



# NetApp Astra Trident の概要

## NetApp Solutions

NetApp  
April 10, 2024

This PDF was generated from [https://docs.netapp.com/ja-jp/netapp-solutions/containers/vtwn\\_trident\\_ontap\\_nfs.html](https://docs.netapp.com/ja-jp/netapp-solutions/containers/vtwn_trident_ontap_nfs.html) on April 10, 2024. Always check docs.netapp.com for the latest.

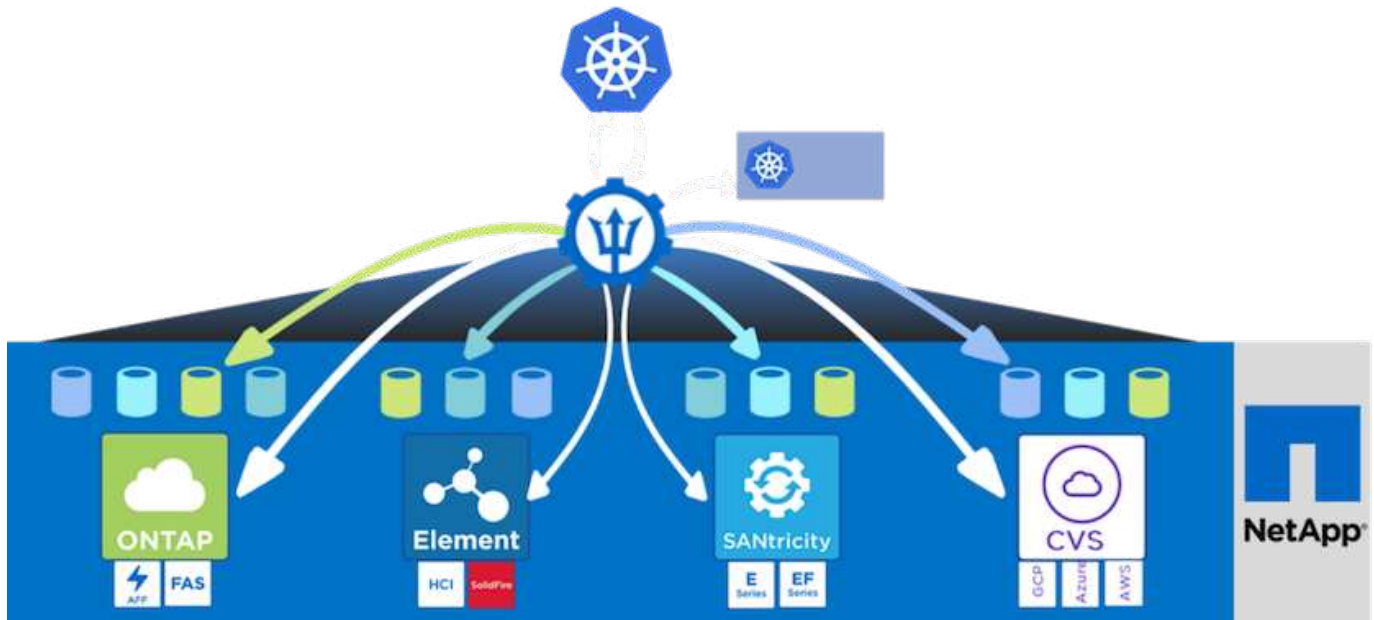
# 目次

Astra Tridentの概要 .....	1
Helmを使用してTridentオペレータを導入 .....	1
ストレージシステムバックエンドを作成 .....	3
NetApp ONTAP の NFS 構成 .....	3
NetApp ONTAP iSCSI 構成 .....	9

# Astra Tridentの概要

Astra Tridentは、コンテナやKubernetesディストリビューション向けの、完全にサポートされているオープンソースのストレージオーケストレーションツールです。{k8s\_distribution\_name} などが挙げられます。Trident は、NetApp ONTAP や Element ストレージシステムを含むネットアップストレージポートフォリオ全体と連携し、NFS 接続と iSCSI 接続もサポートします。Trident を使用すると、ストレージ管理者の手を煩わせることなく、エンドユーザがネットアップストレージシステムからストレージをプロビジョニングして管理できるため、DevOps ワークフローが高速化されます。

