



ネットアップとストレージの統合の概要

NetApp Solutions

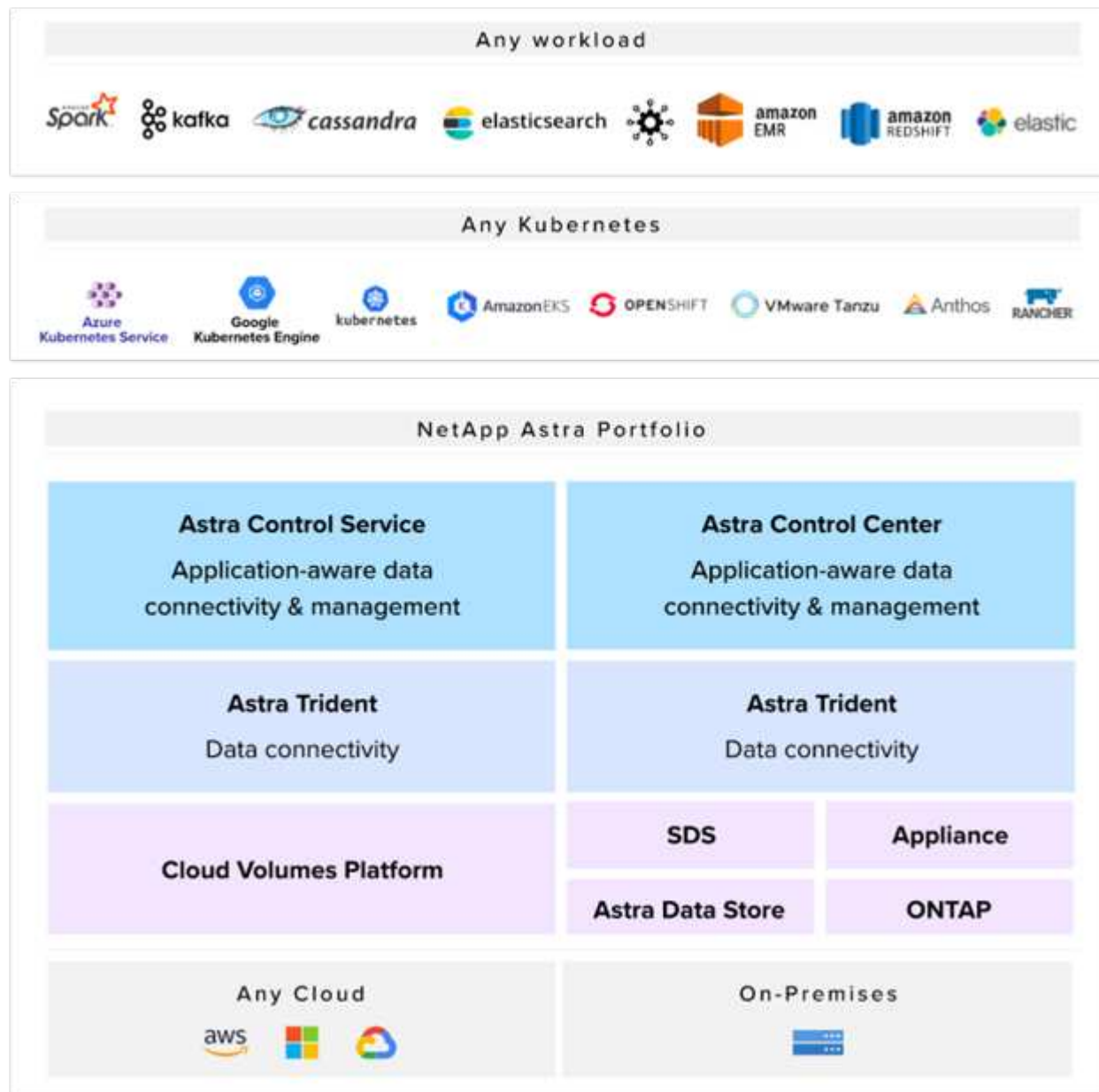
NetApp
April 10, 2024

目次

ネットアップストレージ統合の概要	1
NetApp Astra Controlの概要	2
Astra Tridentの概要	20

ネットアップストレージ統合の概要

ネットアップは、ステートフルなコンテナ化アプリケーションとそのデータのオーケストレーション、管理、保護、移行を支援するための製品を多数提供しています。



NetApp Astra Controlは、ネットアップのデータ保護テクノロジーを基盤とするステートフルKubernetesワークロード向けに、充実したストレージサービスとアプリケーション対応データ管理サービスを提供します。Astra Control Service は、クラウドネイティブの Kubernetes 環境でステートフルワークロードをサポートするために利用できます。Astra Control Centerは、{k8s_distribution_name} などのエンタープライズKubernetesプラットフォームをオンプレミスで導入する場合に、ステートフルワークロードをサポートするために使用できます。詳細については、NetApp Astra Control の Web サイトをご覧ください["こちらをご覧ください"](#)。

