



NetApp 스토리지 통합 개요

NetApp Solutions

NetApp
September 10, 2024

This PDF was generated from https://docs.netapp.com/ko-kr/netapp-solutions/containers/vtwn_astra_register.html on September 10, 2024. Always check docs.netapp.com for the latest.

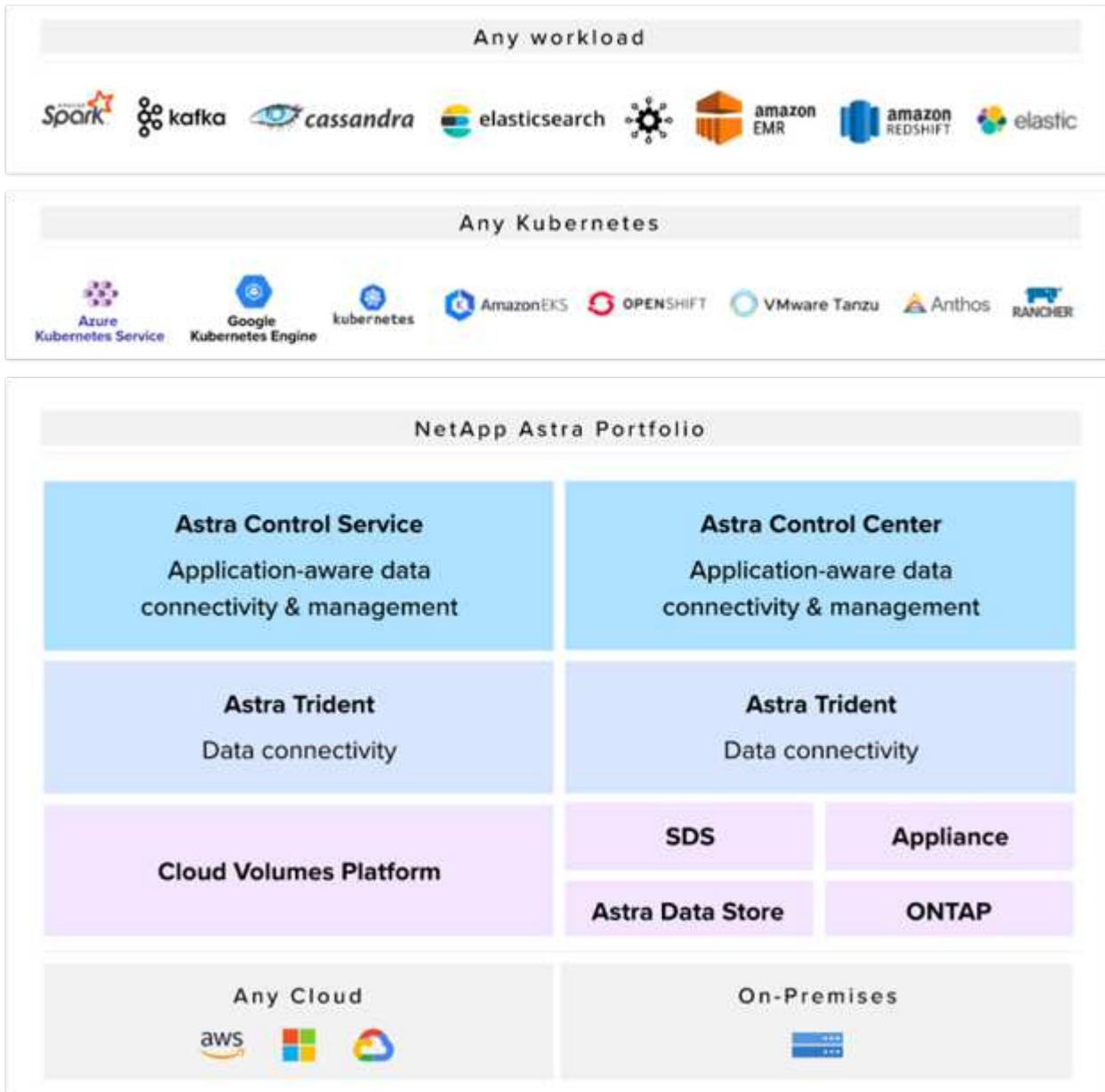
목차

- NetApp 스토리지 통합 개요 1
 - NetApp Astra Control 개요 2
 - Astra Trident 개요 20

NetApp 스토리지 통합 개요

NetApp provides a number of products which assist our customers with orchestrating and managing persistent data in container based environments.

NetApp은 상태 저장 컨테이너 애플리케이션 및 데이터를 오케스트레이션, 관리, 보호 및 마이그레이션하는 데 유용한 다양한 제품을 제공합니다.



NetApp Astra Control은 NetApp 데이터 보호 기술을 기반으로 상태 저장 Kubernetes 워크로드를 위한 풍부한 스토리지 및 애플리케이션 인식 데이터 관리 서비스 세트를 제공합니다. Astra Control Service는 클라우드 네이티브 Kubernetes 구축에서 상태 저장 워크로드를 지원할 수 있습니다. Astra Control Center는 Red Hat OpenShift, Rancher, VMware Tanzu etc과 같은 엔터프라이즈 Kubernetes 플랫폼의 온프레미스 구축에서 상태 저장 워크로드를 지원할 수 있습니다. 자세한 내용은 NetApp Astra Control 웹 사이트를 참조하십시오 ["여기"](#).

NetApp Astra Trident는 컨테이너 및 Kubernetes 배포용 Red Hat OpenShift, Rancher, VMware Tanzu etc의 오픈 소스 및 완전 지원 스토리지 오케스트레이터입니다. 자세한 내용은 Astra Trident 웹 사이트를 참조하십시오 ["여기"](#).

