



# はじめに Astra Trident

NetApp  
June 28, 2024

# 目次

はじめに .....	1
Astra Trident の詳細をご確認ください .....	1
Astra Tridentのクイックスタート .....	10
要件 .....	11

# はじめに

## Astra Trident の詳細をご確認ください

### Astra Trident の詳細をご確認ください

Astra Tridentは、NetAppが ["Astra 製品ファミリー"](#)。Container Storage Interface (CSI) などの業界標準のインターフェイスを使用して、コンテナ化されたアプリケーションの永続性要求を満たすように設計されています。

#### アストラとは

Astra を使用すると、Kubernetes で実行されている大量のデータコンテナ化ワークロードを、パブリッククラウドとオンプレミス間で簡単に管理、保護、移動できます。

Astraは、Astra Tridentを基盤に構築された永続的コンテナストレージをプロビジョニング、提供します。また、Snapshot、バックアップとリストア、アクティビティログ、アクティブクローニングなどの高度なアプリケーション対応データ管理機能も提供し、データ保護、ディザスタ/データリカバリ、データ監査、Kubernetesワークロードの移行のユースケースに対応します。

の詳細を確認してください ["Astraをご利用いただくか、無償トライアルにご登録ください"](#)。

#### Astra Tridentとは

Astra Tridentでは、パブリッククラウドやオンプレミスにあるONTAP (AFF、NetApp FAS、Select、Cloud、Amazon FSx for NetApp ONTAP)、Elementソフトウェア (NetApp HCI、SolidFire)、Azure NetApp Filesサービス、Cloud Volumes Service on Google Cloud

Astra Tridentは、コンテナストレージインターフェイス (CSI) に準拠した動的ストレージオーケストレーションツールで、["Kubernetes"](#)。Astra Tridentは、単一のコントローラポッドと、クラスタ内の各ワーカーノード上のノードポッドとして実行されます。を参照してください ["Astra Tridentのアーキテクチャ"](#) を参照してください。

Astra Tridentは、NetAppストレージプラットフォーム向けのDockerエコシステムと直接統合することもできます。NetApp Docker Volume Plugin (nDVP) は、ストレージプラットフォームからDockerホストへのストレージリソースのプロビジョニングと管理をサポートします。を参照してください ["Astra Trident for Dockerを導入"](#) を参照してください。



Kubernetesを初めて使用する場合は、["Kubernetesの概念とツール"](#)。

#### Astra TridentのTest Driveプログラム

Test Driveプログラムを利用するには、「コンテナ化されたワークロード向けの永続的ストレージの簡単な導入とクローニング」へのアクセスをリクエスト ["ネットアップのテスト用ドライブ"](#) すぐに使用できるラボイメージを使用する。このテストドライブは、3ノードのKubernetesクラスタとAstra Tridentがインストールおよび設定されたサンドボックス環境を提供します。これは、Astra Tridentについて理解を深め、その機能を確認するための優れた方法です。

もう1つのオプションは、["kubeadm インストールガイド"](#) Kubernetes が提供します。



これらの手順を使用して構築したKubernetesクラスタは、本番環境では使用しないでください。本番環境向けクラスタ向けに、ディストリビューションから提供されている本番環境導入ガイドを使用します。

## KubernetesとNetApp製品の統合

NetAppのストレージ製品ポートフォリオは、Kubernetesクラスタのさまざまな要素と統合されているため、高度なデータ管理機能が提供され、Kubernetes環境の機能、機能、パフォーマンス、可用性が強化されます。

### NetApp ONTAP 対応の Amazon FSX

"NetApp ONTAP 対応の Amazon FSX" は、NetApp ONTAPストレージオペレーティングシステムを基盤とするファイルシステムを起動して実行できる、フルマネージドのAWSサービスです。

### Azure NetApp Files の特長

"Azure NetApp Files の特長" は、ネットアップが提供するエンタープライズクラスの Azure ファイル共有サービスです。要件がきわめて厳しいファイルベースのワークロードも、ネットアップが提供するパフォーマンスと充実のデータ管理機能を使用して、Azure でネイティブに実行できます。

### Cloud Volumes ONTAP

"Cloud Volumes ONTAP" は、クラウドで ONTAP データ管理ソフトウェアを実行するソフトウェア型ストレージアプライアンスです。

### Cloud Volumes Service for Google Cloud

"NetApp Cloud Volumes Service for Google Cloud" は、NFS や SMB 経由で NAS ボリュームにオールフラッシュのパフォーマンスを提供する、クラウドネイティブのファイルサービスです。

### Element ソフトウェア

"要素 (Element)" ストレージ管理者は、パフォーマンスを保証し、ストレージの設置面積を合理化することで、ワークロードを統合できます。

### NetApp HCI

"NetApp HCI" 日常業務を自動化し、インフラ管理者がより重要な業務に集中できるようにすることで、データセンターの管理と拡張を簡易化します。

Astra Tridentでは、コンテナ化されたアプリケーション用のストレージデバイスを、基盤となるNetApp HCIストレージプラットフォームに直接プロビジョニングして管理できます。

## NetApp ONTAP

"NetApp ONTAP" は、NetAppのマルチプロトコルユニファイドストレージオペレーティングシステムで、あらゆるアプリケーションに高度なデータ管理機能を提供します。

ONTAP システムには、オールフラッシュ、ハイブリッド、オール HDD のいずれかの構成が採用されており、自社開発のハードウェア（FAS と AFF）、ノーブランド製品（ONTAP Select）、クラウドのみ（Cloud Volumes ONTAP）など、さまざまな導入モデルが用意されています。Astra Tridentは、これらのONTAP導入モデルをサポートしています。