管理者は、プロジェクトのニーズやストレージシステムモデルに基づいて複数のストレージバックエンドを構成し、圧縮、特定のディスクタイプ、QoS レベルなどの高度なストレージ機能を有効にして一定のレベルのパフォーマンスを保証できます。定義されたバックエンドは、プロジェクトの開発者が永続的ボリューム要求（PVC）を作成し、永続的ストレージをオンデマンドでコンテナに接続するために使用できます。



Astra Tridentは、迅速な開発サイクルを実現し、Kubernetesと同様、年間4回リリースされます。

Tridentの最新バージョンは2022年4月に22.04にリリースされました。Trident のどのバージョンがサポートされているかを確認できます [Kubernetes ディストリビューションのテストに使用 "こちらをご覧ください"](#)。

20.04 リリース以降、Trident のセットアップはTrident オペレータによって実行されます。オペレータが大規模な導入を容易にし、Tridentのインストールの一部として導入されるポッドの自己修復などの追加サポートを提供します。

21.01 リリースでは、Trident Operator のインストールを容易にするために Helm チャートを使用できるようになりました。

## Helmを使用してTridentオペレータを導入

1. Trident にはこのファイルを渡すオプションがないため、まず、ユーザクラスタの「kubeconfig」ファイルの場所を環境変数として設定します。

```
<<<<<<< HEAD
[netapp-user@rhel7]$ export KUBECONFIG=~/.tanzu-install/auth/kubeconfig
=====
[netapp-user@rhel7]$ export KUBECONFIG=~/.Tanzu-install/auth/kubeconfig
>>>>>>> eba1007b77b1ef6011dadd158f1df991acc5299f
```

## 2. NetApp Astra Trident Helmリポジトリを追加

```
[netapp-user@rhel7]$ helm repo add netapp-trident
https://netapp.github.io/trident-helm-chart
"netapp-trident" has been added to your repositories
```

## 3. Helmリポジトリを更新します。

```
[netapp-user@rhel7]$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "netapp-trident" chart repository
...Successfully got an update from the "bitnami" chart repository
Update Complete. ☐Happy Helming!☐
```

## 4. Tridentをインストールするための新しいネームスペースを作成します。

```
[netapp-user@rhel7]$ kubectl create ns trident
```

## 5. DockerHubのクレデンシャルを使用してシークレットを作成し、Astra Tridentイメージをダウンロードします。

```
[netapp-user@rhel7]$ kubectl create secret docker-registry docker-registry-cred --docker-server=docker.io --docker-username=netapp-solutions-tme --docker-password=xxxxxxx -n trident
```

## 6. TKGS (vSphere with Tanzu) またはTKG (管理クラスタを含む) で管理されるユーザまたはワークロードクラスタの場合、次の手順を実行してAstra Tridentをインストールします。

- ログインしているユーザに、tridentネームスペースにサービスアカウントを作成する権限があり、tridentネームスペースのサービスアカウントにポッドを作成する権限があることを確認します。
- 以下のHelmコマンドを実行し、作成したネームスペースにTridentオペレータをインストールします。

```
[netapp-user@rhel7]$ helm install trident netapp-trident/trident-operator -n trident --set imagePullSecrets[0]=docker-registry-cred
```

7. TKGI導入によって管理されるユーザまたはワークロードクラスタの場合は、次のHelmコマンドを実行して、作成されたネームスペースにTridentオペレータをインストールします。

```
[netapp-user@rhel7]$ helm install trident netapp-trident/trident-operator -n trident --set imagePullSecrets[0]=docker-registry-cred,kubeletDir="/var/vcap/data/kubelet"
```

8. Tridentポッドが稼働中であることを確認します。

NAME	READY	STATUS	RESTARTS
AGE			
trident-csi-6vv62	2/2	Running	0
14m			
trident-csi-cfd844bcc-sqhcq	6/6	Running	0
12m			
trident-csi-dfcmz	2/2	Running	0
14m			
trident-csi-pb2n7	2/2	Running	0
14m			
trident-csi-qsw6z	2/2	Running	0
14m			
trident-operator-67c94c4768-xw978	1/1	Running	0
14m			

```
[netapp-user@rhel7]$ ./tridentctl -n trident version
+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+
| 22.04.0        | 22.04.0        |
+-----+
```