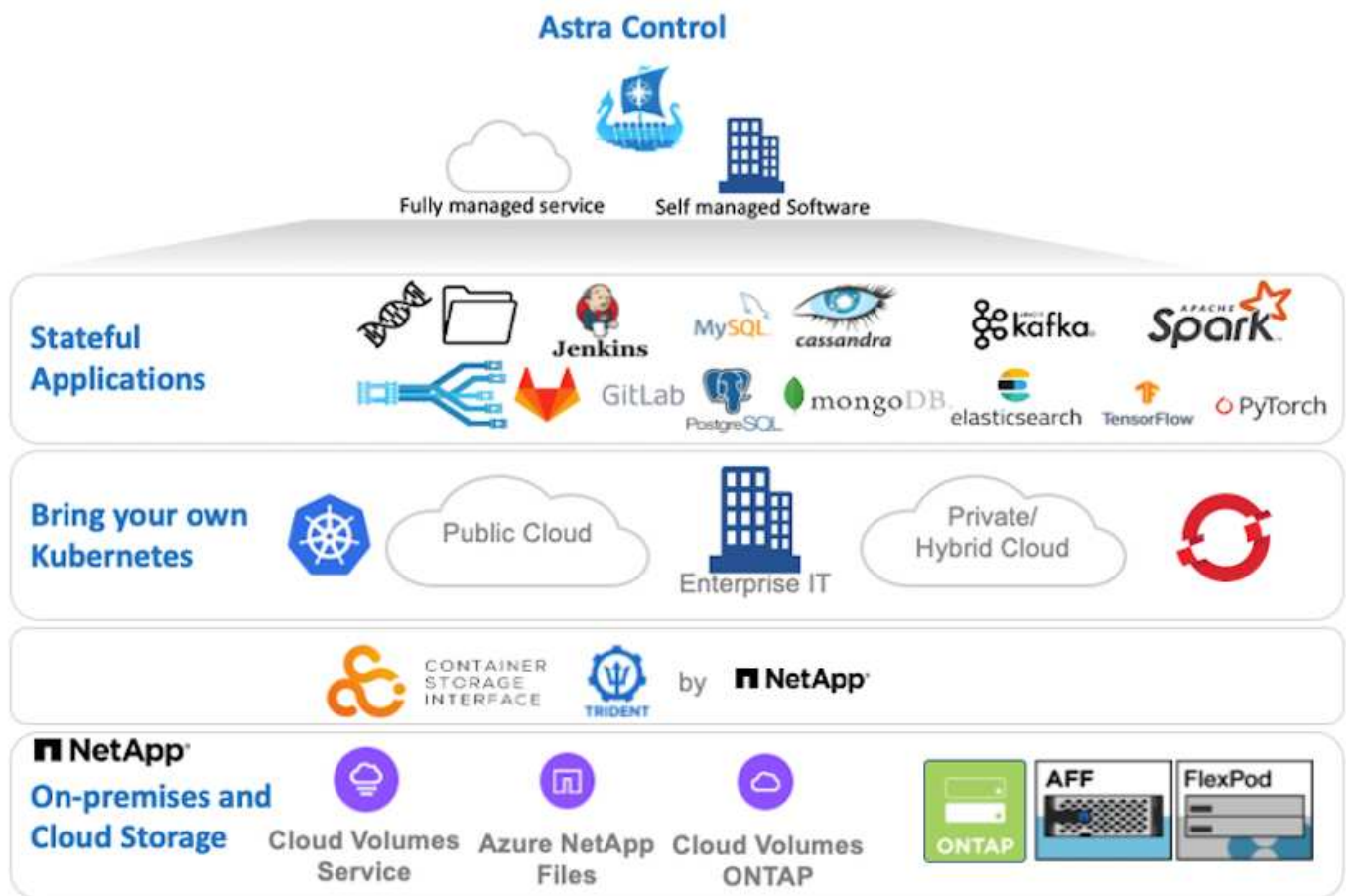
NetApp Astra Tridentは、コンテナ向けのオープンソースで完全にサポートされているストレージオーケストレーションツールであり、{k8s_distribution_name} などのKubernetesディストリビューションに対応しています。詳細については、Astra Trident の Web サイトをご覧ください["こちらをご覧ください"](#)。

次のページには、{solution_name} 解決策 でアプリケーションおよび永続的ストレージの管理用に検証されたネットアップ製品に関する追加情報 があります。

- "ネットアップアストラコントロールセンター"
- "ネットアップアストラ Trident"

NetApp Astra Controlの概要

NetApp Astra Control Center は、オンプレミス環境に導入され、ネットアップのデータ保護テクノロジーを基盤とするステートフル Kubernetes ワークロード向けの充実したストレージサービスとアプリケーション対応データ管理サービスを提供します。



NetApp Astra Control Centerは、Astra Tridentストレージオーケストレーションツールを導入して、NetApp ONTAP ストレージシステムにストレージクラスとストレージバックエンドで構成されている {k8s_distribution_name} クラスタにインストールできます。

Astra Tridentの詳細については、を参照してください ["このドキュメントはこちら"](#)。

クラウド接続環境では、Cloud Insights を使用して高度なモニタリングとテレメトリを提供します。Cloud Insights 接続がない場合は、限定的な監視と計測（7日間の指標）を使用でき、オープン指標エンドポイントを通じてKubernetesの標準の監視ツール（PrometheusとGrafana）にエクスポートされます。

Astra Control Center は、ネットアップの AutoSupport と Active IQ のエコシステムに完全に統合されており、ユーザをサポートし、トラブルシューティングを支援し、使用状況の統計を表示します。

支払い済みのAstra Control Centerに加えて、90日間の評価ライセンスも利用できます。評価版は、EメールとSlackコミュニティチャンネルを通じてサポートされます。お客様は、これらのリソース、その他のナレッジベース記事、および製品サポートダッシュボードから入手できるドキュメントにアクセスできます。

Astraポートフォリオの詳細については、を参照してください ["Astra の Web サイト"](#)。

Astra Control Center自動化

Astra Control Centerには、プログラム経由でアクセスするための完全に機能するREST APIが用意されています。任意のプログラミング言語またはユーティリティを使用して、Astra Control REST APIエンドポイントとやり取りできます。このAPIの詳細については、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"](#)。

すぐに利用できる、Astra Control REST APIと連携するためのソフトウェア開発ツールキットを探している場合、ネットアップはAstra Control Python SDKのツールキットを提供しています。このツールキットはダウンロードが可能です ["こちらをご覧ください"](#)。

プログラミングが適していない状況で構成管理ツールを使用する場合は、ネットアップが公開しているAnsibleプレイブックをクローニングして実行できます ["こちらをご覧ください"](#)。

Astra Control Center のインストールの前提条件

Astra Control Centerのインストールには、次の前提条件が必要です。

- 1つ以上のTanzu Kubernetesクラスタは、管理クラスタまたはTKGSまたはTKGIによって管理されます。TKGワークロードクラスタ1.4 +およびTKGIユーザークラスタ1.12.2+がサポートされています。
- 各Tanzu KubernetesクラスタにAstra Tridentがインストールおよび設定されている必要があります。
- ONTAP 9.5 以降を実行している NetApp ONTAP ストレージシステムが 1 つ以上必要です。



サイトにある各Tanzu Kubernetesインストールでは、永続的ストレージ用の専用SVMを使用することを推奨します。マルチサイト環境では、追加のストレージシステムが必要です。

- Tridentストレージバックエンドは、ONTAP クラスタから作成されたSVMを含む各Tanzu Kubernetesクラスタで設定する必要があります。
- 各Tanzu Kubernetesクラスタに設定されたデフォルトのStorageClassには、Astra Tridentをストレージプロビジョニングツールとして使用します。
- ingressType 「AccTraefik」 を使用している場合は、ロードバランシングとアストラコントロールセンターの公開のために、各Tanzu Kubernetesクラスタにロードバランサをインストールし、設定する必要があります。
- ingressType 「Generic」 を使用している場合は、Astra Control Centerを公開するために、各Tanzu Kubernetesクラスタに入力コントローラをインストールし、設定する必要があります。
- NetApp アストラ Control Center イメージをホストするには、プライベートイメージのレジストリを設定する必要があります。
- Astra Control CenterをインストールしているTanzu Kubernetesクラスタにクラスタ管理者としてアクセスできる必要があります。
- NetApp ONTAP クラスタへの管理者アクセスが必要です。

- RHELまたはUbuntuの管理ワークステーション。

Astra Control Center をインストールします

この解決策 では、Ansibleプレイブックを使用してAstra Control Centerをインストールするための自動手順 について説明します。手順 を手動でインストールしてAstra Control Centerをインストールする場合は、詳細なインストールと操作のガイドに従ってください "[こちらをご覧ください](#)"。