を参照してください。

- ["ネットアップアストラ製品ファミリー"](#)
- ["Astra Control Service のマニュアル"](#)
- ["Astra Control Center のドキュメント"](#)
- ["Astra API ドキュメント"](#)

## Astra Tridentのアーキテクチャ

Astra Tridentは、単一のコントローラポッドと、クラスタ内の各ワーカーノード上のノードポッドとして実行されます。Astra Tridentボリュームをマウントするすべてのホストでノードポッドが実行されている必要があります。

### コントローラポッドとノードポッドについて

Astra Tridentを単一システムとして導入 [Tridentコントローラポッド](#) および1つ以上 [Tridentノードポッド](#) Kubernetesクラスタ上で、標準のKUBSI\_CSI Sidecar Containers\_を使用してCSIプラグインの導入を簡素化します。 ["Kubernetes CSIサイドカーコンテナ"](#) Kubernetes Storageコミュニティが管理しています。

Kubernetes ["ノードセレクタ"](#) および ["寛容さと汚れ"](#) は、特定のノードまたは優先ノードで実行されるようにポッドを制限するために使用されます。コントローラポッドとノードポッドのノードセレクタと許容範囲は、Astra Tridentのインストール時に設定できます。

- コントローラプラグインは、Snapshotやサイズ変更などのボリュームのプロビジョニングと管理を処理します。
- ノードプラグインによって、ノードへのストレージの接続が処理されます。

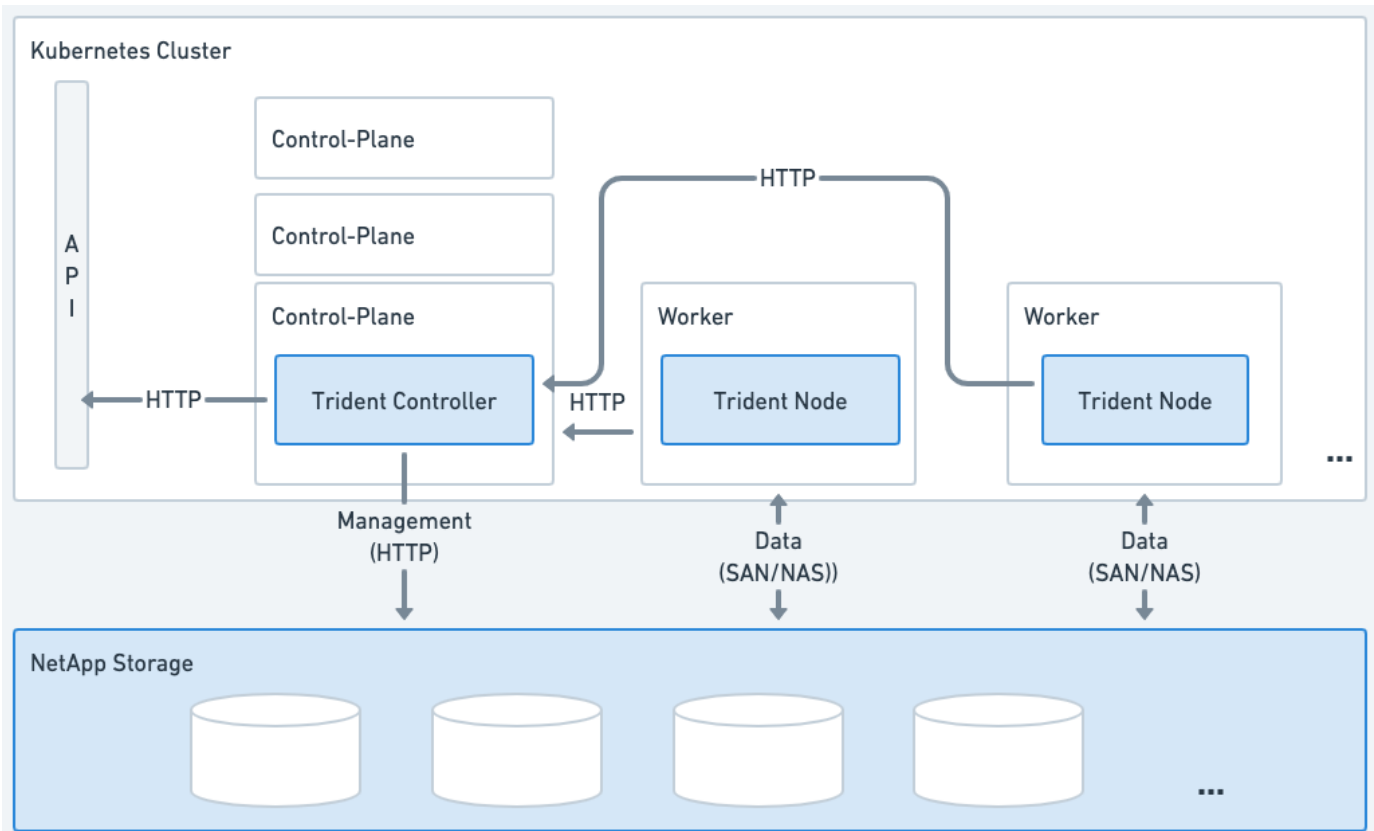


図 1. Kubernetes クラスタに導入される Astra Trident

#### Trident コントローラポッド

Trident コントローラポッドは、CSI コントローラプラグインを実行する単一のポッドです。

- NetApp ストレージ内のボリュームのプロビジョニングと管理を担当
- Kubernetes 環境で管理
- インストールパラメータに応じて、コントロールプレーンノードまたはワーカーノードで実行できます。