## ストレージシステムバックエンドを作成

Astra Trident Operator のインストールが完了したら、使用するネットアップストレージプラットフォームに合わせてバックエンドを設定する必要があります。次のリンクに従って、Astra Tridentのセットアップと設定を続けてください。

- ["NetApp ONTAP NFS"](#)
- ["NetApp ONTAP iSCSI の略"](#)

## NetApp ONTAP の NFS 構成

TridentをNFS経由でNetApp ONTAP ストレージシステムと統合するには、ストレージシステムとの通信を可能にするバックエンドを作成する必要があります。この解決策 では基本的なバックエンドを設定しますが、

よりカスタマイズされたオプションを探している場合は、のマニュアルを参照してください ["こちらをご覧ください"](#)。

## ONTAP でSVMを作成します

1. ONTAP System Managerにログインし、Storage > Storage VMの順に選択し、Addをクリックします。
2. SVMの名前を入力し、NFSプロトコルを有効にし、NFSクライアントアクセスを許可チェックボックスをオンにして、ワークロードクラスタ内でボリュームをPVSとしてマウントできるように、ワーカーノードがオンになっているサブネットをエクスポートポリシールールに追加します。

### Add Storage VM



STORAGE VM NAME

trident\_svm

### Access Protocol

☒ SMB/CIFS, NFS, S3

iSCSI

☐ Enable SMB/CIFS

☒ Enable NFS

☒ Allow NFS client access

Add at least one rule to allow NFS clients to access volumes in this storage VM. [?](#)

EXPORT POLICY

Default

RULES

Rule Index	Clients	Access Protocols	Read-Only Rule	Read/Wr
	0.0.0.0/0	Any	Any	Any



NSX -Tを使用したユーザクラスタまたはワークロードクラスタのNAT配置を使用する場合は、出力サブネット（TKGS0の場合はフローティングIPサブネット（TKGIの場合））をエクスポートポリシールールに追加する必要があります。

3. データLIFの詳細とSVM管理アカウントの詳細を指定し、保存をクリックします。

## NETWORK INTERFACE

Use multiple network interfaces when client traffic is high.

### K8s-Ontap-01

IP ADDRESS

172.21.252.180

SUBNET MASK

24

GATEWAY

172.21.252.1



BROADCAST DOMAIN

Default



## Storage VM Administration

☒ Manage administrator account

USER NAME

vsadmin

PASSWORD

.....

CONFIRM PASSWORD

.....

☐ Add a network interface for storage VM management.

4. アグリゲートをSVMに割り当てます。Storage > Storage VMsと進み、新しく作成したSVMの横にある省略記号をクリックして、Editをクリックします。ボリュームの作成を優先ローカル階層に制限するチェックボックスをオンにして、必要なアグリゲートを関連付けます。

## Edit Storage VM



STORAGE VM NAME

trident\_svm

DEFAULT LANGUAGE

c.utf\_8



DELETED VOLUME RETENTION PERIOD 

12

HOURS

## Resource Allocation

☒ Limit volume creation to preferred local tiers

LOCAL TIERS

K8s\_Ontap\_01\_SSD\_1 ×

Cancel

Save

5. Tridentをインストールするユーザまたはワークロードクラスタに対してNATを使用して配置した場合、ストレージマウント要求はSNATのために非標準ポートから到達する可能性があります。デフォルトでは、ONTAP は、ルートポートから作成されたボリュームマウント要求のみを許可します。したがっ



て、ONTAP CLIにログインし、非標準ポートからのマウント要求を許可する設定を変更してください。

```
ontap-01> vserver nfs modify -vserver tanzu_svm -mount-rootonly disabled
```

## バックエンドとStorageClassesを作成します

1. NFSを提供しているNetApp ONTAP システムの場合は、backendName、managementLIF、dataLIF、SVM、ユーザ名を指定してjumphostでバックエンド構成ファイルを作成します。パスワードなどの詳細情報。