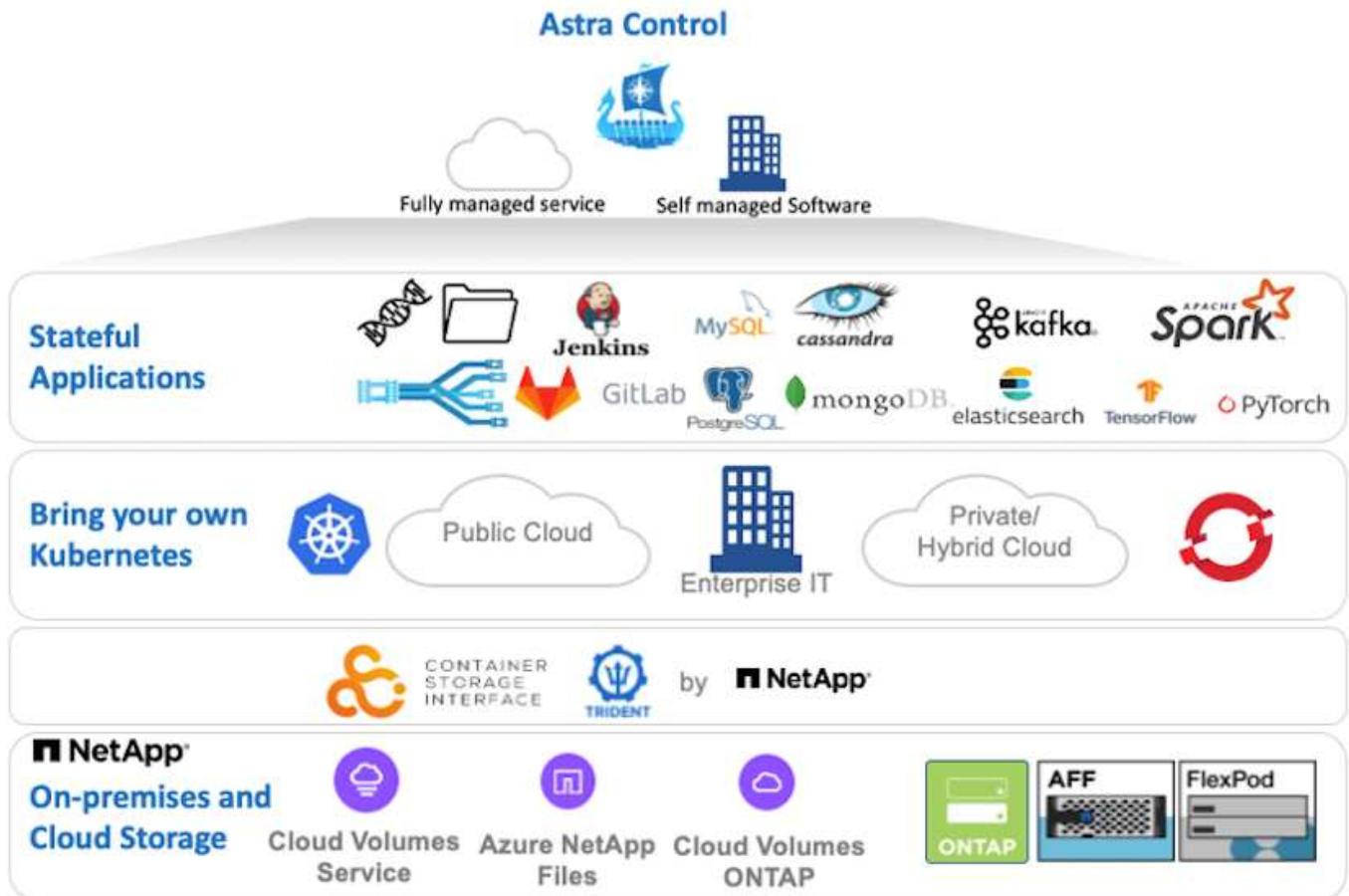
다음 페이지에서는 VMware Tanzu with NetApp 솔루션의 애플리케이션 및 영구 스토리지 관리에 대해 검증된 NetApp 제품에 대한 추가 정보를 제공합니다.

- ["NetApp Astra Control Center를 참조하십시오"](#)
- ["NetApp Astra Trident"](#)

NetApp Astra Control 개요

NetApp Astra Control Center는 NetApp의 신뢰할 수 있는 데이터 보호 기술을 기반으로 사내 환경에 구축된 상태 저장 Kubernetes 워크로드를 위한 풍부한 스토리지 및 애플리케이션 인식 데이터 관리 서비스 세트를 제공합니다.

NetApp Astra Control Center는 사내 환경에 구축되어 NetApp 데이터 보호 기술을 기반으로 하는 상태 저장 Kubernetes 워크로드를 위한 풍부한 스토리지 및 애플리케이션 인식 데이터 관리 서비스 세트를 제공합니다.



NetApp Astra Control Center는 NetApp ONTAP 스토리지 시스템에 스토리지 클래스 및 스토리지 백엔드를 사용하여 구축 및 구성된 Astra Trident 스토리지 오케스트레이터가 있는 VMware Tanzu 클러스터에 설치할 수 있습니다.

Astra Trident에 대한 자세한 내용은 [이 문서는 여기 에서 확인할 수 있습니다](#).

클라우드 연결 환경에서 Astra Control Center는 Cloud Insights를 사용하여 고급 모니터링 및 원격 측정 기능을

제공합니다. Cloud Insights 연결이 없을 경우 제한된 모니터링 및 원격 측정(7일 메트릭)을 사용할 수 있으며 개방형 메트릭 엔드포인트를 통해 Kubernetes 기본 모니터링 툴(Prometheus 및 Grafana)으로 내보낼 수 있습니다.

Astra Control Center는 NetApp AutoSupport 및 Active IQ Digital Advisor(Digital Advisor라고도 함) 에코시스템에 완벽하게 통합되어 사용자를 지원하고, 문제 해결을 지원하며, 사용 통계를 표시합니다.

Astra Control Center의 유료 버전 외에 90일 평가판 라이선스도 제공됩니다. 평가판 버전은 이메일과 Community Slack 채널을 통해 지원됩니다. 고객은 제품 내 지원 대시보드에서 이러한 리소스, 기타 기술 자료 문서 및 문서를 사용할 수 있습니다.

Astra 포트폴리오에 대한 자세한 내용은 ["Astra 웹 사이트"](#)를 참조하십시오.