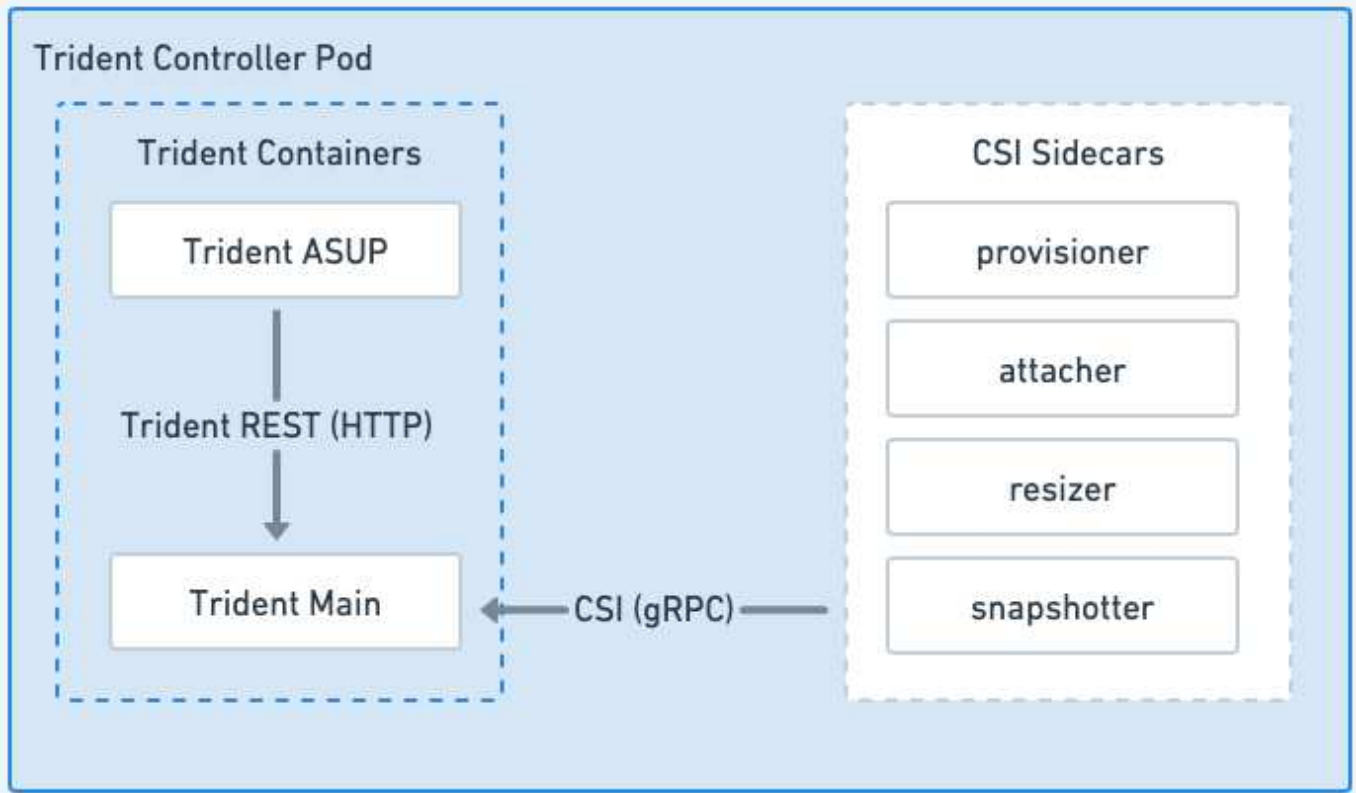


図 2. Tridentコントローラポッドの図

#### Tridentノードポッド

Tridentノードポッドは、CSIノードプラグインを実行する特権ポッドです。

- ホストで実行されているPodのストレージのマウントとアンマウントを担当します。
- Kubernetesデーモンセットで管理
- NetAppストレージをマウントするすべてのノードで実行する必要がある