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap-nas+10.61.181.221",
  "managementLIF": "172.21.224.201",
  "dataLIF": "10.61.181.221",
  "svm": "trident_svm",
  "username": "admin",
  "password": "password"
}
```



カスタムの backendName 値は、簡単に識別できるように NFS を提供するストレージ DriverName とデータ LIF を組み合わせて定義することを推奨します。

2. 次のコマンドを実行してTridentバックエンドを作成します。

```
[netapp-user@rhel7]$ ./tridentctl -n trident create backend -f backend-ontap-nas.json
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE | VOLUMES |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ontap-nas+10.61.181.221 | ontap-nas      | be7a619d-c81d-445c-b80c-5c87a73c5b1e |
| online |         | 0 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

3. バックエンドを作成したら、次にストレージクラスを作成する必要があります。次のストレージクラス定義の例では、必須フィールドと基本フィールドが強調表示されています。パラメータbackendTypeは'新しく作成されたTridentバックエンドのストレージ・ドライバを反映する必要があります

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nfs
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"

```

4. kubectlコマンドを実行して、ストレージクラスを作成します。

```

[netapp-user@rhel7 trident-installer]$ kubectl create -f storage-class-nfs.yaml
storageclass.storage.k8s.io/ontap-nfs created

```

5. ストレージクラスを作成したら、最初の永続的ボリューム要求（PVC）を作成する必要があります。PVC定義の例を次に示します。[storageClassName](ストレージクラス名)フィールドが作成したストレージクラスの名前と一致していることを確認しますプロビジョニングするワークロードに応じて、PVC定義を必要に応じてさらにカスタマイズできます。

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-nfs

```

6. kubectlコマンドを発行して、PVCを作成します。作成中の元のボリュームのサイズによっては作成にしばらく時間がかかることがあるため、作成が完了した時点でこのプロセスを監視できます。

```

[netapp-user@rhel7 trident-installer]$ kubectl create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ kubectl get pvc

```

NAME	STATUS	VOLUME	CAPACITY
basic	Bound	pvc-b4370d37-0fa4-4c17-bd86-94f96c94b42d	1Gi
ACCESS MODES		STORAGECLASS	AGE
RWO		ontap-nfs	7s

# NetApp ONTAP iSCSI 構成

NetApp ONTAP ストレージシステムをiSCSI経由で永続ボリューム用のVMware Tanzu Kubernetesクラスタと統合するには、まず各ノードにログインし、iSCSIボリュームをマウントするためのiSCSIユーティリティまたはパッケージを設定してノードを準備します。そのためには、この『手順』に記載されている手順に従ってください ["リンク"](#)。



NATによるVMware Tanzu Kubernetesクラスタの導入には、この手順 は推奨されません。



TKGIは、変更不可の構成イメージを実行するTanzu Kubernetesクラスタのノードとして、Bosh VMを使用します。また、Bosh VMでiSCSIパッケージを手動で変更しても、リブート後も維持されません。そのため、TKGIによって導入、運用されているTanzu Kubernetesクラスタの永続ストレージにはNFSボリュームを使用することを推奨します。

iSCSIボリュームのクラスタノードの準備が完了したら、ストレージシステムとの通信を可能にするバックエンドを作成する必要があります。この解決策 では基本的なバックエンドを設定しましたが、よりカスタマイズ可能なオプションを探している場合は、のドキュメントを参照してください ["こちらをご覧ください"](#)。

## ONTAP でSVMを作成します

ONTAP でSVMを作成するには、次の手順を実行します。

1. ONTAP System Managerにログインし、Storage > Storage VMの順に選択し、Addをクリックします。
2. SVMの名前を入力し、iSCSIプロトコルを有効にして、データLIFの詳細を指定します。

# Add Storage VM



STORAGE VM NAME

trident\_svm\_iscsi

## Access Protocol

SMB/CIFS, NFS, S3

iSCSI

☒ Enable iSCSI

NETWORK INTERFACE

### K8s-Ontap-01

IP ADDRESS

10.61.181.231

SUBNET MASK

24

GATEWAY

10.61.181.1

BROADCAST DOMAIN

Defa...

