



Los geht's

Astra Trident

NetApp
January 14, 2026

Inhalt

- Los geht's 1
 - Erfahren Sie mehr über Astra Trident 1
 - Erfahren Sie mehr über Astra Trident 1
 - Die Architektur von Astra Trident 3
 - Konzepte 5
 - Der Einstieg in Astra Trident ist schnell möglich 8
 - Was kommt als Nächstes? 9
- Anforderungen 10
 - Kritische Informationen zu Astra Trident 10
 - Unterstützte Frontends (Orchestrators) 10
 - Unterstützte Back-Ends (Storage) 10
 - Anforderungen an die Funktionen 11
 - Getestete Host-Betriebssysteme 11
 - Host-Konfiguration 12
 - Konfiguration des Storage-Systems 12
 - Astra Trident-Ports 12
 - Container-Images und entsprechende Kubernetes-Versionen 12

Los geht's

Erfahren Sie mehr über Astra Trident

Erfahren Sie mehr über Astra Trident

Astra Trident ist ein vollständig unterstütztes Open-Source-Projekt, das von NetApp als Teil der gepflegt wird ["Astra Produktfamilie"](#). Es wurde entwickelt, damit Sie die Persistenz-Anforderungen Ihrer Container-Applikation mithilfe von Standardschnittstellen, wie dem Container Storage Interface (CSI), erfüllen können.

Was ist Astra?

Astra erleichtert Unternehmen das Management, die Sicherung und das Verschieben ihrer datenintensiven Container-Workloads, die auf Kubernetes ausgeführt werden, innerhalb der Public Cloud und vor Ort.

Astra stellt persistenten Container-Storage auf Basis von Astra Trident bereit und bietet diese an. Es bietet außerdem erweiterte applikationsgerechte Datenmanagement-Funktionen wie Snapshot, Backup und Restore, Aktivitätsprotokolle und aktives Klonen für Datensicherung, Disaster/Daten-Recovery, Datenaudit und Migrationsanwendungsfälle für Kubernetes-Workloads.

Weitere Informationen zu ["Astra oder melden Sie sich für die kostenlose Testversion an"](#).

Was ist Astra Trident?

Astra Trident ermöglicht die Nutzung und das Management von Storage-Ressourcen über alle gängigen NetApp Storage-Plattformen hinweg, in der Public Cloud oder lokal, einschließlich ONTAP (AFF, FAS, Select, Cloud, Amazon FSX for NetApp ONTAP), Element Software (NetApp HCI, SolidFire), Azure NetApp Files Service und Cloud Volumes Service auf Google Cloud.

Astra Trident ist ein CSI-konformer dynamischer Storage Orchestrator, der sich nativ integrieren lässt ["Kubernetes"](#). Astra Trident wird als einzelner Controller Pod plus Node Pod auf jedem Worker-Node im Cluster ausgeführt. Siehe ["Die Architektur von Astra Trident"](#) Entsprechende Details.

Astra Trident bietet zudem eine direkte Integration in das Docker Ecosystem für NetApp Storage-Plattformen. Das NetApp Docker Volume Plug-in (nDVP) unterstützt die Bereitstellung und das Management von Storage-Ressourcen von der Storage-Plattform an Docker Hosts. Siehe ["Implementieren Sie Astra Trident für Docker"](#) Entsprechende Details.



Wenn Sie Kubernetes zum ersten Mal verwenden, sollten Sie sich mit dem vertraut machen ["Kubernetes-Konzepte und -Tools"](#).

Machen Sie einen Testlauf mit Astra Trident

Fordern Sie für einen Testlauf den Zugriff auf den „persistenten Storage für Container-Workloads einfach implementieren und klonen“ an. ["NetApp Testversion"](#) Verwenden eines gebrauchsfertigen Laborabbaus. Testlauf bietet eine Sandbox-Umgebung mit einem Kubernetes-Cluster mit drei Nodes und Astra Trident ist installiert und konfiguriert. So können Sie sich besser mit Astra Trident vertraut machen und die zugehörigen Funktionen erkunden.

Eine weitere Option ist die ["Installationsanleitung für kubeadm"](#) Von Kubernetes bereitgestellt.



Verwenden Sie in einer Produktionsumgebung keine Kubernetes-Cluster, die Sie mit diesen Anweisungen erstellen. Nutzen Sie die von Ihrer Distribution bereitgestellten Leitfäden zur Implementierung in Produktionsumgebungen für Cluster.

Kubernetes-Integration in NetApp Produkte

Das NetApp Portfolio an Storage-Produkten kann in viele Aspekte eines Kubernetes Clusters integriert werden und bietet erweiterte Datenmanagement-Funktionen, mit denen die Funktionalität, Funktionalität, Performance und Verfügbarkeit der Kubernetes-Implementierung verbessert werden.

Amazon FSX für NetApp ONTAP

"[Amazon FSX für NetApp ONTAP](#)" Ist ein vollständig gemanagter AWS Service, mit dem Sie Dateisysteme mit dem NetApp ONTAP Storage-Betriebssystem starten und ausführen können.

Azure NetApp Dateien

"[Azure NetApp Dateien](#)" Ist ein Azure-Dateifreigabeservice der Enterprise-Klasse auf der Basis von NetApp. Sie können anspruchsvollste dateibasierte Workloads nativ in Azure ausführen. So erhalten Sie die Performance und das umfassende Datenmanagement, die Sie von NetApp gewohnt sind.