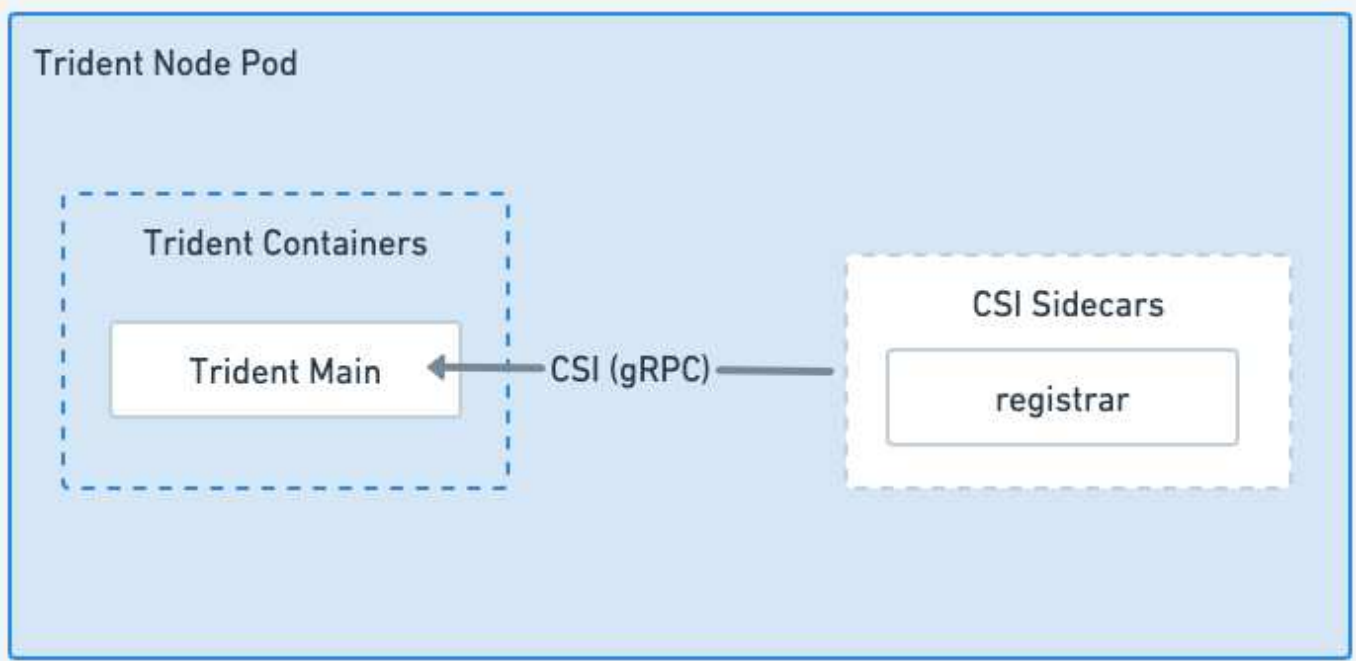


図 3. Tridentノードのポッド図

サポートされる **Kubernetes** クラスタアーキテクチャ

Astra Trident は、次の Kubernetes アーキテクチャでサポートされています。

Kubernetes クラスタアーキテクチャ	サポートされま す	デフォルトのインストールです
単一マスター、コンピューティング	はい。	はい。
複数のマスター、コンピューティング	はい。	はい。
マスター、`etcd`コンピューティング	はい。	はい。
マスター、インフラ、コンピューティング	はい。	はい。

## 概念

### プロビジョニング

Trident の Astra プロビジョニングの主なフェーズは 2 つあります。最初のフェーズでは、ストレージクラスを適切なバックエンドストレージプールのセットに関連付け、プロビジョニング前の必要な準備として実行します。2番目のフェーズではボリュームの作成自体が行われ、保留状態のボリュームのストレージクラスに関連付けられたストレージプールの中からストレージプールを選択する必要があります。



## ストレージクラスの関連付け

バックエンドストレージプールをストレージクラスに関連付けるには、ストレージクラスの要求された属性とその両方が必要です `storagePools`、`additionalStoragePools` および `excludeStoragePools` リスト。ストレージクラスを作成すると、Trident はバックエンドごとに提供される属性とプールを、ストレージクラスから要求された属性とプールと比較します。要求された属性とプール名がストレージプールの属性と名前ですべて一致した場合、Astra Trident がそのストレージプールを、そのストレージクラスに適した一連のストレージプールに追加します。さらに、Trident の Astra では、にリストされているすべてのストレージプールが追加されます `additionalStoragePools` 属性がストレージクラスの要求した属性の一部または全部を満たしていない場合も、そのセットにリストされます。を使用する必要があります `excludeStoragePools` ストレージクラスに対して使用するストレージプールを上書きおよび削除するリスト。Astra Trident では、新しいバックエンドを追加するたびに同様のプロセスが実行され、ストレージプールが既存のストレージクラスのストレージクラスを満たしているかどうかを確認され、除外済みとマークされているストレージが削除されます。

## ボリュームの作成

Trident がさらに、ストレージクラスとストレージプールの間の関連付けを使用して、ボリュームのプロビジョニング先を決定します。ボリュームを作成すると、最初にそのボリュームのストレージクラス用の一連のストレージプールが Trident から取得されます。また、ボリュームにプロトコルを指定した場合、Astra Trident は要求されたプロトコルを提供できないストレージプールを削除します（たとえば、NetApp HCI / SolidFire バックエンドはファイルベースのボリュームを提供できませんが、ONTAP NAS バックエンドはブロックベースのボリュームを提供できません）。Trident がこのセットの順序をランダム化し、ボリュームを均等に分散してから、各ストレージプールでボリュームを順番にプロビジョニングしようとします。成功した場合は正常に返され、プロセスで発生したエラーが記録されます。Astra Trident は、要求されたストレージクラスとプロトコルで使用可能なすべてのストレージプールで \* プロビジョニングに失敗した場合にのみ、障害 \* を返します。

## ボリューム Snapshot

Trident がドライバ用のボリュームスナップショットの作成をどのように処理するかについては、こちらをご覧ください。

ボリューム Snapshot の作成方法について説明します

- をクリックします `ontap-nas`、`ontap-san`、`gcp-cvs` および `azure-netapp-files` ドライバ、各永続ボリューム (PV) は FlexVol にマッピングされます。その結果、ボリューム Snapshot は ネットアップ Snapshot として作成されます。NetApp のスナップショット・テクノロジーは競合するスナップショット・テクノロジーよりも安定性'拡張性'リカバリ性'パフォーマンスを提供します Snapshot コピーは、作成時とストレージスペースの両方で非常に効率的です。
- をクリックします `ontap-nas-flexgroup` ドライバ、各永続ボリューム (PV) は FlexGroup にマッピングされます。その結果、ボリューム Snapshot は NetApp FlexGroup Snapshot として作成されます。NetApp のスナップショット・テクノロジーは競合するスナップショット・テクノロジーよりも安定性'拡張性'リカバリ性'パフォーマンスを提供します Snapshot コピーは、作成時とストレージスペースの両方で非常に効率的です。
- をクリックします `ontap-san-economy` ドライバと PVS は、共有 FlexVol 上に作成された LUN にマッピングされます。PVS のボリューム Snapshot は、関連付けられた LUN の FlexClone を実行することで実現されます。ONTAP FlexClone テクノロジーを使用すると、大規模なデータセットのコピーをほぼ瞬時に作成できます。コピーと親でデータブロックが共有されるため、メタデータに必要な分しかストレージは消費されません。
- をクリックします `solidfire-san` ドライバ。各 PV は、NetApp Element ソフトウェア / NetApp HCI クラスタ上に作成された LUN にマッピングされます。ボリューム Snapshot は、基盤となる LUN の Element