Astra Control Center 자동화

Astra Control Center에는 프로그래밍 방식의 액세스를 위한 완전한 기능의 REST API가 있습니다. 사용자는 프로그래밍 언어 또는 유틸리티를 사용하여 Astra Control REST API 끝점과 상호 작용할 수 있습니다. 이 API에 대한 자세한 내용은 설명서를 참조하십시오 ["여기"](#).

Astra Control REST API와 상호 작용하기 위해 미리 만들어진 소프트웨어 개발 툴킷을 찾고 있다면 NetApp에서 다운로드할 수 있는 Astra Control Python SDK와 함께 툴킷을 제공합니다 ["여기"](#).

프로그래밍이 현재 상황에 맞지 않고 구성 관리 툴을 사용하려는 경우 NetApp에서 게시하는 Ansible 플레이북을 클론 복제 및 실행할 수 있습니다 ["여기"](#).

Astra Control Center 설치 필수 구성 요소

Astra Control Center를 설치하려면 다음과 같은 필수 구성 요소가 필요합니다.

- 관리 클러스터 또는 TKGS 또는 TKGI에서 관리하는 하나 이상의 Tanzu Kubernetes 클러스터 TKG 워크로드 클러스터 1.4+ 및 TKGI 사용자 클러스터 1.12.2+가 지원됩니다.
- Astra Trident는 각 Tanzu Kubernetes 클러스터에 이미 설치 및 구성되어 있어야 합니다.
- ONTAP 9.5 이상을 실행 중인 NetApp ONTAP 스토리지 시스템 하나 이상



사이트에 각 Tanzu Kubernetes를 설치하여 영구 스토리지용 전용 SVM을 사용하는 것이 모범 사례입니다. 다중 사이트 배포에는 추가 스토리지 시스템이 필요합니다.

- Trident 스토리지 백엔드는 ONTAP 클러스터에서 지원하는 SVM을 사용하여 각 Tanzu Kubernetes 클러스터에서 구성해야 합니다.
- 스토리지 프로비저닝자로 Astra Trident가 있는 각 Tanzu Kubernetes 클러스터에 구성된 기본 StorageClass입니다.
- 로드 밸런싱을 위해 각 Tanzu Kubernetes 클러스터에 로드 밸런싱 장치를 설치하고 구성해야 하며, 이때 ingressType 'AccTraefik'를 사용하는 경우 Astra Control Center가 표시됩니다.
- ingressType "Generic"을 사용하는 경우 Astra Control Center를 노출하기 위해 각 Tanzu Kubernetes 클러스터에 수신 컨트롤러를 설치하고 구성해야 합니다.
- NetApp Astra Control Center 이미지를 호스팅하도록 프라이빗 이미지 레지스트리를 구성해야 합니다.
- Astra Control Center가 설치되어 있는 Tanzu Kubernetes 클러스터에 대한 클러스터 관리자 액세스 권한이 있어야 합니다.

- NetApp ONTAP 클러스터에 대한 관리 액세스 권한이 있어야 합니다.
- RHEL 또는 Ubuntu 관리 워크스테이션.

Astra Control Center를 설치합니다

이 솔루션은 Ansible 플레이북을 사용하여 Astra Control Center를 설치하는 자동화된 절차를 설명합니다. Astra Control Center를 설치하기 위한 수동 절차를 찾고 있는 경우 자세한 설치 및 작동 설명서를 따르십시오 ["여기"](#).

1. Astra Control Center를 배포하는 Ansible 플레이북을 사용하려면 Ansible이 설치된 Ubuntu/RHEL 시스템이 있어야 합니다. 절차를 따르십시오 ["여기"](#) Ubuntu 및 RHEL의 경우
2. Ansible 콘텐츠를 호스팅하는 GitHub 저장소의 클론을 생성합니다.

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

3. NetApp Support 사이트에 로그인하여 NetApp Astra Control Center의 최신 버전을 다운로드하십시오. 그렇게 하려면 NetApp 계정에 연결된 라이선스가 필요합니다. 타볼을 다운로드한 후 워크스테이션으로 전송합니다.



Astra Control 평가판 라이선스를 시작하려면 를 방문하십시오 ["Astra 등록 사이트입니다"](#).

4. Astra Control Center가 설치될 사용자 또는 워크로드 Tanzu Kubernetes 클러스터에 대한 관리자 액세스 권한이 있는 kubeconfig 파일을 만들거나 얻습니다.
5. 디렉터리를 na_astra_control_suite로 변경합니다.

```
cd na_astra_control_suite
```

6. VAR/VAR.yml 파일을 편집하여 필요한 정보로 변수를 입력합니다.

```
#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"

#KubereneTS/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. Playbook을 실행하여 Astra Control Center를 구축합니다. 특정 구성에 대한 루트 권한이 Playbook에 필요합니다.

Playbook을 실행하는 사용자가 root 이거나 암호 없는 sudo가 구성된 경우 다음 명령을 실행하여 플레이북을 실행합니다.

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

사용자에게 암호 기반 sudo 액세스가 구성된 경우 다음 명령을 실행하여 플레이북을 실행한 다음 sudo 암호를 입력합니다.

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

설치 후 단계

1. 설치가 완료되는 데 몇 분 정도 걸릴 수 있습니다. NetApp-Astra-cc 네임스페이스의 모든 Pod와 서비스가 실행 중인지 확인합니다.

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

2. 설치가 완료되었는지 확인하려면 'acc-operator-controller-manager' 로그를 확인하십시오.

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



다음 메시지는 Astra Control Center가 성공적으로 설치되었음을 나타냅니다.

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

3. Astra Control Center에 로그인하기 위한 사용자 이름은 CRD 파일에 제공된 관리자의 이메일 주소이며 암호는 Astra Control Center UUID에 추가된 문자열 ACC- 입니다. 다음 명령을 실행합니다.


```
[netapp-user@rhel7 ~]$ oc get astracontrolcenters -n netapp-astra-cc
```

NAME	UUID
astra	345c55a5-bf2e-21f0-84b8-b6f2bce5e95f



이 예에서 암호는 'ACC-345c55a5-bf2e-21f0-84b8-b6f2bce5e95f'입니다.

4. ingressType이 AccTraefik인 경우 traefik 서비스 로드 밸런싱 장치 IP를 가져옵니다.