1. Astra Control Centerを導入するAnsibleプレイブックを使用するには、AnsibleがインストールされたUbuntu / RHELマシンが必要です。手順に従います "[こちらをご覧ください](#)" UbuntuおよびRHELの場合。
2. Ansible コンテンツをホストする GitHub リポジトリをクローニングします。

```
git clone https://github.com/NetApp-
Automation/na_astra_control_suite.git
```

3. ネットアップサポートサイトにログインし、最新バージョンの NetApp Astra Control Center をダウンロードします。そのためには、ネットアップアカウントにライセンスを関連付ける必要があります。tar ファイルをダウンロードしたら、ワークステーションに転送します。



Astra Control の試用版ライセンスの使用を開始するには、にアクセスしてください "[Astra 登録サイト](#)"。

4. Astra Control CenterをインストールするユーザまたはワークロードのTanzu Kubernetesクラスタに管理者アクセスでkubeconfigファイルを作成または取得します。
5. ディレクトリを'na_Astra_control_site'に変更します

```
cd na_astra_control_suite
```

6. 「vars/vars.yml」 ファイルを編集し、必要な情報を変数に入力します。

```
#Define whether or not to push the Astra Control Center images to your
private registry [Allowed values: yes, no]
push_images: yes

#The directory hosting the Astra Control Center installer
installer_directory: /home/admin/

#Specify the ingress type. Allowed values - "AccTraefik" or "Generic"
#"AccTraefik" if you want the installer to create a LoadBalancer type
service to access ACC, requires MetalLB or similar.
#"Generic" if you want to create or configure ingress controller
yourself, installer just creates a ClusterIP service for traefik.
ingress_type: "AccTraefik"
```

```
#Name of the Astra Control Center installer (Do not include the
extension, just the name)
astra_tar_ball_name: astra-control-center-22.04.0

#The complete path to the kubeconfig file of the kubernetes/openshift
cluster Astra Control Center needs to be installed to.
hosting_k8s_cluster_kubeconfig_path: /home/admin/cluster-kubeconfig.yml

#Namespace in which Astra Control Center is to be installed
astra_namespace: netapp-astra-cc

#Astra Control Center Resources Scaler. Leave it blank if you want to
accept the Default setting.
astra_resources_scaler: Default

#Storageclass to be used for Astra Control Center PVCs, it must be
created before running the playbook [Leave it blank if you want the PVCs
to use default storageclass]
astra_trident_storageclass: basic

#Reclaim Policy for Astra Control Center Persistent Volumes [Allowed
values: Retain, Delete]
storageclass_reclaim_policy: Retain

#Private Registry Details
astra_registry_name: "docker.io"

#Whether the private registry requires credentials [Allowed values: yes,
no]
require_reg_creds: yes

#If require_reg_creds is yes, then define the container image registry
credentials
#Usually, the registry namespace and usernames are same for individual
users
astra_registry_namespace: "registry-user"
astra_registry_username: "registry-user"
astra_registry_password: "password"

#Kubereneets/OpenShift secret name for Astra Control Center
#This name will be assigned to the K8s secret created by the playbook
astra_registry_secret_name: "astra-registry-credentials"

#Astra Control Center FQDN
acc_fqdn_address: astra-control-center.cie.netapp.com

#Name of the Astra Control Center instance
```

```
acc_account_name: ACC Account Name

#Administrator details for Astra Control Center
admin_email_address: admin@example.com
admin_first_name: Admin
admin_last_name: Admin
```

7. プレイブックを実行して Astra Control Center を導入します。Playbookには、特定の構成用のroot権限が必要です。

プレイブックを実行しているユーザがrootである場合、またはパスワードなしのsudoが設定されている場合は、次のコマンドを実行してプレイブックを実行します。

```
ansible-playbook install_acc_playbook.yml
```

ユーザにパスワードベースのsudoアクセスが設定されている場合は、次のコマンドを実行してこのPlaybookを実行し、sudoパスワードを入力します。

```
ansible-playbook install_acc_playbook.yml -K
```

インストール後の手順

1. インストールが完了するまでに数分かかることがあります。NetApp-AstrA-cc' ネームスペース内のすべてのポッドとサービスが稼働していることを確認します

```
[netapp-user@rhel7 ~]$ kubectl get all -n netapp-astra-cc
```

2. 「acc-operator-controller-manager」ログをチェックし、インストールが完了したことを確認します。

```
[netapp-user@rhel7 ~]$ kubectl logs deploy/acc-operator-controller-  
manager -n netapp-acc-operator -c manager -f
```



次のメッセージは、Astra Control Center のインストールが正常に完了したことを示します。

```
{"level":"info","ts":1624054318.029971,"logger":"controllers.AstraControlCenter","msg":"Successfully Reconciled AstraControlCenter in [seconds]s","AstraControlCenter":"netapp-astra-cc/astra","ae.Version":"[22.04.0]"}
```

3. Astra Control Center にログインするためのユーザ名は、CRD ファイルに提供された管理者の電子メール

アドレスで、パスワードは Astra Control Center UUID に付加された文字列「ACC-」です。次のコマンドを実行します。

```
[netapp-user@rhel7 ~]$ oc get astracontrolcenters -n netapp-astra-cc
NAME      UUID
astra     345c55a5-bf2e-21f0-84b8-b6f2bce5e95f
```



この例では、パスワードは「ACC-345c55a5-bf2e-21f0-84b8-b6f2bce5e95f」です。

4. ingressTypeがAccTraefikの場合は、traefikサービスロードバランサIPを取得します。

```
[netapp-user@rhel7 ~]$ oc get svc -n netapp-astra-cc | egrep
'EXTERNAL|traefik'
```

NAME	EXTERNAL-IP	PORT(S)	TYPE	CLUSTER-IP
traefik	10.61.186.181	80:30343/TCP, 443:30060/TCP	LoadBalancer	172.30.99.142
AGE		16m		

5. Astra Control Center CRD ファイルに指定された FQDN を指す DNS サーバーのエントリを、traefik サービスの「external-IP」に追加します。

New Host

Name (uses parent domain name if blank):
astra-control-center

Fully qualified domain name (FQDN):
astra-control-center.cie.netapp.com.