Snapshot で表されます。これらの Snapshot はポイントインタイムコピーであり、消費するシステムリソースとスペースはごくわずかです。

- を使用して作業している場合 `ontap-nas` および `ontap-san` ドライバ、ONTAP スナップショットは、FlexVol のポイントインタイムコピーであり、FlexVol 自体のスペースを消費します。その結果、ボリューム内の書き込み可能なスペースが、Snapshot の作成やスケジュール設定にかかる時間を短縮できます。この問題に対処する簡単な方法の 1 つは、Kubernetes を使用してサイズを変更することでボリュームを拡張することです。もう 1 つの方法は、不要になった Snapshot を削除することです。Kubernetes で作成されたボリューム Snapshot を削除すると、関連付けられている ONTAP Snapshot が Astra Trident から削除されます。Kubernetes で作成されていない ONTAP スナップショットも削除できます。

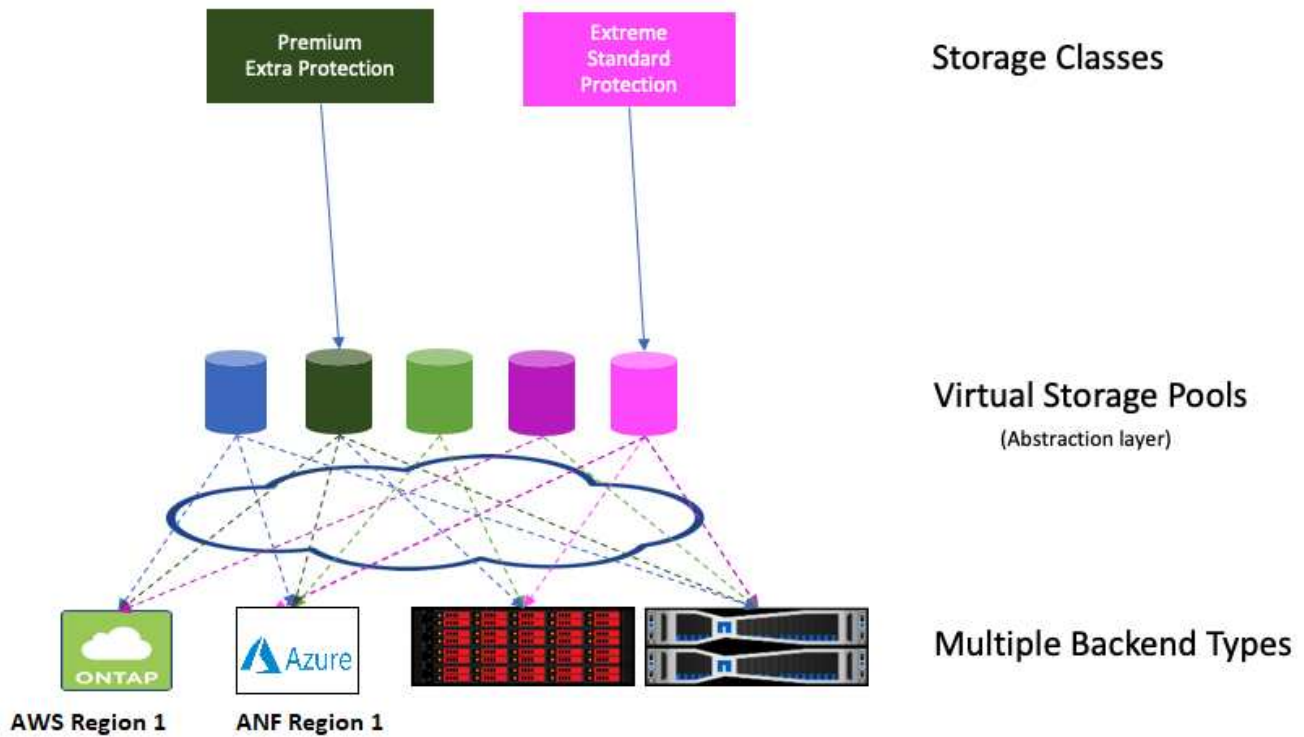
ネットアップの Trident では、ボリューム Snapshot を使用して PVS を新規作成できます。これらの Snapshot から PVS を作成するには、サポート対象の ONTAP および CVS バックエンドに対して FlexClone テクノロジーを使用します。Snapshot から PV を作成する場合、元のボリュームは Snapshot の親ボリュームの FlexClone1 になります。。 `solidfire-san` ドライバは、Element ソフトウェアのボリュームクローンを使用して Snapshot から PVS を作成します。ここで、Element Snapshot からクローンを作成します。