```
[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 파일에서 제공하는 FQDN을 가리키는 DNS 서버의 entry를 traefik 서비스의 'external-ip'에 추가한다.

New Host

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

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

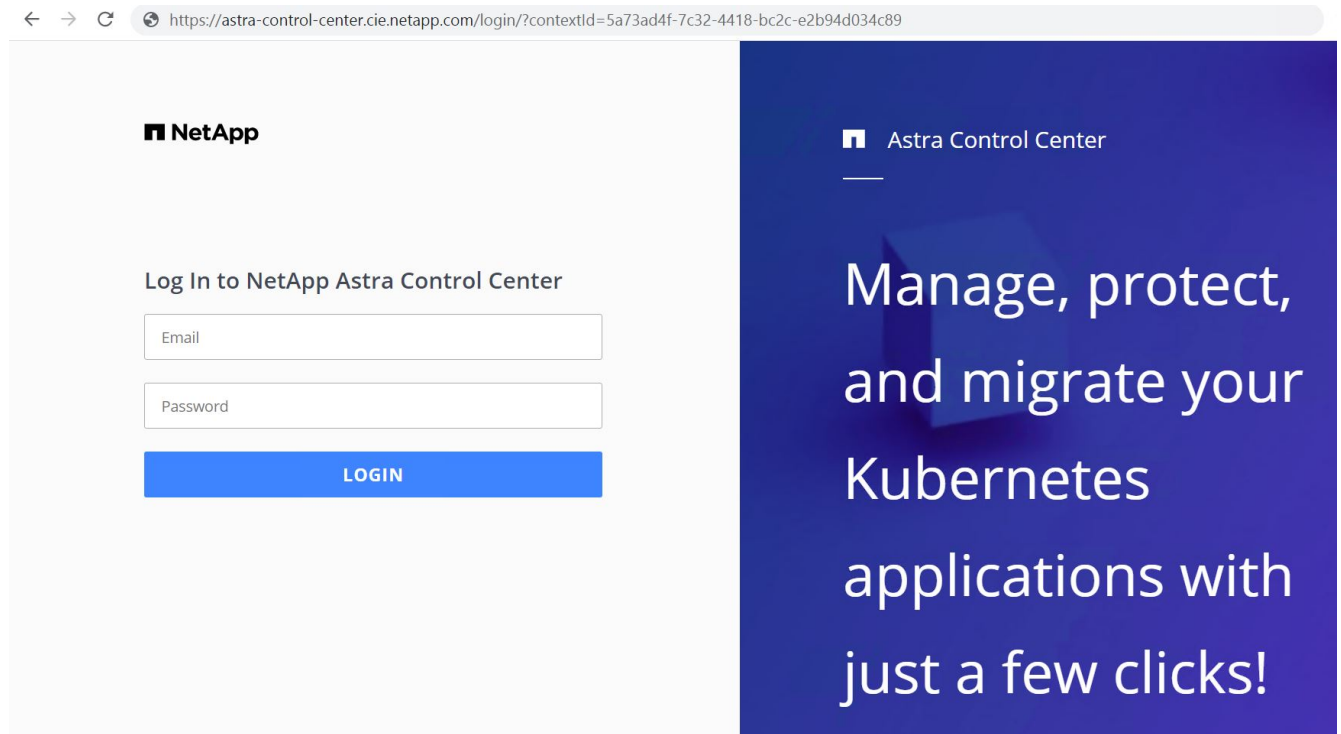
IP address:
10.61.186.181

☒ Create associated pointer (PTR) record

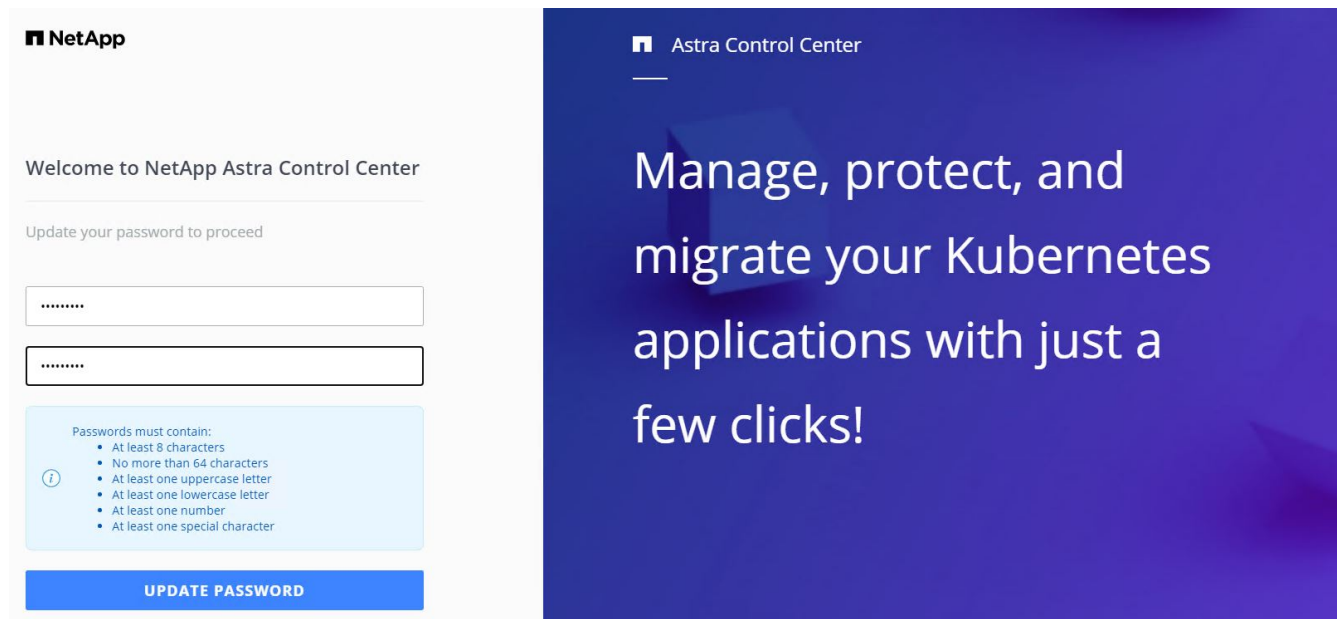
☐ Allow any authenticated user to update DNS records with the same owner name