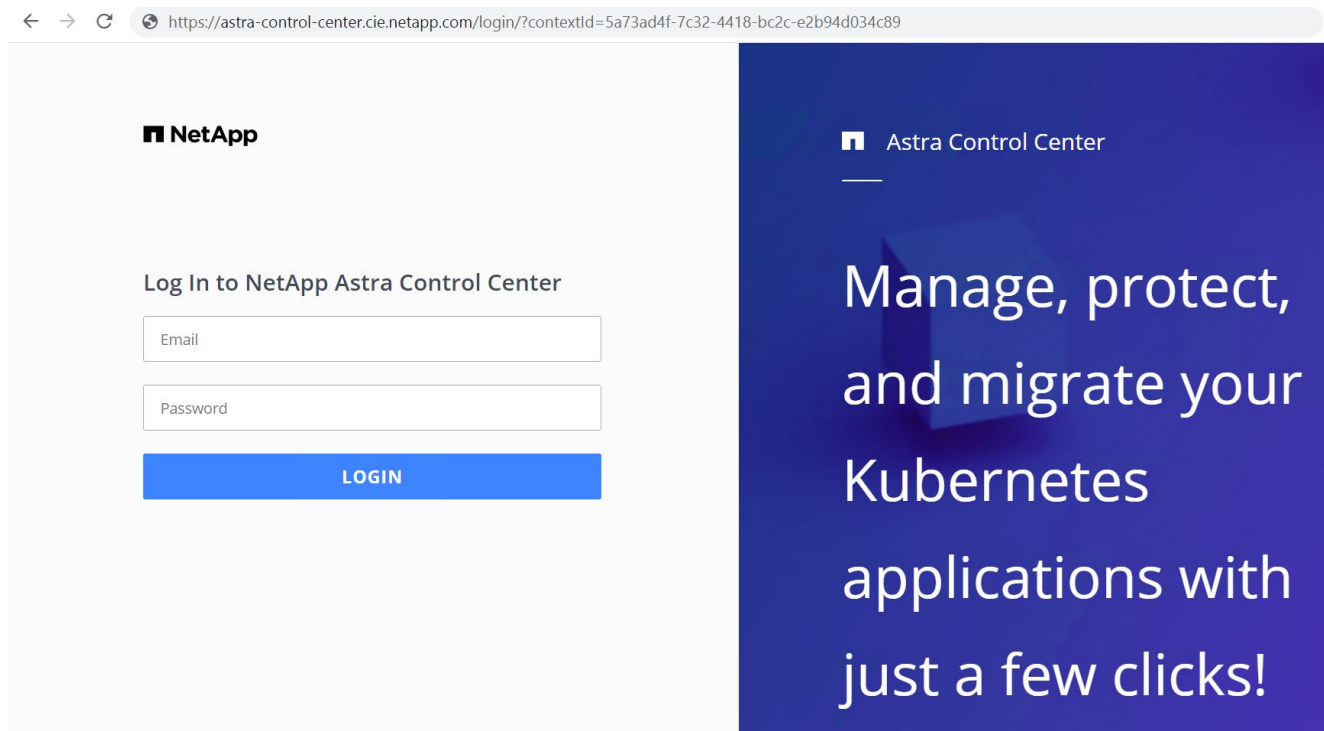
IP address:
10.61.186.181

☒ Create associated pointer (PTR) record

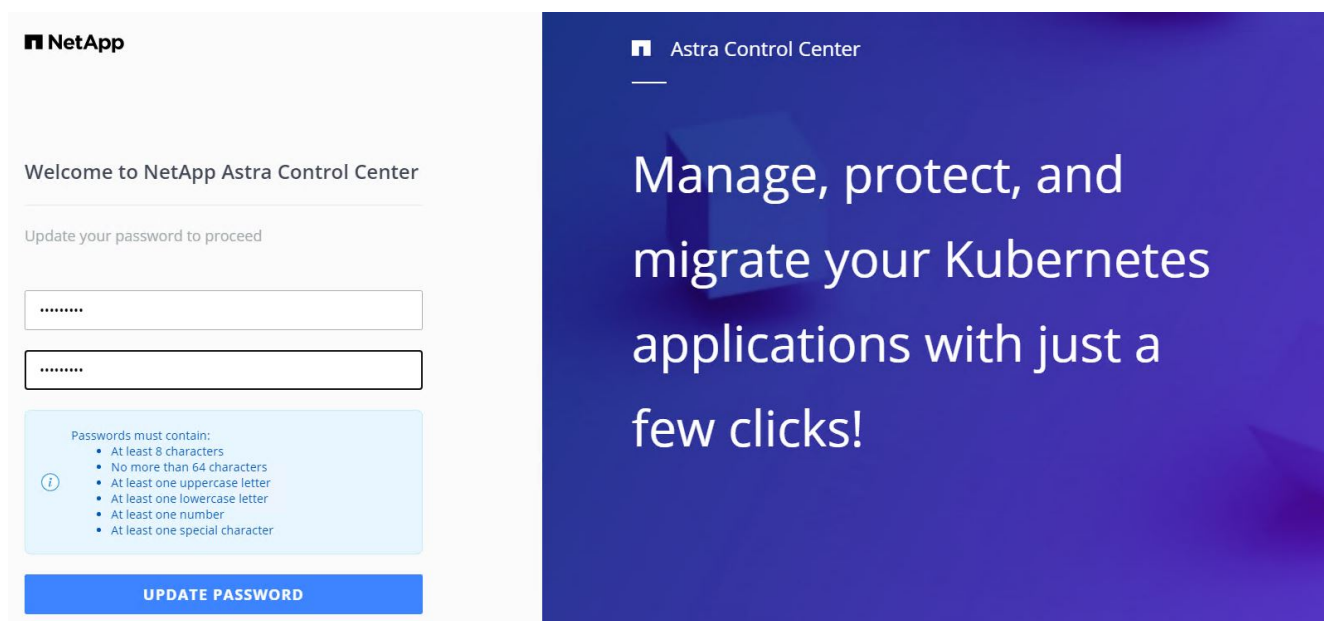
☐ Allow any authenticated user to update DNS records with the same owner name