## 仮想プール

仮想プールは、Astra Trident ストレージバックエンドと Kubernetes の間に抽象化レイヤを提供します `StorageClasses`。管理者は、を作成することなく、バックエンドに依存しない共通の方法で、各バックエンドの場所、パフォーマンス、保護などの側面を定義できます `StorageClass` 目的の条件を満たすために使用する物理バックエンド、バックエンドプール、またはバックエンドタイプを指定します。

仮想プールについて説明します

ストレージ管理者は、任意の Astra Trident バックエンドに JSON または YAML 定義ファイルで仮想プールを定義できます。



仮想プールリストの外部で指定されたすべての要素はバックエンドにグローバルであり、すべての仮想プールに適用されます。一方、各仮想プールは、1つまたは複数の要素を個別に指定できます（バックエンドグローバルな要素を上書きします）。



- 仮想プールを定義する場合は、バックエンド定義内の既存の仮想プールの順序を変更しないでください。
- 既存の仮想プールの属性を変更しないことをお勧めします。変更を行うには、新しい仮想プールを定義する必要があります。

ほとんどの項目はバックエンド固有の用語で指定されます。アスペクト値は、バックエンドのドライバの外部には表示されず、での照合には使用できません StorageClasses。代わりに、管理者が各仮想プールに1つ以上のラベルを定義します。各ラベルはキー：値のペアで、ラベルは一意的バックエンド間で共通です。側面と同様に、ラベルはプールごとに指定することも、バックエンドに対してグローバルに指定することもできます。名前と値があらかじめ定義されている側面とは異なり、管理者は必要に応じてラベルキーと値を定義する完全な裁量を持っています。ストレージ管理者は、仮想プールごとにラベルを定義したり、ボリュームをラベルでグループ化したりできます。

A StorageClass セレクタパラメータ内のラベルを参照して、使用する仮想プールを指定します。仮想プールセレクタでは、次の演算子がサポートされます。

演算子	例	プールのラベル値は次のとおりです。
=	パフォーマンス = プレミアム	一致
!=	パフォーマンス != 非常に優れています	一致しません
in	場所 (東部、西部)	値のセットに含まれています

演算子	例	プールのラベル値は次のとおりです。
notin	パフォーマンス記名（シルバー、ブロンズ）	値のセットに含まれていません
<key>	保護	任意の値で存在します
!<key>	!保護	存在しません

## ボリュームアクセスグループ

Trident がどのように活用されているかをご確認ください "[ボリュームアクセスグループ](#)"。



CHAP を使用する場合は、このセクションを無視してください。CHAP では、管理を簡易化し、以下に説明する拡張の制限を回避することが推奨されます。また、CSI モードで Astra Trident を使用している場合は、このセクションを無視できます。Astra Trident は、強化された CSI プロビジョニングツールとしてインストールされた場合、CHAP を使用します。

ボリュームアクセスグループについて学習する

Astra Trident は、ボリュームアクセスグループを使用して、プロビジョニングするボリュームへのアクセスを制御できる CHAP が無効になっている場合は、というアクセスグループが検索されます trident 構成に1つ以上のアクセスグループIDを指定していない場合。

Astra Trident は、設定されたアクセスグループに新しいボリュームを関連付けますが、アクセスグループ自体の作成や管理は行いません。アクセスグループには、ストレージバックエンドを Astra Trident に追加する前に存在する必要があります。また、そのバックエンドでプロビジョニングされたボリュームをマウントできる可能性がある Kubernetes クラスタ内のすべてのノードの iSCSI IQN が含まれている必要があります。ほとんどのインストール環境では、クラスタ内のすべてのワーカーノードがこれに含まれます。

Kubernetes クラスタに 64 個を超えるノードがある場合は、複数のアクセスグループを使用する必要があります。各アクセスグループには最大 64 個の IQN を含めることができ、各ボリュームは 4 つのアクセスグループに属することができます。最大 4 つのアクセスグループを設定すると、クラスタ内の任意のノードから最大 256 ノードのサイズのすべてのボリュームにアクセスできるようになります。ボリュームアクセスグループの最新の制限については、[を参照してください](#)。

デフォルト設定を使用している設定から変更する場合 trident 他のユーザも使用するアクセスグループには、のIDを追加します trident リスト内のアクセスグループ。

## Astra Trident のクイックスタート

Astra Trident をインストールすると、わずかな手順でストレージリソースの管理を開始できます。作業を開始する前に、"[Astra Trident の要件](#)"。



Docker については、[を参照してください](#)。"[Trident for Docker が必要です](#)"。



### 1 Astra Trident をインストール

Astra Trident には、さまざまな環境や組織に最適化された複数のインストール方法とモードが用意されていま

す。

## "Astra Trident をインストール"

2

ワーカーノードを準備します

Kubernetesクラスタ内のすべてのワーカーノードが、ポッド用にプロビジョニングしたボリュームをマウントできる必要があります。

### "ワーカーノードを準備します"

3

バックエンドを作成します

バックエンドは、Astra Trident とストレージシステムの関係性を定義します。Trident がストレージシステムとの通信方法を Trident から指示し、Astra Trident がボリュームをプロビジョニングする方法も解説します。

### "バックエンドの設定" ストレージシステム

4

Kubernetesストレージクラスの作成

Kubernetes StorageClassオブジェクトでは、Astra Tridentがプロビジョニングツールとして指定され、カスタマイズ可能な属性を使用してボリュームをプロビジョニングするためのストレージクラスを作成できます。Astra Tridentは、Astra Tridentプロビジョニングツールを指定する、Kubernetesオブジェクト用の一致するストレージクラスを作成します。