Add Host Cancel

6. FQDN을 검색하여 Astra Control Center GUI에 로그인합니다.



7. CRD에 제공된 관리자 이메일 주소를 사용하여 처음으로 Astra Control Center GUI에 로그인할 경우 비밀번호를 변경해야 합니다.



8. Astra Control Center에 사용자를 추가하려면 계정 > 사용자 로 이동하여 추가 를 클릭하고 사용자 세부 정보를 입력한 다음 추가 를 클릭합니다.

Add user

USER DETAILS

First name: Nikhil

Last name: Kulkarni

Email address: tme_nik@netapp.com

PASSWORD

Temporary password: *****

Confirm temporary password: *****

Passwords must contain:

- At least 8 characters
- No more than 64 characters
- At least one lowercase letter
- At least one uppercase letter
- At least one number
- At least one special character

USER ROLE

Role: Owner

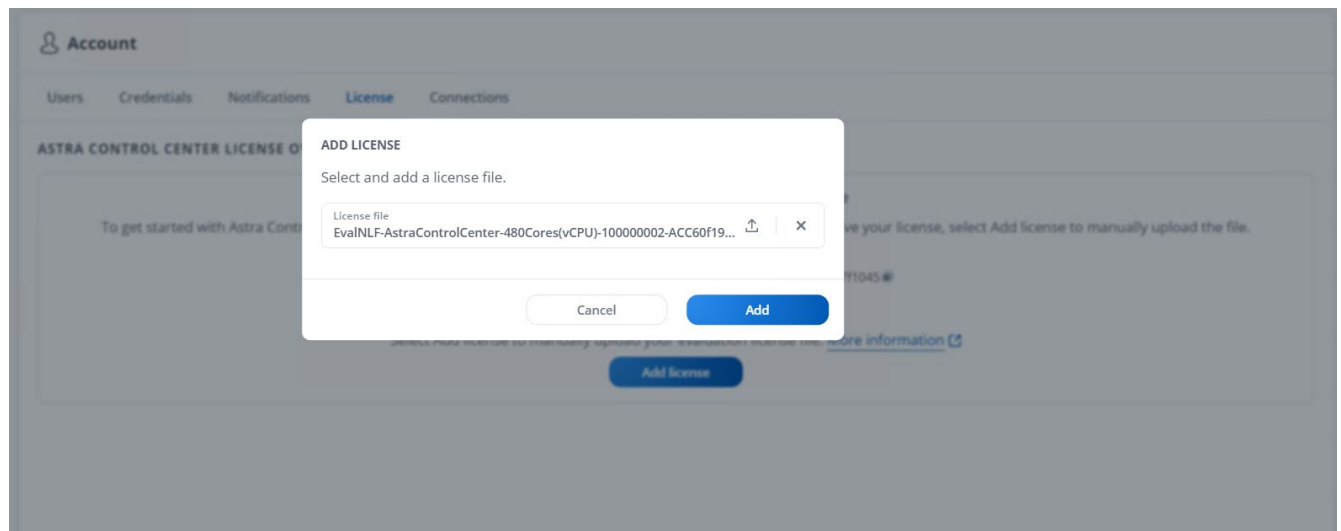
Buttons: Cancel, Add ✓

ADD NEW USER

Add new user

Add a new user to your Astra Control Center account. New users will be prompted to update their password the first time they log in to Astra Control Center. They will also inherit access to account-wide credentials according to their role. Read more in [users](#).

9. Astra Control Center를 사용하려면 모든 기능을 사용할 수 있는 라이선스가 필요합니다. 라이선스를 추가하려면 계정 > 라이선스 로 이동하고 라이선스 추가 를 클릭한 다음 라이선스 파일을 업로드합니다.




NetApp Astra Control Center의 설치 또는 구성 관련 문제가 발생할 경우 알려진 문제에 대한 기술 자료를 이용할 수 있습니다 ["여기"](#).

Astra Control Center에 VMware Tanzu Kubernetes 클러스터를 등록합니다

Astra Control Center에서 워크로드를 관리할 수 있도록 하려면 먼저 Tanzu Kubernetes 클러스터를 등록해야 합니다.

VMware Tanzu Kubernetes 클러스터 등록

1. 첫 번째 단계는 Astra Control Center에 Tanzu Kubernetes 클러스터를 추가하고 관리하는 것입니다. 클러스터로 이동하고 클러스터 추가를 클릭하고 Tanzu Kubernetes 클러스터에 대한 kubeconfig 파일을 업로드한 다음 스토리지 선택을 클릭합니다.

 **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

 **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](#).

Cancel

Next →

2. Astra Control Center가 적합한 스토리지 클래스를 감지합니다. 이제 NetApp ONTAP에서 SVM이 지원하는 Trident를 사용하여 스토리지 클래스의 볼륨 프로비저닝 방법을 선택하고 검토를 클릭합니다. 다음 창에서 세부 정보를 확인하고 Add Cluster를 클릭합니다.
3. 클러스터를 추가하면 검색 상태로 이동하고 Astra Control Center에서 클러스터를 검사하고 필요한 에이전트를 설치합니다. 정상적으로 등록되면 클러스터 상태가 정상 상태로 바뀝니다.

 **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 Center에서 관리할 모든 Tanzu Kubernetes 클러스터는 관리 클러스터에 설치된 에이전트가 해당 레지스트리에서 이미지를 가져올 때 설치에 사용된 이미지 레지스트리에 액세스할 수 있어야 합니다.

4. Astra Control Center에서 백엔드를 관리할 스토리지 리소스로 ONTAP 클러스터를 가져옵니다. Tanzu Kubernetes 클러스터를 Astra에 추가하고 스토리지 클래스를 구성하면 스토리지 클래스를 지원하는 ONTAP 클러스터를 자동으로 검색하고 검사하지만 관리를 위해 Astra Control Center로 가져오지 않습니다.