Add Host Cancel

6. Astra Control Center GUI に、FQDN を参照してログインします。

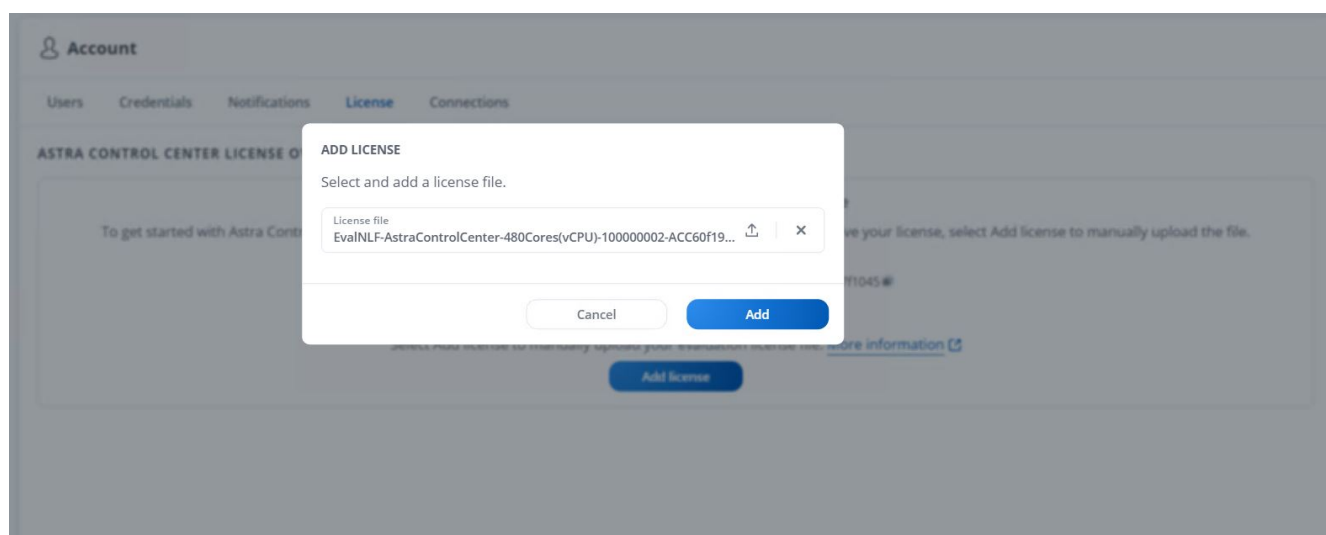


7. CRD で提供された管理者メールアドレスを使用して初めて Astra Control Center GUI にログインする場合は、パスワードを変更する必要があります。



8. ユーザーを Astra Control Center に追加する場合は、[アカウント]>[ユーザー] の順に選択し、[追加] をクリックしてユーザーの詳細を入力し、[追加] をクリックします。

9. Astra Control Centerのすべての機能が動作するには、ライセンスが必要です。ライセンスを追加するには、[アカウント] > [ライセンス] の順に選択し、[ライセンスの追加] をクリックして、ライセンスファイルをアップロードします。




NetApp Astra Control Center のインストールまたは設定で問題が発生した場合は、既知の問題のナレッジベースを利用できます ["こちらをご覧ください"](#)。

VMware Tanzu Kubernetes クラスタを Astra Control Center に登録します

Astra Control Center でワークロードを管理できるようにするには、まず Tanzu Kubernetes クラスタを登録する必要があります。

VMware Tanzu Kubernetes クラスタを登録します

1. 最初の手順は、Tanzu Kubernetes クラスタを Astra Control Center に追加して管理することです。[クラスタ] に移動して [クラスタの追加] をクリックし、Tanzu Kubernetes クラスタの kubeconfig ファイルをアップロードして、[ストレージの選択] をクリックします。

 **Add Kubernetes cluster**

STEP 1/3: CREDENTIALS

×

CREDENTIALS

Provide Astra Control access to your Kubernetes and OpenShift clusters by entering a kubeconfig credential.

Follow [instructions](#) on how to create a dedicated admin-role kubeconfig.

[Upload file](#) [Paste from clipboard](#)


Kubeconfig YAML file
tkgi-kubeconfig.txt

↑ ×

Credential name
tkgi-acc

Cancel

Next →


 **ADDING CLUSTERS**

Adding a cluster allows Astra Control to install its storage services, and enable data management operations on your containerized applications.

For more details on required versions or cloud specific setup refer to the documentation.

Read more in [Adding clusters](#).

2. Astra Control Center で、対象となるストレージクラスが検出される。次に、ストレージクラスが NetApp ONTAP 上の SVM がサポートする Trident を使用してボリュームをプロビジョニングする方法を選択し、Review（確認）をクリックします。次のペインで詳細を確認し、Add Cluster をクリックします。
3. クラスタが追加されると、Astra Control Center がクラスタを検査し、必要なエージェントをインストールしながら、クラスタは Discovering ステータスに移行します。正常に登録されると、クラスタ・ステータスは「Healthy」に変わります。



 **Clusters**

Actions ▾

+ Add Kubernetes cluster

Search

1-1 of 1 entries < >

<input type="checkbox"/>	Name ↓	State	Type	Version	Actions
<input type="checkbox"/>	tkgi-acc	Healthy	 Kubernetes	v1.22.6+vmware.1	



Astra Control Center で管理するすべての Tanzu Kubernetes クラスタは、管理対象クラスタにインストールされたエージェントとしてそのインストールに使用されたイメージレジストリにアクセスする必要があります。このレジストリからイメージがプルされます。

4. ONTAP クラスタをストレージリソースとして Astra Control Center でバックエンドとして管理するようにインポートします。Tanzu Kubernetes クラスタが Astra に追加され、ストレージクラスが設定されている

場合、ストレージクラスをサポートするONTAP クラスタは自動的に検出されて検査されますが、管理対象のAstraコントロールセンターにはインポートされません。

Backends

+

Add

Search

★

Q

1

1-1 of 1 entries

<

>

Name ↓	State	Capacity	Throughput	Type	Cluster	Cloud	Actions
172.21.224.201(trident)	<div><div></div><div>Discovered</div></div>	Not available yet	Not available yet	ONTAP	Not applicable	Not applicable	<div><div></div></div>

5. ONTAP クラスタをインポートするには、バックエンドに移動し、ドロップダウンをクリックして、管理対象のONTAP クラスタの横にあるManageを選択します。ONTAP クラスタの資格情報を入力し、[情報の確認]をクリックして、[ストレージバックエンドのインポート]をクリックします。

Manage ONTAP storage backend

STEP 1/2: CREDENTIALS

X

CREDENTIALS

Enter cluster administrator credentials for the ONTAP storage backend you want to manage.

Cluster management IP address
172.21.224.201

User name
admin

Password

MANAGING STORAGE BACKENDS

Storage backends provide storage to your Kubernetes applications.

Managing storage clusters in Astra Control as a storage backend will allow you to get linkages between PVs and the storage backend. You will also see capacity and health details of the storage backend, including performance metrics if Astra Control is connected to Cloud Insights.

Read more in [Storage type](#).

ONTAP

Cancel

Next →


6. バックエンドを追加すると、ステータスが Available に変わります。これらのバックエンドには、Tanzu Kubernetesクラスタ内の永続ボリュームおよびONTAP システム上の対応するボリュームに関する情報が含まれるようになりました。

12

Backends


<

7. Astra Control Centerを使用してTanzu Kubernetesクラスタ間でバックアップおよびリストアを実行するには、S3プロトコルをサポートするオブジェクトストレージバケットをプロビジョニングする必要があります。現在サポートされているオプションは、ONTAP S3、StorageGRID、AWS S3、およびMicrosoft Azure Blob Storageです。このインストールのために、AWS S3 バケットを設定します。バケットに移動し、バケットの追加をクリックして、汎用 S3 を選択します。S3バケットとクレデンシャルの詳細を入力してアクセスし、Make this Bucket the Default Bucket for the Cloud（このバケットをクラウドのデフォルトバケットにする）チェックボックスをオンにして、Add（追加）をクリックします。

 **Add bucket**

Enter the access details of your existing object store bucket to allow Astra Control to store your application backups.