Cloud Volumes ONTAP

"[Cloud Volumes ONTAP](#)" Ist eine rein softwarebasierte Storage Appliance, die die ONTAP Datenmanagement-Software in der Cloud ausführt.

Cloud Volumes Service für Google Cloud

"[NetApp Cloud Volumes Service für Google Cloud](#)" Ist ein Cloud-nativer Fileservice, der NAS-Volumes über NFS und SMB mit All-Flash-Performance bereitstellt.

Element Software

"[Element](#)" Storage-Administrator kann Workloads konsolidieren, indem die Performance garantiert und der Storage-Bedarf vereinfacht und optimiert wird.

NetApp HCI

"[NetApp HCI](#)" Vereinfacht das Management und die Skalierung des Datacenters durch Automatisierung von Routineaufgaben und ermöglicht es Infrastrukturadministratoren, sich auf wichtigere Funktionen zu konzentrieren.

Astra Trident kann Storage-Geräte für Container-Applikationen direkt auf der zugrunde liegenden NetApp HCI Storage-Plattform bereitstellen und managen.

NetApp ONTAP

"[NetApp ONTAP](#)" Ist das Unified Storage-Betriebssystem NetApp für mehrere Protokolle und bietet für jede Applikation erweiterte Datenmanagementfunktionen.

ONTAP Systeme verfügen über rein Flash-basierte, hybride oder rein HDD-basierte Konfigurationen und bieten eine Vielzahl unterschiedlicher Implementierungsmodelle, darunter speziell entwickelte Hardware (FAS und AFF), White-Box (ONTAP Select) und rein Cloud-basierte Cloud Volumes ONTAP Systeme. Astra Trident unterstützt diese ONTAP Implementierungsmodelle.

Finden Sie weitere Informationen

- "[Die NetApp Astra-Produktfamilie](#)"
- "[Dokumentation des Astra Control Service](#)"
- "[Astra Control Center-Dokumentation](#)"
- "[Astra API-Dokumentation](#)"

Die Architektur von Astra Trident

Astra Trident wird als einzelner Controller Pod plus Node Pod auf jedem Worker-Node im Cluster ausgeführt. Der Node Pod muss auf jedem Host ausgeführt werden, auf dem Sie ein Astra Trident Volume mounten möchten.

Allgemeines zu Controller-Pods und Node-Pods

Astra Trident wird einzeln implementiert [Trident Controller Pod](#) Und eines oder mehrere [Trident Node Pods](#) Auf dem Kubernetes-Cluster und verwendet standardmäßige Kubernetes *CSI Sidecar Container*, um die Implementierung von CSI-Plug-ins zu vereinfachen. "[Kubernetes CSI Sidecar-Container](#)" Werden von der Kubernetes Storage Community unterhalten.

Kubernetes "[Knotenauswahl](#)" Und "[Toleranzen und Verfleckungen](#)" Werden verwendet, um die Ausführung eines Pod auf einem bestimmten oder bevorzugten Node einzuschränken. Während der Astra Trident Installation können Node-Selektoren und Toleranzen für Controller- und Node-Pods konfiguriert werden.

- Das Controller-Plug-in übernimmt Volume-Bereitstellung und -Management, beispielsweise Snapshots und Größenanpassungen.
- Das Node-Plug-in verarbeitet das Verbinden des Speichers mit dem Node.