Backends

+ Add

Search

★

🔍

1

1-1 of 1 entries

<

>

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

5. ONTAP 클러스터를 가져오려면 백엔드에서 드롭다운을 클릭하고 관리할 ONTAP 클러스터 옆의 관리를 선택합니다. ONTAP 클러스터 자격 증명을 입력하고 정보 검토 를 클릭한 다음 스토리지 백엔드 가져오기 를 클릭합니다.

Manage ONTAP storage backend

STEP 1/2: CREDENTIALS

✕

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. 백엔드가 추가되면 상태가 사용 가능으로 변경됩니다. 이러한 백엔드는 이제 Tanzu Kubernetes 클러스터의 영구 볼륨과 ONTAP 시스템의 해당 볼륨에 대한 정보를 갖게 됩니다.


Backends

7. Astra Control Center를 사용하여 Tanzu Kubernetes 클러스터 전체에서 백업 및 복구를 수행하려면 S3 프로토콜을 지원하는 오브젝트 스토리지 버킷을 프로비저닝해야 합니다. 현재 지원되는 옵션은 ONTAP S3, StorageGRID, AWS S3, Microsoft Azure Blob 스토리지입니다. 이 설치를 위해 AWS S3 버킷을 구성하려고 합니다. Bucket 으로 이동하여 Bucket 추가 를 클릭하고 Generic S3 를 선택합니다. S3 버킷과 자격 증명에 액세스하기 위한 세부 정보를 입력하고 이 Bucket을 클라우드의 기본 버킷으로 설정 확인란을 클릭한 다음 추가를 클릭합니다.

 **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

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](#)

Cancel

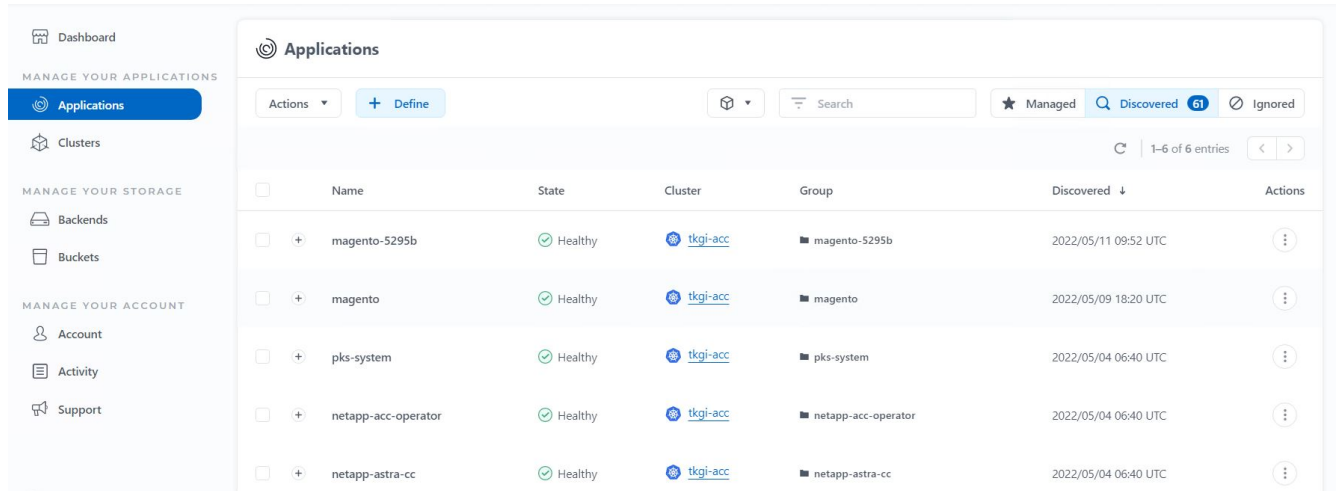
Add 

보호할 애플리케이션을 선택하십시오

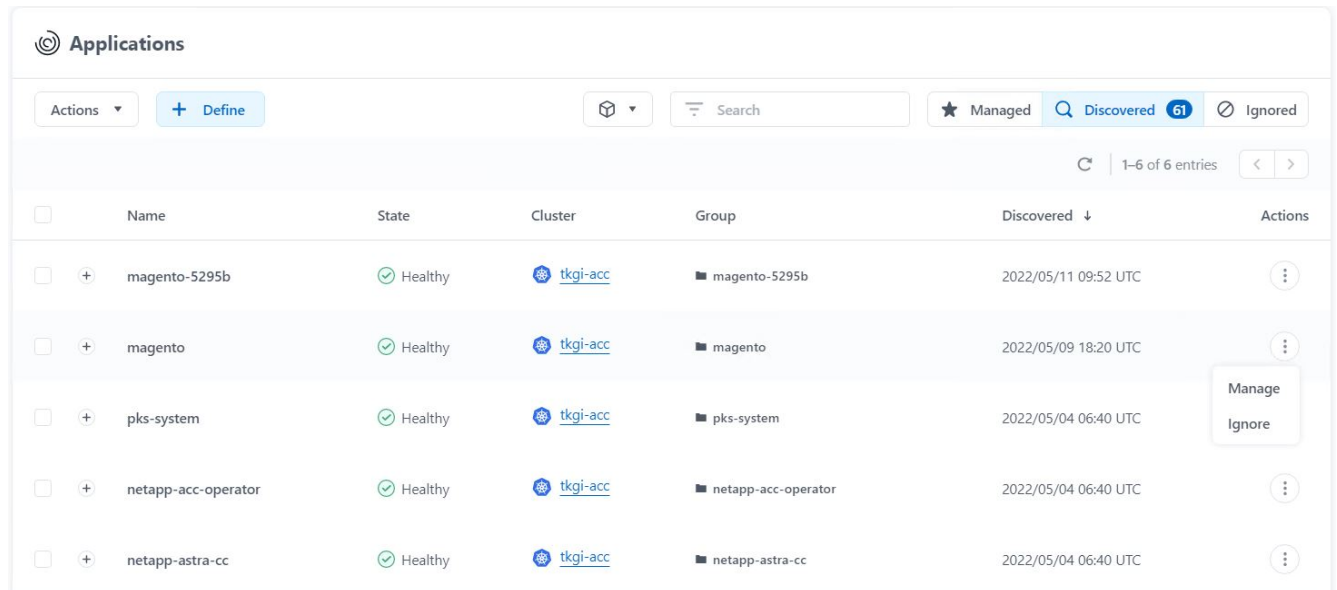
Tanzu Kubernetes 클러스터를 등록한 후에는 Astra Control Center를 통해 구축된 애플리케이션을 검색하고 관리할 수 있습니다.