### "ストレージクラスを作成する。"

5

ボリュームをプロビジョニングする

A\_PersistentVolume\_ (PV) は、Kubernetesクラスタ上のクラスタ管理者がプロビジョニングする物理ストレージリソースです。*PersistentVolumeClaim* (PVC) は、クラスタ上のPersistentVolumeへのアクセス要求です。

設定したKubernetes StorageClassを使用してPVへのアクセスを要求するPersistentVolume (PV) とPersistentVolumeClaim (PVC) を作成します。その後、PVをポッドにマウントできます。

### "ボリュームをプロビジョニングする"

## 次の手順

バックエンドの追加、ストレージクラスの管理、バックエンドの管理、ボリューム処理の実行が可能になりました。

## 要件

Astra Tridentをインストールする前に、次の一般的なシステム要件を確認してください。個々のバックエンドには追加の要件がある場合があります。

## Astra Tridentに関する重要な情報

- Astra Tridentに関する次の重要な情報をお読みください。\*

### **<strong> : Trident </strong>** に関する重要な情報

- TridentでKubernetes 1.29がサポートされるようになりました。Kubernetesをアップグレードする前にAstra Tridentをアップグレードしてください。
- Astra Tridentは、SAN環境でマルチパス構成を厳密に使用し、推奨される値をに設定します  
`find_multipaths: no` multipath.confファイル内。

非マルチパス構成またはを使用 `find_multipaths: yes` または `find_multipaths: smart` multipath.confファイルの値が原因でマウントが失敗します。Astra Tridentでは、`find_multipaths: no` 21.07リリース以降

## サポートされるフロントエンド（オーケストレーションツール）

Trident Astra は、次のような複数のコンテナエンジンとオーケストレーションツールをサポート

- Anthosオンプレミス（VMware）とAnthos（ベアメタル1.16）
- Kubernetes 1.23~1.29
- OpenShift 4.10-4.15

Trident オペレータは、次のリリースでサポートされています。

- Anthosオンプレミス（VMware）とAnthos（ベアメタル1.16）
- Kubernetes 1.23~1.29
- OpenShift 4.10-4.15

Astra Tridentは、Google Kubernetes Engine（GKE）、Amazon Elastic Kubernetes Services（EKS）、Azure Kubernetes Service（AKS）、Mirantis Kubernetes Engine（MKE）、Rancher、VMware Tanzu Portfolioなど、他のフルマネージド/自己管理型Kubernetesソリューションとも連携します。

Astra TridentとONTAPは、["KubeVirt"](#)。



Astra TridentがインストールされているKubernetesクラスタを1.24から1.25以降にアップグレードする前に、[を参照してください。](#) ["Helmインストールのアップグレード"](#)。

## サポートされるバックエンド（ストレージ）

Astra Trident を使用するには、次のバックエンドを 1 つ以上サポートする必要があります。

- NetApp ONTAP 対応の Amazon FSX
- Azure NetApp Files の特長
- Cloud Volumes ONTAP
- Cloud Volumes Service for GCP

- FAS/AFF / Select 9.5以降
- ネットアップオール SAN アレイ（ASA）
- NetApp HCI / Elementソフトウェア11以降

## 機能の要件

次の表は、このリリースの Astra Trident で利用できる機能と、サポートする Kubernetes のバージョンをまとめたものです。

フィーチャー（Feature）	Kubernetes のバージョン	フィーチャーゲートが必要ですか？
Astra Trident	1.23 ~ 1.29	いいえ
ボリューム Snapshot	1.23 ~ 1.29	いいえ
ボリューム Snapshot からの PVC	1.23 ~ 1.29	いいえ
iSCSI PV のサイズ変更	1.23 ~ 1.29	いいえ
ONTAP 双方向 CHAP	1.23 ~ 1.29	いいえ
動的エクスポートポリシー	1.23 ~ 1.29	いいえ
Trident のオペレータ	1.23 ~ 1.29	いいえ
CSI トポロジ	1.23 ~ 1.29	いいえ

## テスト済みのホストオペレーティングシステム

Astra Tridentは特定のオペレーティングシステムを正式にサポートしていませんが、動作確認済みのものは次のとおりです。

- OpenShift Container Platform（AMD64およびARM64）でサポートされているRed Hat CoreOS（RHCOS）のバージョン
- RHEL 8+（AMD64およびARM64）



NVMe/TCPにはRHEL 9以降が必要です。

- Ubuntu 22.04以降（AMD64およびARM64）
- Windows Server 2019（AMD64）

デフォルトでは、Astra Trident はコンテナで実行されるため、任意の Linux ワーカーで実行されます。ただし、その場合、使用するバックエンドに応じて、標準の NFS クライアントまたは iSCSI イニシエータを使用して Astra Trident が提供するボリュームをマウントする必要があります。

。 tridentctl ユーティリティは、これらのLinuxディストリビューションでも動作します。

## ホストの設定

Kubernetesクラスタ内のすべてのワーカーノードが、ポッド用にプロビジョニングしたボリュームをマウントできる必要があります。ワーカーノードを準備するには、ドライバの選択に基づいてNFS、iSCSI、またはNVMeのツールをインストールする必要があります。

["ワーカーノードを準備します"](#)

## ストレージシステムの構成：

Astra Tridentでは、バックエンド構成でストレージシステムを使用する前に、変更が必要になる場合があります。

["バックエンドを設定"](#)

## Astra Trident ポート

Astra Tridentが通信するには、特定のポートへのアクセスが必要です。

["Astra Trident ポート"](#)