Type

 Generic S3

Existing bucket name

na-tanzu-astra/na-astra-tkgi

Description (optional)

S3 server name or IP address

s3.us-east-1.amazonaws.com

☒ Make this bucket the default bucket for this cloud

?

SELECT CREDENTIALS

Astra Control requires S3 access credentials with the roles necessary to facilitate Kubernetes application data management.


Add

Use existing

Select credential

AWS Creds

Cancel

Add 

BUCKETS

Astra Control stores backups in your existing object store buckets. The first bucket added for a selected cloud will be designated as the default bucket for backup and clone operations.

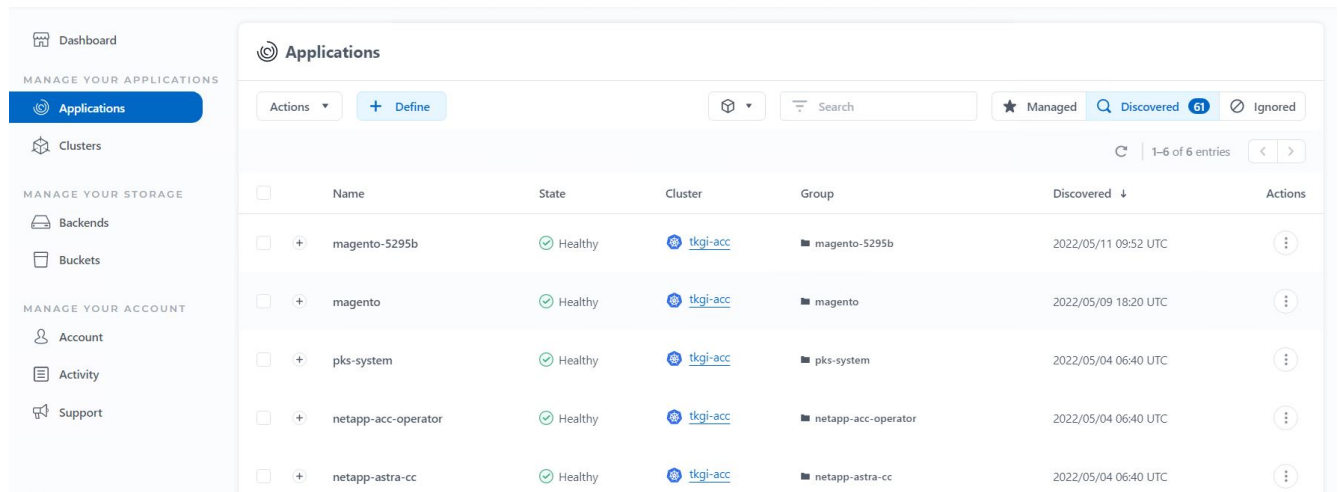
Read more in [Storage buckets](#).

保護するアプリケーションを選択します

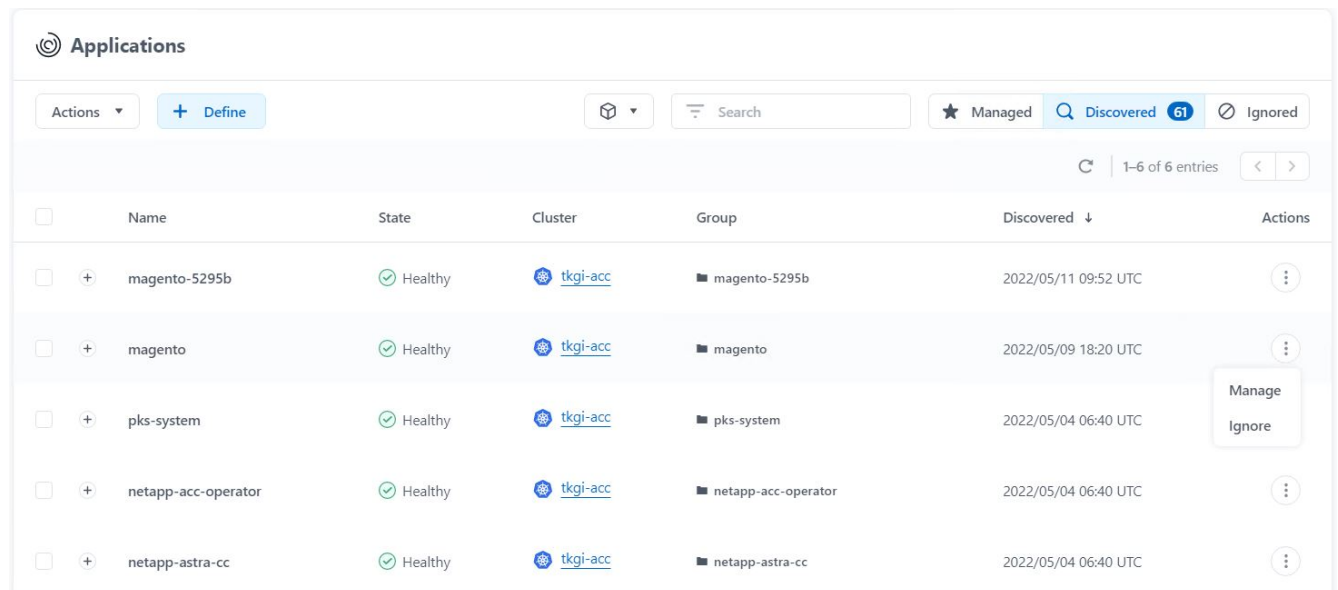
Tanzu Kubernetesクラスタを登録したら、Astra Control Centerを使用して導入および管理されているアプリケーションを検出できます。

アプリケーションを管理します

1. Tanzu KubernetesクラスタとONTAP バックエンドがAstraコントロールセンターに登録されると、コントロールセンターは指定されたONTAP バックエンドで構成されたストレージクラスを使用しているすべてのネームスペース内のアプリケーションを自動的に検出します。



2. [アプリケーション] > [検出済み] の順に移動し、Astra を使用して管理するアプリケーションの横にあるドロップダウンメニューをクリックします。[管理] をクリックします。



3. アプリケーションが[使用可能（Available）] 状態になり、[アプリケーション（Apps）] セクションの[管理（Managed）] タブで表示できます。

Applications

Actions ▾

+ Define

All clusters ▾

⌵

Search

★ Managed

🔍

Discovered

60

🚫

Ignored

↺

1-1 of 1 entries

⏪

⏩

<input type="checkbox"/>	Name	State	Protection	Cluster	Group	Discovered ↓	Actions
<input type="checkbox"/>	magento	<div>🟢</div> Healthy	<div>⚠️</div> Unprotected	<div>🌐</div> tkgi-acc	<div>📁</div> magento	2022/05/09 18:20 UTC	<div>⋮</div>

