual pool to use by referencing the labels within a selector parameter. Virtual pool selectors support six operators:

Operator	Example	Description
`=`	performance=premium	A pool's label value must match
`!=`	performance!=extreme	A pool's label value must not match
`in`	location in (east, west)	A pool's label value must be in the set of values

Operator	Example	Description
`notin`	performance notin (silver, bronze)	A pool’s label value must not b

Volume access groups

Learn more about how Astra Trident uses volume access groups.

> Ignore this section if you are using CHAP, which is recommended to simplify management and avoid the scaling limit described below. In addition, if you are using Astra Trident in CSI mode, you can ignore this section. Astra Trident uses CHAP when installed as an enhanced CSI provisioner.

Astra Trident can use volume access groups to control access to the volumes that it provisions. If CHAP is disabled, it expects to find an access group called `trident` unless you specify one or more access group IDs in the configuration.

While Astra Trident associates new volumes with the configured access group(s), it does not create or otherwise manage access groups themselves. The access group(s) must exist before the storage backend is added to Astra Trident, and they need to contain the iSCSI IQNs from every node in the Kubernetes cluster that could potentially mount the volumes provisioned by that backend. In most installations, that includes every worker node in the cluster.

For Kubernetes clusters with more than 64 nodes, you should use multiple access groups. Each access group may contain up to 64 IQNs, and each volume can belong to four access groups. With the maximum four access groups configured, any node in a cluster up to 256 nodes in size will be able to access any volume. For latest limits on volume access groups, see here.

If you’re modifying the configuration from one that is using the default `trident` access group to one that uses others as well, include the ID for the `trident` access group in the list.