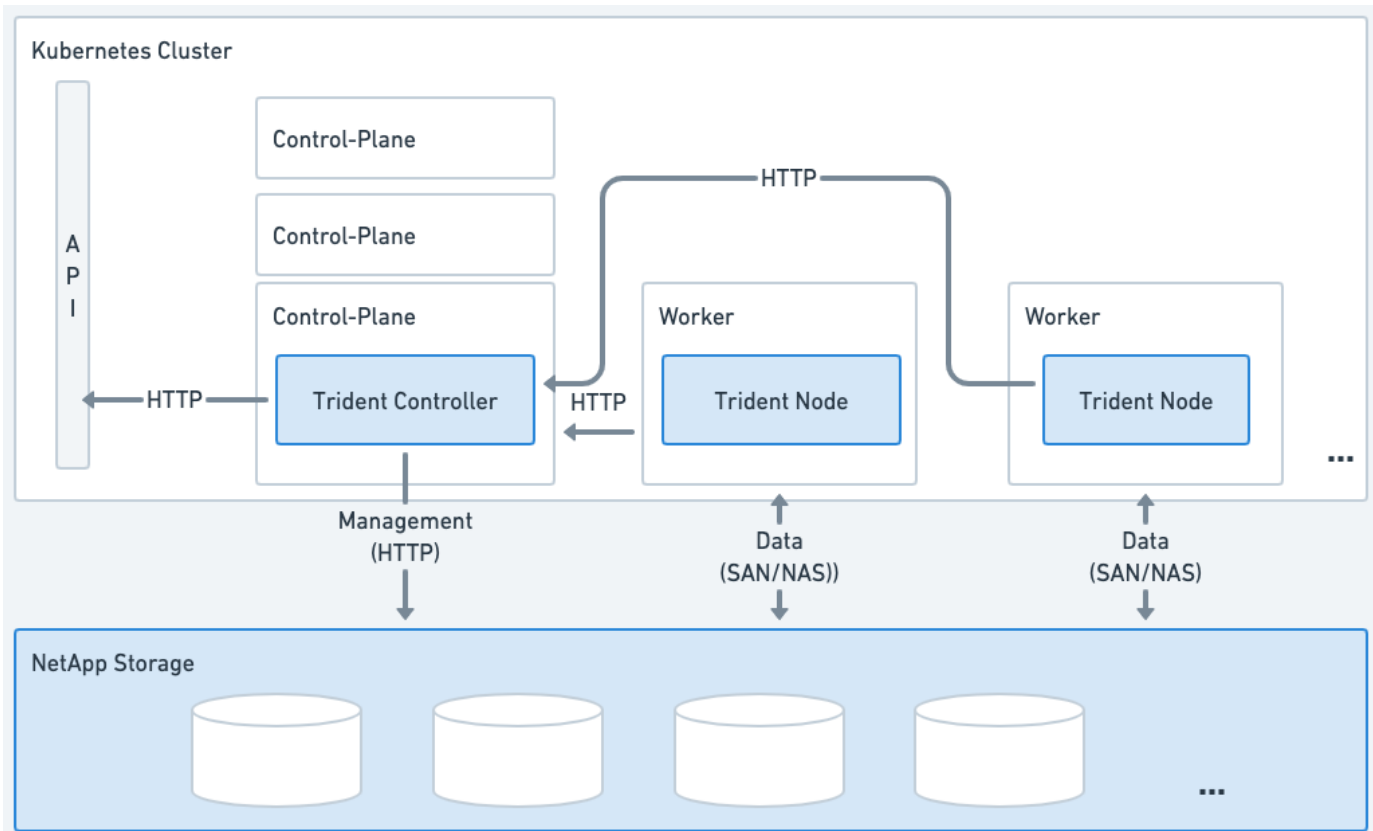


Abbildung 1. Astra Trident wird auf dem Kubernetes-Cluster implementiert

Trident Controller Pod

Beim Trident Controller Pod handelt es sich um einen einzelnen Pod, auf dem das CSI Controller Plug-in ausgeführt wird.

- Verantwortlich für die Bereitstellung und das Management von Volumes in NetApp Storage
- Management durch eine Kubernetes-Implementierung
- Kann je nach Installationsparameter auf der Steuerebene oder auf den Arbeitsknoten ausgeführt werden.

Abbildung 2. Trident Controller Pod-Diagramm

Trident Node Pods

Trident Node Pods sind privilegierte Pods, auf denen das CSI Node Plug-in ausgeführt wird.

- Verantwortlich für das Mounten und Entmounten von Speicher für Pods, die auf dem Host ausgeführt werden
- Gemanagt von einem Kubernetes DaemonSet
- Muss auf jedem Node ausgeführt werden, auf dem NetApp Storage gemountet werden soll

Abbildung 3. Trident Node Pod-Diagramm

Unterstützte Kubernetes-Cluster-Architekturen

Astra Trident wird durch die folgenden Kubernetes-Architekturen unterstützt:

Kubernetes-Cluster-Architekturen	Unterstützt	Standardinstallation
Ein Master Computing	Ja.	Ja.
Mehrere Master-Computer und Computing-Ressourcen	Ja.	Ja.
Master, etcd, Datenverarbeitung	Ja.	Ja.
Master, Infrastruktur, Computing	Ja.	Ja.

Konzepte

Bereitstellung

Die Bereitstellung in Astra Trident besteht aus zwei Hauptphasen. In der ersten Phase wird eine Speicherklasse mit einem Satz geeigneter Back-End-Speicherpools verknüpft. Diese werden vor der Bereitstellung als notwendig vorbereitet. Die zweite Phase umfasst die Volume-Erstellung selbst und erfordert die Auswahl eines Speicherpools aus denen, die mit der Storage-Klasse des ausstehenden Volumes verknüpft sind.

Storage-Klassen-Zuordnung

Das Zuordnen von Back-End Storage-Pools zu einer Storage-Klasse hängt sowohl von den angeforderten Attributen der Storage-Klasse als auch von deren `storagePools`, `additionalStoragePools`, und `excludeStoragePools` Listen. Wenn Sie eine Storage-Klasse erstellen, vergleicht Trident die von jedem seiner Back-Ends angebotenen Attribute und Pools mit den von der Storage-Klasse angeforderten Attributen. Wenn die Attribute und der Name eines Storage Pools mit allen angeforderten Attributen und Pool-Namen übereinstimmen, fügt Astra Trident diesem Satz an geeigneten Storage-Pools für diese Storage-Klasse hinzu. Außerdem fügt Astra Trident alle im aufgeführten Storage-Pools hinzu `additionalStoragePools` Listen Sie zu diesem Satz auf, auch wenn seine Attribute nicht alle oder eines der angeforderten Attribute der Storage-Klasse erfüllen. Sie sollten das verwenden `excludeStoragePools` Liste zum Überschreiben und Entfernen von Speicherpools, die für eine Speicherklasse verwendet werden. Astra Trident führt jedes Mal einen ähnlichen Prozess durch, wenn Sie ein neues Back-End hinzufügen. Er überprüft, ob die Storage Pools die Anforderungen der vorhandenen Storage-Klassen erfüllen und entfernt alle, die als ausgeschlossen markiert wurden.