アプリケーションを保護

アプリケーションワークロードを Astra Control Center で管理した後、それらのワークロードの保護設定を構成できます。

アプリケーションスナップショットを作成します

アプリケーションのSnapshotコピーを作成すると、ONTAP Snapshotコピーとアプリケーションメタデータのコピーが作成されます。このコピーを使用して、アプリケーションを特定の時点の状態にリストアまたはクローニングできます。

1. アプリケーションのスナップショットを作成するには、[アプリ] > [管理] タブに移動し、Snapshot コピーを作成するアプリケーションをクリックします。アプリケーション名の横にあるドロップダウンメニューをクリックし、Snapshot をクリックします。

APPLICATION STATUS

Healthy

APPLICATION PROTECTION STATUS

Unprotected

Images

docker.io/bitnami/elasticsearch:6.8.12-debian-10-r61
 docker.io/bitnami/magento:2.4.1-debian-10-r14
 docker.io/bitnami/mariadb:10.3.24-debian-10-r49

Protection schedule

Disabled

Group

magento

Cluster

[tkgi-acc](#)

Actions ▾

Snapshot
 Backup
 Clone
 Restore
 Unmanage

2. スナップショットの詳細を入力し、[次へ] をクリックして、[スナップショット] をクリックします。Snapshot の作成には約 1 分かかり、作成が完了するとステータスを確認できるようになります。

Snapshot namespace application

STEP 1/2: DETAILS

×

SNAPSHOT DETAILS

Name

magento-snapshot-20220516212403

CREATING APPLICATION SNAPSHOTS

Astra Control can take a quick snapshot of your application configuration and persistent storage. Enter a snapshot name to get started.

Read more in [Protect apps](#)

Namespace application
magento

Namespace
magento

Cluster
tkgi-acc

Cancel

Next →

アプリケーションのバックアップを作成します

アプリケーションのバックアップは、アプリケーションのアクティブな状態とそのリソースの設定をキャプチャしてファイルに変換し、リモートのオブジェクトストレージバケットに格納します。

1. Astra Control Center で管理対象アプリケーションのバックアップとリストアを行うには、バックアップ ONTAP システムのスーパーユーザ設定を前提条件として設定する必要があります。そのためには、次のコマンドを入力します。

```
ONTAP::> export-policy rule modify -vserver ocp-trident -policyname default -ruleindex 1 -superuser sys
ONTAP::> export-policy rule modify -policyname default -ruleindex 1 -anon 65534 -vserver ocp-trident
```

2. Astra Control Center で管理対象アプリケーションのバックアップを作成するには、[アプリ] > [管理] タブに移動し、バックアップを作成するアプリケーションをクリックします。アプリケーション名の横にあるドロップダウンメニューをクリックし、[バックアップ] をクリックします。

magento

Actions

Snapshot

Backup

Clone

Restore

Unmanage

APPLICATION STATUS

Healthy

APPLICATION PROTECTION STATUS

Unprotected

Images
docker.io/bitnami/elasticsearch:6.8.12-debian-10-r61
docker.io/bitnami/magento:2.4.1-debian-10-r14
docker.io/bitnami/mariadb:10.3.24-debian-10-r49


Protection schedule
Disabled

Group
 magento


Cluster
 tkgi-acc

3. バックアップの詳細を入力し、バックアップファイルを保存するオブジェクトストレージバケットを選択して次へをクリックします。詳細を確認したら、バックアップをクリックします。アプリケーションのサイズとデータによっては、バックアップに数分かかることがあり、バックアップが正常に完了したあとで

バックアップのステータスを確認できるようになります。

 **Back up namespace application**

STEP 1/2: DETAILS



BACKUP DETAILS

Name
magento-backup-20220516212622

☐ Back up from an existing snapshot ?


BACKUP DESTINATION


Bucket
na-tanzu-astra/na-astra-tkgi Available Default


CREATING APPLICATION BACKUPS

Astra Control can take a backup of your application configuration and persistent storage. Persistent storage backups are transferred to your object store. Enter a backup name to get started.

Read more in [Application backups](#).

 Namespace application
magento

 Namespace
magento

 Cluster
tkgi-acc


Cancel


Next →

アプリケーションのリストア

ボタンを押すだけで、アプリケーションを同じクラスタ内の元のネームスペースまたはリモートクラスタにリストアし、アプリケーションを保護してディザスタリカバリに使用できます。

1. アプリケーションを復元するには、[アプリ]>[管理]タブに移動し、該当するアプリをクリックします。アプリケーション名の横にあるドロップダウンメニューをクリックし、[復元]をクリックします。

 **magento**



Actions


Snapshot

Backup


Clone

Restore

Unmanage

 APPLICATION STATUS

Healthy

 APPLICATION PROTECTION STATUS

Unprotected

Images

docker.io/bitnami/elasticsearch:6.8.12-debian-10-r61
docker.io/bitnami/magento:2.4.1-debian-10-r14
docker.io/bitnami/mariadb:10.3.24-debian-10-r49


Protection schedule

Disabled

Group

■ magento

Cluster

 tkgi

2. リストアネームスペースの名前を入力し、リストア先のクラスタを選択して、既存の Snapshot からリストアするかアプリケーションのバックアップからリストアするかを選択します。次へをクリックします。

Restore namespace application

STEP 1/2: DETAILS

✕

RESTORE DETAILS

Destination cluster

tkgi-acc

Destination namespace

magento

RESTORE SOURCE

Filter

Snapshots

Backups

Application backup	State	On-Schedule/On-Demand	Created ↑
<input type="radio"/> magento-backup-20220516212730	<input checked="" type="checkbox"/> Healthy	<input checked="" type="checkbox"/> On-Demand	2022/05/16 21:27 UTC

RESTORING APPLICATIONS

Astra Control can restore your application configuration and persistent storage. Select a source snapshot or backup for the restored application.

- Namespace application magento
- Namespace magento
- Cluster tkgi-acc

Cancel

Next →

- レビューペインで「restore」と入力し、詳細を確認した後で「Restore」をクリックします。

Restore namespace application

STEP 2/2: SUMMARY

✕

REVIEW RESTORE INFORMATION

All existing resources associated with this namespace application will be deleted and replaced with the source backup "magento-backup-20220516212730" taken on 2022/05/16 21:27 UTC. Persistent volumes will be deleted and recreated. External resources with dependencies on this namespace application might be impacted.

We recommend taking a snapshot or a backup of your namespace application before proceeding.

BACKUP

magento-backup-20220516212730

ORIGINAL GROUP

magento

ORIGINAL CLUSTER

tkgi-acc

RESOURCE LABELS

Config Maps

app.kubernetes.io/name: elasticsearch +9

Deployments

RESTORE

magento

DESTINATION GROUP

magento

DESTINATION CLUSTER

tkgi-acc

RESOURCE LABELS

Config Maps

app.kubernetes.io/name: elasticsearch +9

Deployments

Are you sure you want to restore the namespace application "magento"?