애플리케이션 관리

1. Tanzu Kubernetes 클러스터 및 ONTAP 백엔드가 Astra Control Center에 등록되면, 제어 센터는 지정된 ONTAP 백엔드로 구성된 스토리지 시스템을 사용하는 모든 네임스페이스에서 애플리케이션을 자동으로 검색하기 시작합니다.



2. 앱 > 검색됨 으로 이동한 후 Astra를 사용하여 관리하려는 애플리케이션 옆에 있는 드롭다운 메뉴를 클릭합니다. 관리 를 클릭합니다.



3. 응용 프로그램이 사용 가능 상태로 전환되고 앱 섹션의 관리 탭에서 볼 수 있습니다.

Applications

Actions ▾

+ Define

All clusters ▾

≡

Search

★ Managed

🔍

Discovered60

🚫

Ignored

↺

1-1 of 1 entries

⏪

⏩

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

애플리케이션 보호

Astra Control Center에서 애플리케이션 워크로드를 관리하고 나면 해당 워크로드에 대한 보호 설정을 구성할 수 있습니다.

애플리케이션 스냅샷을 생성합니다

애플리케이션의 스냅샷은 ONTAP 스냅샷 복사본과 애플리케이션 메타데이터의 복사본을 생성합니다. 이 복사본은 스냅샷 복사본을 기준으로 애플리케이션을 특정 시점으로 복원 또는 클론 복제하는 데 사용할 수 있습니다.

- 응용 프로그램의 스냅샷을 만들려면 앱 > 관리 탭으로 이동하여 스냅샷 복사본을 만들 응용 프로그램을 클릭합니다. 응용 프로그램 이름 옆의 드롭다운 메뉴를 클릭하고 스냅샷 을 클릭합니다.

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

- 스냅샷 세부 정보를 입력하고 다음 을 클릭한 다음 스냅샷 을 클릭합니다. 스냅샷을 생성하는 데 약 1분이 소요되며 스냅샷이 성공적으로 생성된 후 상태를 사용할 수 있습니다.

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 →

애플리케이션 백업을 생성합니다

애플리케이션 백업에서는 애플리케이션의 활성 상태와 애플리케이션 리소스의 구성을 캡처하여 파일로 저장한 다음 원격 오브젝트 스토리지 버킷에 저장합니다.