## コンテナイメージと対応する Kubernetes バージョン

エアギャップのある環境では、Astra Trident のインストールに必要なコンテナイメージを次の表に示します。を使用します tridentctl images 必要なコンテナイメージのリストを確認するコマンド。

Kubernetes のバージョン	コンテナイメージ
v1.3.0	<ul style="list-style-type: none"><li>• Docker .io / NetApp / Trident : 24.02.0</li><li>• docker.io / netapp/trident-autosupport : 24.02</li><li>• registry.k8s.io/sig-storage/csi-provisioner : v4.0.0</li><li>• registry.k8s.io/sig-storage/csi-attacher : v4.5.0</li><li>• registry.k8s.io/sig-storage/csi-resizer : v1.9.3</li><li>• registry.k8s.io/sig-storage/csi-snapshotter : v6.3.3</li><li>• registry.k8s.io/sig-storage/csi-node-driver-registrar : v2.10.0</li><li>• docker.io/netapp/trident-operator : 24.02.0 (オプション)</li></ul>



Kubernetes のバージョン	コンテナイメージ
v1.24.0	<ul style="list-style-type: none"> <li>• Docker .io / NetApp / Trident : 24.02.0</li> <li>• docker.io / netapp/trident-autosupport : 24.02</li> <li>• registry.k8s.io/sig-storage/csi-provisioner : v4.0.0</li> <li>• registry.k8s.io/sig-storage/csi-attacher : v4.5.0</li> <li>• registry.k8s.io/sig-storage/csi-resizer : v1.9.3</li> <li>• registry.k8s.io/sig-storage/csi-snapshotter : v6.3.3</li> <li>• registry.k8s.io/sig-storage/csi-node-driver-registrar : v2.10.0</li> <li>• docker.io/netapp/trident-operator : 24.02.0 (オプション)</li> </ul>
v1.25.0	<ul style="list-style-type: none"> <li>• Docker .io / NetApp / Trident : 24.02.0</li> <li>• docker.io / netapp/trident-autosupport : 24.02</li> <li>• registry.k8s.io/sig-storage/csi-provisioner : v4.0.0</li> <li>• registry.k8s.io/sig-storage/csi-attacher : v4.5.0</li> <li>• registry.k8s.io/sig-storage/csi-resizer : v1.9.3</li> <li>• registry.k8s.io/sig-storage/csi-snapshotter : v6.3.3</li> <li>• registry.k8s.io/sig-storage/csi-node-driver-registrar : v2.10.0</li> <li>• docker.io/netapp/trident-operator : 24.02.0 (オプション)</li> </ul>
v1.26.0	<ul style="list-style-type: none"> <li>• Docker .io / NetApp / Trident : 24.02.0</li> <li>• docker.io / netapp/trident-autosupport : 24.02</li> <li>• registry.k8s.io/sig-storage/csi-provisioner : v4.0.0</li> <li>• registry.k8s.io/sig-storage/csi-attacher : v4.5.0</li> <li>• registry.k8s.io/sig-storage/csi-resizer : v1.9.3</li> <li>• registry.k8s.io/sig-storage/csi-snapshotter : v6.3.3</li> <li>• registry.k8s.io/sig-storage/csi-node-driver-registrar : v2.10.0</li> <li>• docker.io/netapp/trident-operator : 24.02.0 (オプション)</li> </ul>

Kubernetes のバージョン	コンテナイメージ
v1.27.0	<ul style="list-style-type: none"> <li>• Docker .io / NetApp / Trident : 24.02.0</li> <li>• docker.io / netapp/trident-autosupport : 24.02</li> <li>• registry.k8s.io/sig-storage/csi-provisioner : v4.0.0</li> <li>• registry.k8s.io/sig-storage/csi-attacher : v4.5.0</li> <li>• registry.k8s.io/sig-storage/csi-resizer : v1.9.3</li> <li>• registry.k8s.io/sig-storage/csi-snapshotter : v6.3.3</li> <li>• registry.k8s.io/sig-storage/csi-node-driver-registrar : v2.10.0</li> <li>• docker.io/netapp/trident-operator : 24.02.0 (オプション)</li> </ul>
v1.28.0	<ul style="list-style-type: none"> <li>• Docker .io / NetApp / Trident : 24.02.0</li> <li>• docker.io / netapp/trident-autosupport : 24.02</li> <li>• registry.k8s.io/sig-storage/csi-provisioner : v4.0.0</li> <li>• registry.k8s.io/sig-storage/csi-attacher : v4.5.0</li> <li>• registry.k8s.io/sig-storage/csi-resizer : v1.9.3</li> <li>• registry.k8s.io/sig-storage/csi-snapshotter : v6.3.3</li> <li>• registry.k8s.io/sig-storage/csi-node-driver-registrar : v2.10.0</li> <li>• docker.io/netapp/trident-operator : 24.02.0 (オプション)</li> </ul>
v1.29.0	<ul style="list-style-type: none"> <li>• Docker .io / NetApp / Trident : 24.02.0</li> <li>• docker.io / netapp/trident-autosupport : 24.02</li> <li>• registry.k8s.io/sig-storage/csi-provisioner : v4.0.0</li> <li>• registry.k8s.io/sig-storage/csi-attacher : v4.5.0</li> <li>• registry.k8s.io/sig-storage/csi-resizer : v1.9.3</li> <li>• registry.k8s.io/sig-storage/csi-snapshotter : v6.3.3</li> <li>• registry.k8s.io/sig-storage/csi-node-driver-registrar : v2.10.0</li> <li>• docker.io/netapp/trident-operator : 24.02.0 (オプション)</li> </ul>

## 著作権に関する情報

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および/または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。