☐ Use the same subnet mask, gateway, and broadcast domain for all of the following interfaces

IP ADDRESS

10.61.181.232

SUBNET MASK

24

GATEWAY

10.61.181.1

BROADCAST DOMAIN

Defa...

3. SVM管理アカウントの詳細を入力し、保存をクリックします。

## Storage VM Administration

☒ Manage administrator account

USER NAME

vsadmin

PASSWORD

.....

CONFIRM PASSWORD

.....

☐ Add a network interface for storage VM management.

Save

Cancel

4. アグリゲートをSVMに割り当てるには、Storage > Storage VMに移動し、新しく作成したSVMの横にある省略記号をクリックしてEditをクリックします。ボリュームの作成を優先ローカル階層に制限するチェックボックスをオンにし、必要なアグリゲートを関連付けます。

## Edit Storage VM



STORAGE VM NAME

trident\_svm\_iscsi

DEFAULT LANGUAGE

c.utf\_8



DELETED VOLUME RETENTION PERIOD 

12

HOURS

## Resource Allocation

☒ Limit volume creation to preferred local tiers

LOCAL TIERS

K8s\_Ontap\_01\_SSD\_1 

Cancel

Save

### バックエンドと**StorageClasses**を作成します

1. NFSを提供しているNetApp ONTAP システムの場合は、backendName、managementLIF、dataLIF、SVM、ユーザ名を指定してjumpshostでバックエンド構成ファイルを作成します。パスワードなどの詳細情報。

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "ontap-san+10.61.181.231",
  "managementLIF": "172.21.224.201",
  "dataLIF": "10.61.181.231",
  "svm": "trident_svm_iscsi",
  "username": "admin",
  "password": "password"
}
```

2. 次のコマンドを実行してTridentバックエンドを作成します。

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-san.json
+-----+-----+
+-----+-----+-----+-----+
|           NAME           | STORAGE DRIVER |           UUID           |
| STATE | VOLUMES | |           |           |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ontap-san+10.61.181.231 | ontap-san      | 6788533c-7fea-4a35-b797- |
| fb9bb3322b91 | online |      0 | |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

3. バックエンドを作成したら、次にストレージクラスを作成する必要があります。次のストレージクラス定義の例では、必須フィールドと基本フィールドが強調表示されています。パラメータbackendTypeは'新しく作成されたTridentバックエンドのストレージ・ドライバを反映する必要がありますまた、名前フィールドの値もメモしておきます。この値は、以降の手順で参照する必要があります。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-iscsi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
```



このファイルに定義されているオプションのフィールド「fsType」があります。iSCSIバックエンドでは、この値を特定のLinuxファイルシステムタイプ（XFS、ext4など）に設定するか、またはTanzu Kubernetesクラスタが使用するファイルシステムを決定できるようにするために削除できます。

4. kubectlコマンドを実行して、ストレージクラスを作成します。

```
[netapp-user@rhel7 trident-installer]$ kubectl create -f storage-class-iscsi.yaml
storageclass.storage.k8s.io/ontap-iscsi created
```

5. ストレージクラスを作成したら、最初の永続的ボリューム要求（PVC）を作成する必要があります。PVC定義の例を次に示します。[storageClassName](ストレージクラス名)フィールドが作成したストレージクラスの名前と一致していることを確認します。プロビジョニングするワークロードに応じて、PVC定義を必要に応じてさらにカスタマイズできます。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-iscsi
```

6. kubectlコマンドを発行して、PVCを作成します。作成中の元のボリュームのサイズによっては作成にしばらく時間がかかることがあるため、作成が完了した時点でこのプロセスを監視できます。

```
[netapp-user@rhel7 trident-installer]$ kubectl create -f pvc-basic.yaml
persistentvolumeclaim/basic created
```

```
[netapp-user@rhel7 trident-installer]$ kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY
basic	Bound	pvc-7ceac1ba-0189-43c7-8f98-094719f7956c	1Gi
		ontap-iscsi	3s



## 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。