Type restore below to confirm.

Confirm to restore

restore

← Back

Restore ✓

- 新しいアプリケーションは、Astra Control Center が選択したクラスタ上のアプリケーションを復元している間、Restoring 状態になります。アプリケーションのすべてのリソースが Astra によってインストールおよび検出されると、アプリケーションは Available 状態になります。

<div> <div>Applications</div> <div> <div>Actions</div> <div>+ Define</div> <div>All clusters</div> <div>Search</div> <div>Managed</div> <div>Discovered 60</div> <div>Ignored</div> </div> </div>						
<div> <div>1-1 of 1 entries</div> <div>< ></div> </div>						
<input type="checkbox"/>	Name	State	Protection	Cluster	Group	Discovered ↓
<input type="checkbox"/>	magento	Healthy	Unprotected	tkgi-acc	magento	2022/05/09 18:20 UTC

アプリケーションのクローニング

アプリケーションは、開発 / テストやアプリケーションの保護およびディザスタリカバリ目的で、元のクラスタまたはリモートクラスタにクローニングできます。同じストレージバックエンドで同じクラスタ内にあるアプリケーションをクローニングする場合、NetApp FlexClone テクノロジを使用します。FlexClone テクノロジを使用すると、PVC のクローンを瞬時に作成し、ストレージスペースを節約できます。

1. アプリケーションをクローンするには、[アプリケーション (Apps)] > [管理 (Managed)] タブに移動し、該当するアプリケーションをクリックします。アプリケーション名の横にあるドロップダウンメニューをクリックし、Clone をクリックします。

magento

APPLICATION STATUS

Healthy

APPLICATION PROTECTION STATUS

Unprotected

Images

docker.io/bitnami/elasticsearch:6.8.12-debian-10-r61

docker.io/bitnami/magento:2.4.1-debian-10-r14

docker.io/bitnami/mariadb:10.3.24-debian-10-r49

Protection schedule

Disabled

Group

magento

Cluster

tkgi-acc

Actions

Snapshot

Backup

Clone

Restore

Unmanage

2. 新しいネームスペースの詳細を入力し、クローニング先のクラスタを選択します。クローンを既存のSnapshotから作成するか、バックアップから作成するか、アプリケーションの現在の状態から作成するかを選択します。詳細を確認したら、[次へ]をクリックし、確認ペインの[複製]をクリックします。

Clone namespace application

STEP 1/2: DETAILS

✕

CLONE DETAILS

Clone namespace
magento-bef7f

Destination cluster
tkgi-acc

☐ Clone from an existing snapshot or backup

CLONING APPLICATIONS

Astra Control can create a clone of your application configuration and persistent storage. Persistent storage backups are transferred from your object store, so choosing a clone from an existing backup will complete the fastest. Enter a clone name to get started.

Not all applications may support cloning.

Read more in [Clone applications](#).

- Namespace application
magento
- Namespace
magento
- Cluster
tkgi-acc

Cancel

Next →

3. 新しいアプリケーションは Discovering 状態になり、Astra Control Center は選択したクラスタにアプリケーションを作成します。アプリケーションのすべてのリソースが Astra によってインストールおよび検出されると、アプリケーションは Available 状態になります。

Applications

Actions ▾

+ Define

All clusters ▾

Search

★ Managed

Q Discovered 60

Ignored

1-2 of 2 entries

< >

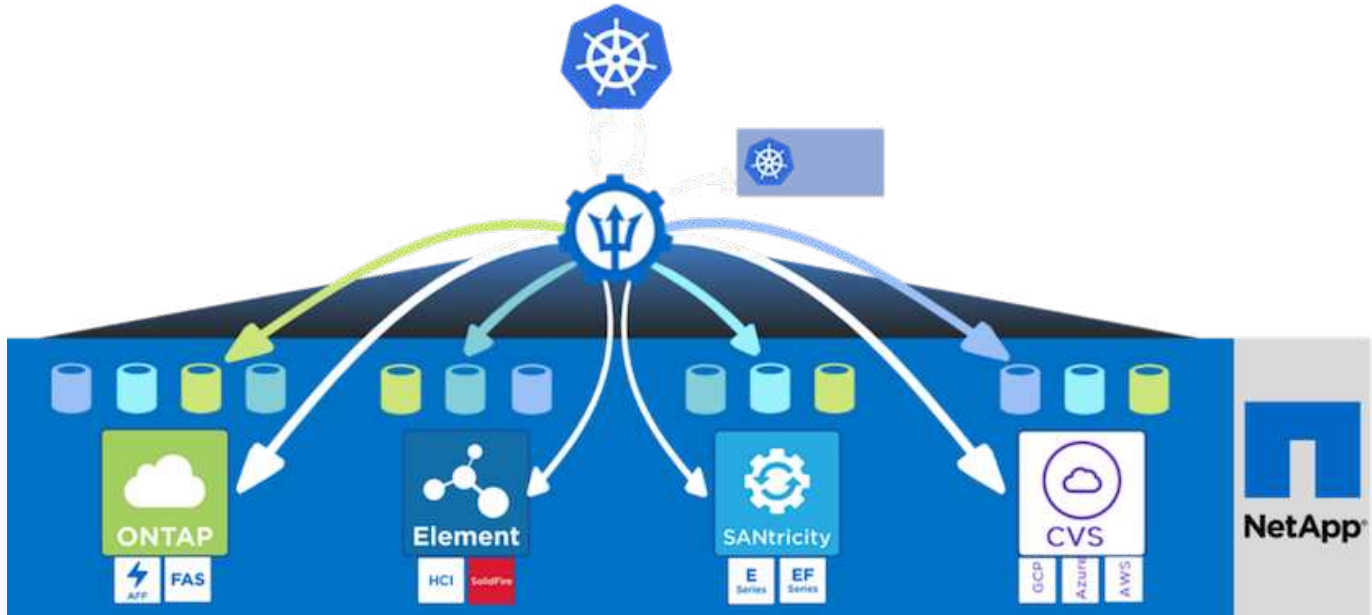
<input type="checkbox"/>	Name	State	Protection	Cluster	Group	Discovered ↓	Actions
<input type="checkbox"/>	magento-bef7f	✓ Healthy	⚠ Unprotected	tkgi-acc	■ magento-bef7f	2022/05/16 21:31 UTC	⋮
<input type="checkbox"/>	magento	✓ Healthy	ℹ Partially protected	tkgi-acc	■ magento	2022/05/09 18:20 UTC	⋮

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名前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名前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名前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、作成した名前スペースに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、作成された名前スペースに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-sqhcg	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> vservers nfs modify -vservers 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-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jumpshot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。