Volume-Erstellung

Astra Trident verwendet dann die Zuordnungen zwischen Storage-Klassen und Storage-Pools, um zu bestimmen, wo Volumes bereitgestellt werden sollen. Wenn Sie ein Volume erstellen, erhält Astra Trident zunächst die Reihe von Storage-Pools für dieses Volume in der Storage-Klasse. Wenn Sie ein Protokoll für das Volume angeben, entfernt Astra Trident die Storage-Pools, die das angeforderte Protokoll nicht bereitstellen können (beispielsweise kann ein NetApp HCI/SolidFire Backend kein dateibasiertes Volume bereitstellen, während ein ONTAP NAS-Backend kein blockbasiertes Volume bereitstellen kann). Astra Trident randomisiert die Reihenfolge dieser daraus resultierenden Sets, um eine gleichmäßige Verteilung der Volumes zu ermöglichen und es anschließend zu iterieren und dabei zu versuchen, das Volume wiederum auf jedem Storage-Pool bereitzustellen. Wenn sie erfolgreich ist, wird sie erfolgreich zurückgegeben, und es werden alle Fehler protokolliert, die im Prozess aufgetreten sind. Astra Trident gibt einen Fehler zurück **nur wenn** sie nicht auf allen * den Storage Pools zur Verfügung steht für die angeforderte Storage-Klasse und das gewünschte Protokoll.

Volume Snapshots

Erfahren Sie mehr darüber, wie Astra Trident die Erstellung von Volume-Snapshots für seine Treiber steuert.

Erfahren Sie mehr über die Erstellung von Volume Snapshots

- Für das `ontap-nas`, `ontap-san`, `gcp-cvs`, und `azure-netapp-files` Treiber, wird jedes Persistent Volume (PV) einer FlexVol zugeordnet. Volume Snapshots werden im Ergebnis als NetApp Snapshots erstellt. NetApp Snapshots liefern weitaus mehr Stabilität, Skalierbarkeit, Wiederherstellbarkeit und Performance als vergleichbare Systeme. Diese Snapshot-Kopien sind äußerst schnell und platzsparend, da sie erstellt und gespeichert werden müssen.
- Für das `ontap-nas-flexgroup` Treiber: Jedes Persistent Volume (PV) ist einem FlexGroup zugeordnet. Im Ergebnis werden Volume Snapshots als NetApp FlexGroup Snapshots erstellt. NetApp Snapshots liefern weitaus mehr Stabilität, Skalierbarkeit, Wiederherstellbarkeit und Performance als vergleichbare Systeme. Diese Snapshot-Kopien sind äußerst schnell und platzsparend, da sie erstellt und gespeichert werden müssen.
- Für das `ontap-san-economy` Treiber, PVS werden LUNs zugeordnet, die auf gemeinsam genutzten FlexVols erstellt wurden. VolumeSnapshots von PVS werden durch FlexClones der zugehörigen LUN erreicht. Mit der ONTAP FlexClone Technologie ist es nahezu sofort möglich, Kopien selbst von größten Datensätzen zu erstellen. Kopien nutzen Datenblöcke gemeinsam mit ihren Eltern und verbrauchen somit keinen Storage, außer was für Metadaten erforderlich ist.
- Für das `solidfire-san` Treiber: Jedes PV wird einer auf der NetApp Element Software/dem NetApp HCI Cluster erstellten LUN zugeordnet. VolumeSnapshots werden durch Element Snapshots der zugrunde liegenden LUN dargestellt. Diese Snapshots sind zeitpunktgenaue Kopien, die nur eine kleine Menge an Systemressourcen und Platz beanspruchen.
- Bei der Arbeit mit dem `ontap-nas` Und `ontap-san` Treiber, ONTAP Snapshots sind zeitpunktgenaue Kopien der FlexVol und verbrauchen Platz auf der FlexVol selbst. Das kann dazu führen, dass der beschreibbare Speicherplatz auf dem Volume mit der Zeit verkürzt wird, wenn Snapshots erstellt/geplant werden. Eine einfache Möglichkeit dieser Bewältigung ist, das Volumen durch die Anpassung über Kubernetes zu vergrößern. Eine weitere Option ist das Löschen von nicht mehr benötigten Snapshots. Wenn ein über Kubernetes erstellter VolumeSnapshot gelöscht wird, löscht Astra Trident den zugehörigen ONTAP-Snapshot. ONTAP Snapshots, die nicht über Kubernetes erstellt wurden, können auch gelöscht werden.

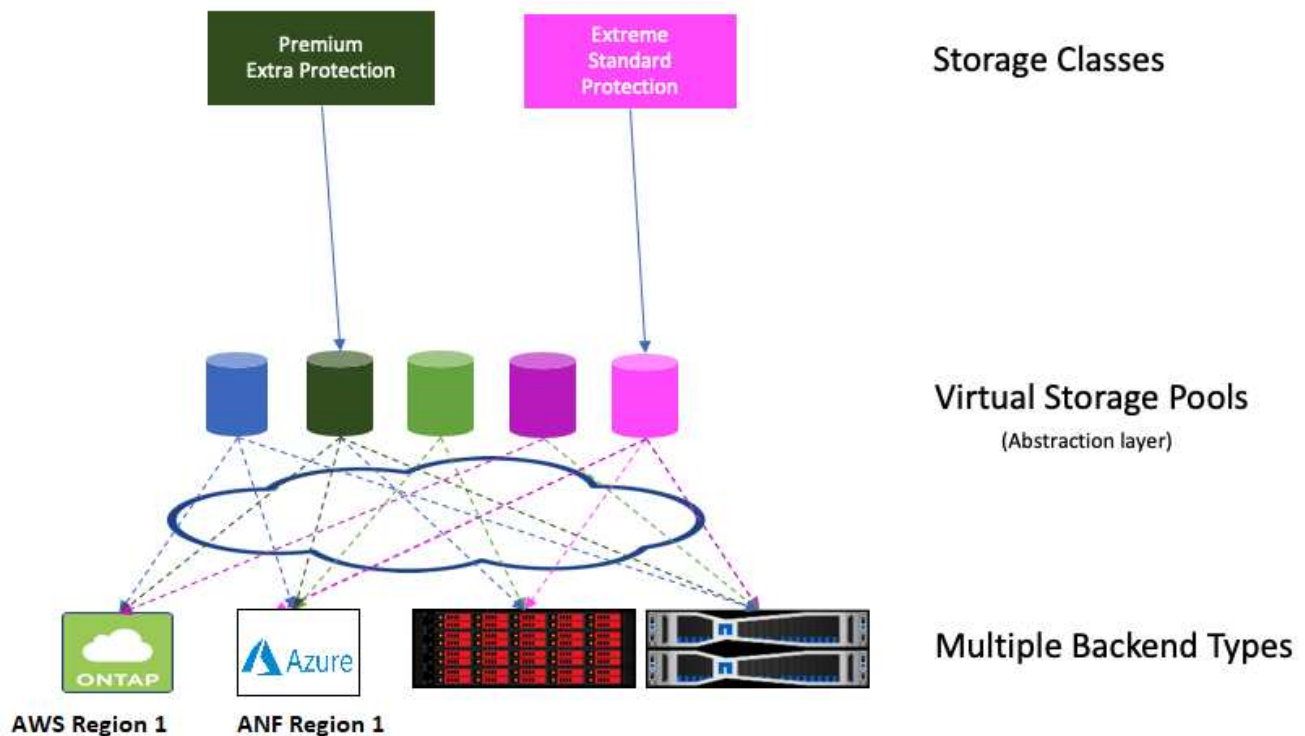
Mit Astra Trident können Sie VolumeSnapshot verwenden, um neue PVS daraus zu erstellen. Die Erstellung von PVS aus diesen Snapshots wird mithilfe der FlexClone Technologie für unterstützte ONTAP- und CVS-Back-Ends durchgeführt. Wenn ein PV aus einem Snapshot erstellt wird, ist das Back-Volume ein FlexClone des übergeordneten Volume des Snapshots. Der `solidfire-san` Der Treiber verwendet Volume Clones der Element Software, um PVS aus Snapshots zu erstellen. Hier erstellt es aus dem Element Snapshot einen Klon.

Virtuelle Pools

Virtuelle Pools stellen eine Abstraktionsschicht zwischen Astra Trident Storage-Back-Ends und Kubernetes bereit `StorageClasses`. Sie ermöglichen es Administratoren, für jedes Back-End-System Aspekte wie Standort, Performance und Schutz zu definieren, ohne dafür eine `StorageClass` Legen Sie fest, welches physische Backend-, Backend-Pool- oder Backend-Typ für die gewünschten Kriterien verwendet werden soll.

Erfahren Sie mehr über virtuelle Pools

Der Storage-Administrator kann virtuelle Pools auf einem beliebigen Astra Trident Back-End in einer JSON- oder YAML-Definitionsdatei definieren.



Jeder außerhalb der Liste der virtuellen Pools angegebene Aspekt ist global für das Backend und gilt für alle virtuellen Pools, während jeder virtuelle Pool einen oder mehrere Aspekte einzeln angeben kann (alle Backend-globalen Aspekte außer Kraft setzen).



- Versuchen Sie beim Definieren virtueller Pools nicht, die Reihenfolge vorhandener virtueller Pools in einer Backend-Definition neu anzuordnen.
- Wir empfehlen, Attribute für einen vorhandenen virtuellen Pool zu ändern. Sie sollten einen neuen virtuellen Pool definieren, um Änderungen vorzunehmen.

Die meisten Aspekte werden Backend-spezifisch angegeben. Entscheidend ist, dass die Aspect-Werte nicht außerhalb des Back-End-Treibers angezeigt werden und nicht für die Abstimmung in verfügbar sind `StorageClasses`. Stattdessen definiert der Administrator eine oder mehrere Labels für jeden virtuellen Pool. Jedes Etikett ist ein Schlüssel:Wert-Paar, und Etiketten können häufig über eindeutige Back-Ends hinweg verwendet werden. Wie Aspekte können auch Labels pro Pool oder global zum Backend angegeben werden. Im Gegensatz zu Aspekten, die vordefinierte Namen und Werte haben, hat der Administrator volle Entscheidungsbefugnis, Beschriftungsschlüssel und -Werte nach Bedarf zu definieren. Storage-Administratoren können Labels je virtuellen Pool definieren und Volumes nach Label gruppieren.

A `StorageClass` identifiziert den virtuellen Pool, der verwendet werden soll, indem auf die Beschriftungen in einem Auswahlparameter Bezug gesetzt wird. Virtuelle Pool-Selektoren unterstützen folgende Operatoren:

Operator	Beispiel	Der Wert für die Bezeichnung eines Pools muss:
=	Performance=Premium	Übereinstimmung
!=	Performance!=extrem	Keine Übereinstimmung
in	Lage in (Osten, Westen)	Werden Sie im Satz von Werten
notin	Performance-Dose (Silber, Bronze)	Nicht im Wertungsset sein
<key>	Darstellt	Mit einem beliebigen Wert existieren
!<key>	!Schutz	Nicht vorhanden

Volume-Zugriffsgruppen

Erfahren Sie mehr über die Einsatzmöglichkeiten von Astra Trident "[Volume-Zugriffsgruppen](#)".



Ignorieren Sie diesen Abschnitt, wenn Sie CHAP verwenden. Dies wird empfohlen, um die Verwaltung zu vereinfachen und die unten beschriebene Skalierungsgrenze zu vermeiden. Wenn Sie Astra Trident im CSI-Modus verwenden, können Sie diesen Abschnitt ignorieren. Astra Trident verwendet CHAP, wenn es als erweiterte CSI-bereitstellung installiert ist.

Erfahren Sie mehr über Volume Access Groups

Astra Trident kann über Volume-Zugriffsgruppen den Zugriff auf die Volumes steuern, die es bereitstellt. Wenn CHAP deaktiviert ist, wird erwartet, dass eine Zugriffsgruppe mit dem Namen gefunden wird `trident`. Es sei denn, Sie geben eine oder mehrere Zugriffsgruppen-IDs in der Konfiguration an.

Astra Trident ordnet den konfigurierten Zugriffsgruppen neue Volumes zu, allerdings werden dafür keine Zugriffsgruppen selbst erstellt oder anderweitig gemanagt. Die Zugriffsgruppen müssen vorhanden sein, bevor das Storage-Back-End zu Astra Trident hinzugefügt wird. Sie müssen die iSCSI-IQNs von jedem Node im Kubernetes-Cluster enthalten, der die über dieses Back-End bereitgestellten Volumes mounten kann. In den meisten Installationen umfasst dies alle Worker Nodes im Cluster.

Bei Kubernetes-Clustern mit mehr als 64 Nodes sollten Sie mehrere Zugriffsgruppen verwenden. Jede Zugriffsgruppe kann bis zu 64 IQNs enthalten, und jedes Volume kann zu vier Zugriffsgruppen gehören. Bei maximal vier Zugriffsgruppen kann jeder Node in einem Cluster mit einer Größe von bis zu 256 Nodes auf beliebige Volumes zugreifen. Aktuelle Grenzwerte für Volume-Zugriffsgruppen finden Sie unter "[Hier](#)".

Wenn Sie die Konfiguration von einer Konfiguration ändern, die den Standard verwendet `trident`, geben Sie die ID für die Zugriffsgruppe für eine Gruppe, die auch andere verwendet, geben Sie die ID für die ein `trident` Zugriffsgruppe in der Liste.

Der Einstieg in Astra Trident ist schnell möglich

Sie können Astra Trident installieren und mit dem Management von Storage-Ressourcen in wenigen Schritten beginnen. Bevor Sie beginnen, überprüfen Sie "[Anforderungen von Astra Trident](#)".



Informationen zu Docker finden Sie unter "[Astra Trident für Docker](#)".

1

Installation Von Astra Trident

Astra Trident bietet mehrere Installationsmethoden und -Modi, die für eine Vielzahl von Umgebungen und Organisationen optimiert sind.

["Installation Von Astra Trident"](#)

2

Bereiten Sie den Knoten „Worker“ vor

Alle Worker-Nodes im Kubernetes-Cluster müssen in der Lage sein, die Volumes, die Sie für Ihre Pods bereitgestellt haben, zu mounten.

["Bereiten Sie den Knoten „Worker“ vor"](#)

3

Erstellen Sie ein Backend

Ein Backend definiert die Beziehung zwischen Astra Trident und einem Storage-System. Er erzählt Astra Trident, wie man mit diesem Storage-System kommuniziert und wie Astra Trident Volumes darauf bereitstellen sollte.

["Konfigurieren Sie ein Backend"](#) Für Ihr Speichersystem

4

Kubernetes StorageClass erstellen

Das Objekt Kubernetes StorageClass gibt Astra Trident als bereitstellung an und ermöglicht die Erstellung einer Storage-Klasse zur Bereitstellung von Volumes mit anpassbaren Attributen. Astra Trident erstellt eine passende Storage-Klasse für Kubernetes-Objekte, mit der der Astra Trident-bereitstellung angegeben wird.

["Erstellen Sie eine Speicherklasse"](#)

5

Bereitstellen eines Volumes

Ein *PersistentVolume* (PV) ist eine physische Speicherressource, die vom Cluster-Administrator auf einem Kubernetes-Cluster bereitgestellt wird. Das *PersistentVolumeClaim* (PVC) ist eine Anforderung für den Zugriff auf das PersistentVolume auf dem Cluster.

Erstellen Sie ein PersistentVolume (PV) und ein PersistentVolumeClaim (PVC), das die konfigurierte Kubernetes StorageClass verwendet, um Zugriff auf das PV anzufordern. Anschließend können Sie das PV an einem Pod montieren.

["Bereitstellen eines Volumes"](#)

Was kommt als Nächstes?

Sie können nun zusätzliche Back-Ends hinzufügen, Storage-Klassen managen, Back-Ends managen und Volume-Operationen durchführen.

Anforderungen

Vor der Installation von Astra Trident sollten Sie diese allgemeinen Systemanforderungen überprüfen. Spezifische Back-Ends können zusätzliche Anforderungen haben.

Kritische Informationen zu Astra Trident

Sie müssen die folgenden wichtigen Informationen über Astra Trident lesen.

Informationen über Astra TriperelT

- Kubernetes 1.28 wird jetzt in Trident unterstützt. Aktualisieren Sie Astra Trident vor dem Upgrade von Kubernetes.
- Astra Trident setzt die Verwendung von Multipathing-Konfiguration in SAN-Umgebungen strikt um und empfiehlt den Nutzen von `find_multipaths: no` In Multipath.conf Datei.

Verwendung einer Konfiguration ohne Multipathing oder Verwendung von `find_multipaths: yes` Oder `find_multipaths: smart` Der Wert in der Multipath.conf-Datei führt zu Mount-Fehlern. Astra Trident hat die Verwendung von empfohlen `find_multipaths: no` Seit der Version 21.07.

Unterstützte Frontends (Orchestrators)

Astra Trident unterstützt mehrere Container-Engines und Orchestrierungslösungen. Dazu gehören:

- Anthos On-Premises (VMware) und Anthos auf Bare Metal 1.12
- Kubernetes 1.23–1.28
- Mirantis Kubernetes Engine 3.5
- OpenShift 4.10 - 4.13

Der Trident-Operator wird durch folgende Versionen unterstützt:

- Anthos On-Premises (VMware) und Anthos auf Bare Metal 1.12
- Kubernetes 1.23–1.28
- OpenShift 4.10 - 4.13

Astra Trident ist auch mit einer Vielzahl weiterer vollständig gemanagter und selbst verwalteter Kubernetes-Angebote kompatibel, darunter Google Kubernetes Engine (GKE), Amazon Elastic Kubernetes Services (EKS), Azure Kubernetes Service (AKS), Rancher und VMware Tanzu Portfolio.



Bevor Sie ein Kubernetes Cluster von 1.24 auf 1.25 oder höher aktualisieren, auf die Astra Trident installiert ist, finden Sie unter "[Aktualisieren einer Helm-Installation](#)".

Unterstützte Back-Ends (Storage)

Zur Verwendung von Astra Trident benötigen Sie ein oder mehrere der folgenden unterstützten Back-Ends:

- Amazon FSX für NetApp ONTAP

- Azure NetApp Dateien
- Cloud Volumes ONTAP
- Cloud Volumes Service für GCP
- FAS/All Flash FAS/Select 9.5 oder höher
- NetApp All-SAN-Array (ASA)
- NetApp HCI/Element Software 11 oder höher

Anforderungen an die Funktionen

Die nachfolgende Tabelle enthält einen Überblick über die Funktionen dieser Version von Astra Trident und die von ihm unterstützten Versionen von Kubernetes.

Merkmal	Kubernetes-Version	Funktionstore erforderlich?
Astra Trident	1.23 - 1.28	Nein
Volume Snapshots	1.23 - 1.28	Nein
PVC aus Volume Snapshots	1.23 - 1.28	Nein
ISCSI PV-Größe	1.23 - 1.28	Nein
Bidirektionales ONTAP-CHAP	1.23 - 1.28	Nein
Dynamische Exportrichtlinien	1.23 - 1.28	Nein
Trident Operator	1.23 - 1.28	Nein
CSI-Topologie	1.23 - 1.28	Nein

Getestete Host-Betriebssysteme

Der Astra Trident unterstützt zwar bestimmte Betriebssysteme offiziell nicht, doch ist es bekannt, dass folgende Betriebssysteme funktionieren:

- Redhat CoreOS (RHCOS) Versionen, die von OpenShift Container Platform (AMD64 und ARM64) unterstützt werden
- RHEL 8 ODER HÖHER (AMD64 UND ARM64)



Für NVMe/TCP ist RHEL 9 oder höher erforderlich.

- Ubuntu 22.04 oder höher (AMD64 und ARM64)
- Windows Server 2019 (AMD64)

Standardmäßig wird Astra Trident in einem Container ausgeführt und läuft daher auf jedem Linux-Mitarbeiter. Diese Mitarbeiter müssen jedoch in der Lage sein, die Volumes, die Astra Trident bietet, je nach den von Ihnen verwendeten Back-Ends mit dem standardmäßigen NFS-Client oder iSCSI-Initiator zu mounten.

Der `tridentctl` Utility läuft auch auf jeder dieser Linux-Distributionen.

Host-Konfiguration

Alle Worker-Nodes im Kubernetes-Cluster müssen in der Lage sein, die Volumes, die Sie für Ihre Pods bereitgestellt haben, zu mounten. Um die Worker-Nodes vorzubereiten, müssen Sie auf der Grundlage Ihrer Treiberauswahl NFS-, iSCSI- oder NVMe-Tools installieren.

["Bereiten Sie den Knoten „Worker“ vor"](#)

Konfiguration des Storage-Systems

Astra Trident erfordert möglicherweise Änderungen an einem Storage-System, bevor es mit einer Backend-Konfiguration verwendet werden kann.

["Back-Ends konfigurieren"](#)

Astra Trident-Ports

Astra Trident erfordert Zugriff auf spezifische Ports für die Kommunikation.

["Astra Trident-Ports"](#)

Container-Images und entsprechende Kubernetes-Versionen

Bei luftvergaschten Installationen ist die folgende Liste eine Referenz für Container-Images, die für die Installation von Astra Trident erforderlich sind. Verwenden Sie die `tridentctl images` Befehl zum Überprüfen der Liste der erforderlichen Container-Images.

Kubernetes-Version	Container-Image
V1.23.0	<ul style="list-style-type: none">• <code>docker.io/netapp/Trident:23.10.0</code>• <code>docker.io/netapp/Trident-AutoSupport:23.10</code>• <code>Registry.k8s.io/SIG-Storage/csi-provisioner:v3.6.0</code>• <code>Registry.k8s.io/SIG-Storage/csi-Attacher:v4.4.0</code>• <code>Registry.k8s.io/SIG-Storage/csi-resizer:v1.9.0</code>• <code>Registry.k8s.io/SIG-Storage/csi-snapshotter:v6.3.0</code>• <code>Registry.k8s.io/SIG-Storage/csi-Node-driver-Registrar:v2.9.0</code>• <code>docker.io/netapp/Trident-Operator:23.10.0</code> (optional)

Kubernetes-Version	Container-Image
V1.24.0	<ul style="list-style-type: none"> • docker.io/netapp/Trident:23.10.0 • docker.io/netapp/Trident-AutoSupport:23.10 • Registry.k8s.io/SIG-Storage/csi-provisioner:v3.6.0 • Registry.k8s.io/SIG-Storage/csi-Attacher:v4.4.0 • Registry.k8s.io/SIG-Storage/csi-resizer:v1.9.0 • Registry.k8s.io/SIG-Storage/csi-snapshotter:v6.3.0 • Registry.k8s.io/SIG-Storage/csi-node-driver-registrar:v2.9.0 • docker.io/netapp/Trident-Operator:23.10.0 (optional)
V1.25.0	<ul style="list-style-type: none"> • docker.io/netapp/Trident:23.10.0 • docker.io/netapp/Trident-AutoSupport:23.10 • Registry.k8s.io/SIG-Storage/csi-provisioner:v3.6.0 • Registry.k8s.io/SIG-Storage/csi-Attacher:v4.4.0 • Registry.k8s.io/SIG-Storage/csi-resizer:v1.9.0 • Registry.k8s.io/SIG-Storage/csi-snapshotter:v6.3.0 • Registry.k8s.io/SIG-Storage/csi-node-driver-registrar:v2.9.0 • docker.io/netapp/Trident-Operator:23.10.0 (optional)
V1.26.0	<ul style="list-style-type: none"> • docker.io/netapp/Trident:23.10.0 • docker.io/netapp/Trident-AutoSupport:23.10 • Registry.k8s.io/SIG-Storage/csi-provisioner:v3.6.0 • Registry.k8s.io/SIG-Storage/csi-Attacher:v4.4.0 • Registry.k8s.io/SIG-Storage/csi-resizer:v1.9.0 • Registry.k8s.io/SIG-Storage/csi-snapshotter:v6.3.0 • Registry.k8s.io/SIG-Storage/csi-node-driver-registrar:v2.9.0 • docker.io/netapp/Trident-Operator:23.10.0 (optional)

Kubernetes-Version	Container-Image
V1.27.0	<ul style="list-style-type: none"> • docker.io/netapp/Trident:23.10.0 • docker.io/netapp/Trident-AutoSupport:23.10 • Registry.k8s.io/SIG-Storage/csi-provisioner:v3.6.0 • Registry.k8s.io/SIG-Storage/csi-Attacher:v4.4.0 • Registry.k8s.io/SIG-Storage/csi-resizer:v1.9.0 • Registry.k8s.io/SIG-Storage/csi-snapshotter:v6.3.0 • Registry.k8s.io/SIG-Storage/csi-node-driver-registrar:v2.9.0 • docker.io/netapp/Trident-Operator:23.10.0 (optional)
V1.28.0	<ul style="list-style-type: none"> • docker.io/netapp/Trident:23.10.0 • docker.io/netapp/Trident-AutoSupport:23.10 • Registry.k8s.io/SIG-Storage/csi-provisioner:v3.6.0 • Registry.k8s.io/SIG-Storage/csi-Attacher:v4.4.0 • Registry.k8s.io/SIG-Storage/csi-resizer:v1.9.0 • Registry.k8s.io/SIG-Storage/csi-snapshotter:v6.3.0 • Registry.k8s.io/SIG-Storage/csi-node-driver-registrar:v2.9.0 • docker.io/netapp/Trident-Operator:23.10.0 (optional)

Copyright-Informationen

Copyright © 2026 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFTE SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGEND EINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.