1. Astra Control Center에서 관리 대상 애플리케이션을 백업 및 복구하려면 백업 ONTAP 시스템에 대한 고급 사용자 설정을 사전 요구 사항으로 구성해야 합니다. 이렇게 하려면 다음 명령을 입력합니다.

```
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 Center에서 관리 대상 응용 프로그램의 백업을 생성하려면 Apps > Managed 탭으로 이동하여 백업할 응용 프로그램을 클릭합니다. 응용 프로그램 이름 옆의 드롭다운 메뉴를 클릭하고 백업을 클릭합니다.

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

3. 백업 세부 정보를 입력하고 백업 파일을 보관할 객체 스토리지 버킷을 선택한 후 다음 을 클릭하고 세부 정보를 검토한 후 백업 을 클릭합니다. 애플리케이션 및 데이터의 크기에 따라 백업이 몇 분 정도 걸릴 수 있으며 백업이 성공적으로 완료된 후 백업 상태를 사용할 수 있게 됩니다.

 **Back up namespace application**

STEP 1/2: DETAILS

X

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 →

응용 프로그램을 복원하는 중입니다.

버튼을 한 번만 누르면 애플리케이션을 동일한 클러스터의 원래 네임스페이스 또는 애플리케이션 보호 및 재해 복구를 위해 원격 클러스터로 복원할 수 있습니다.

1. 응용 프로그램을 복원하려면 앱 > 관리 탭으로 이동하여 해당 앱을 클릭합니다. 응용 프로그램 이름 옆의 드롭다운 메뉴를 클릭하고 복원을 클릭합니다.


magento



APPLICATION STATUS

Healthy

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

tkc

Actions

- Snapshot
- Backup
- Clone
- Restore
- Unmanage

2. 복원 네임스페이스의 이름을 입력하고 복원할 클러스터를 선택한 다음 기존 스냅샷이나 응용 프로그램 백업에서 복원할지 여부를 선택합니다. 다음을 클릭합니다.

Restore namespace application

STEP 1/2: DETAILS

X

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 →

3. 검토 창에서 Restore를 입력하고 세부 정보를 검토한 후 Restore를 클릭합니다.

Restore namespace application

STEP 2/2: SUMMARY

X

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 ✓

4. 새 애플리케이션은 Restoring 상태로, Astra Control Center는 선택한 클러스터의 애플리케이션을 복구합니다. 응용 프로그램의 모든 리소스가 Astra에 의해 설치 및 감지되면 응용 프로그램은 사용 가능 상태로 전환됩니다.

18

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>✔ Healthy</div>	<div>⚠ Unprotected</div>	<div>⚙️ tkgi-acc</div>	<div>📁 magento</div>	2022/05/09 18:20 UTC	<div>⋮</div>

애플리케이션 클론 생성

개발/테스트 또는 애플리케이션 보호 및 재해 복구를 위해 애플리케이션을 원래 클러스터 또는 원격 클러스터에 복제할 수 있습니다. 동일한 스토리지 백엔드에서 동일한 클러스터 내에 애플리케이션을 클론 복제하면 NetApp FlexClone 기술이 사용되므로 PVC를 즉시 클로닝하고 스토리지 공간을 절약할 수 있습니다.

1. 응용 프로그램을 복제하려면 앱 > 관리 탭으로 이동하고 해당 앱을 클릭합니다. 애플리케이션 이름 옆의 드롭다운 메뉴를 클릭하고 클론 을 클릭합니다.

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. 새 네임스페이스의 세부 정보를 입력하고 복제할 클러스터를 선택한 다음 기존 스냅샷, 백업 또는 애플리케이션의 현재 상태에서 클론을 생성할지 여부를 선택합니다. 세부 정보를 검토한 후 다음 을 클릭하고 검토 창에서 복제 를 클릭합니다.

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. 새 애플리케이션은 검색 상태로 전환되지만, Astra Control Center는 선택한 클러스터에 애플리케이션을 생성합니다. 응용 프로그램의 모든 리소스가 Astra에 의해 설치 및 감지되면 응용 프로그램은 사용 가능 상태로 전환됩니다.

Applications

Actions ▾
+ Define

All clusters ▾
Search

★ Managed

🔍 Discovered 60

🚫 Ignored

1-2 of 2 entries

< >

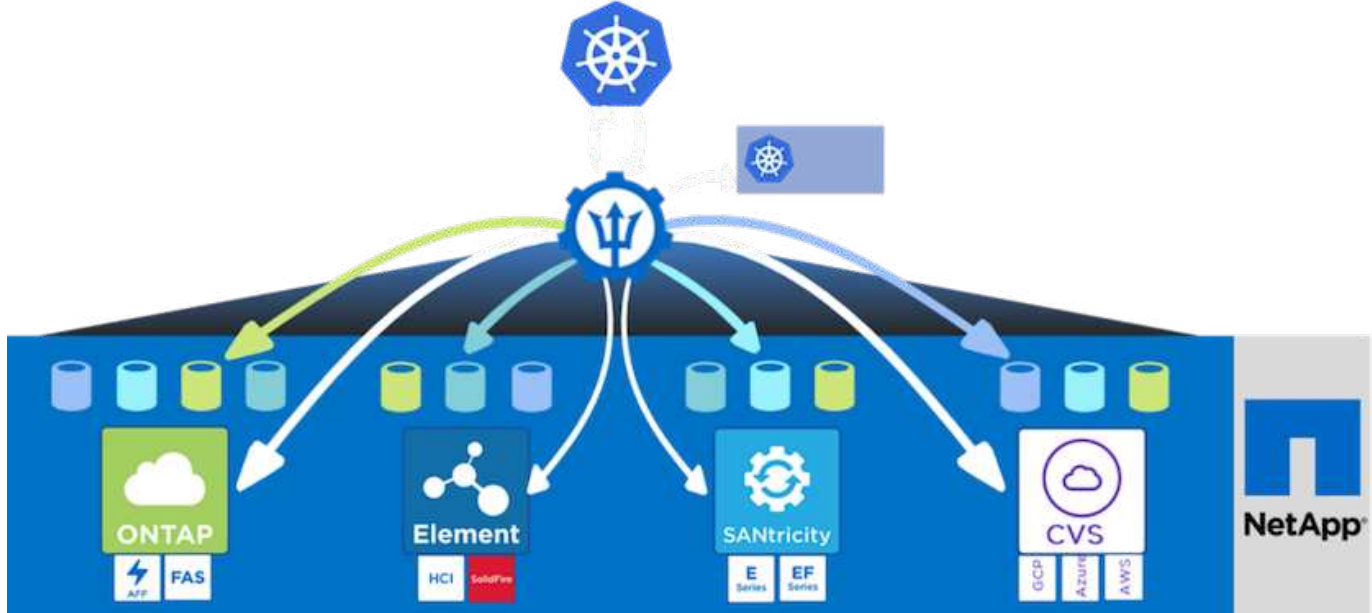
<input type="checkbox"/>	Name	State	Protection	Cluster	Group	Discovered ↓	Actions
<input type="checkbox"/>	magento-bef7f	🟢 Healthy	⚠️ Unprotected	tkgi-acc	magento-bef7f	2022/05/16 21:31 UTC	⋮
<input type="checkbox"/>	magento	🟢 Healthy	ℹ️ Partially protected	tkgi-acc	magento	2022/05/09 18:20 UTC	⋮

Astra Trident 개요

Astra Trident는 VMware Tanzu를 비롯한 컨테이너 및 Kubernetes 배포를 위한 오픈 소스 및 완벽하게 지원되는 스토리지 오케스트레이터입니다.

Astra Trident는 컨테이너 및 Kubernetes 배포용 Red Hat OpenShift, VMware Tanzu, Anthos by Google Cloud, Rancher etc의 완전 지원되는 오픈 소스 스토리지 오케스트레이터입니다. Trident는 NetApp ONTAP 및 Element 스토리지 시스템을 포함한 전체 NetApp 스토리지 포트폴리오와 연동되며 NFS 및 iSCSI 연결도 지원합니다. Trident는 최종 사용자가 스토리지 관리자의 개입 없이 NetApp 스토리지 시스템에서 스토리지를 프로비저닝 및 관리할 수 있도록 하여 DevOps 워크플로우를 가속합니다.

관리자는 특정 수준의 성능을 보장하는 압축, 특정 디스크 유형 또는 QoS 수준을 비롯한 고급 스토리지 기능을 지원하는 스토리지 시스템 모델과 프로젝트 요구사항에 따라 여러 스토리지 백엔드를 구성할 수 있습니다. 이러한 백엔드를 정의한 후, 개발자는 프로젝트의 이러한 백엔드를 사용하여 지속적인 PVC(Volume Claim)를 생성하고 필요에 따라 컨테이너에 영구 저장소를 연결할 수 있습니다.



Astra Trident는 빠른 개발 주기를 제공하며 Kubernetes와 마찬가지로 1년에 4회 릴리즈됩니다.

Astra Trident의 최신 버전은 2022년 4월 22.04입니다. Kubernetes 배포를 찾을 수 있는 Trident의 버전에 대한 지원 매트릭스입니다 ["여기"](#).

20.04 릴리즈부터 Trident 운영자가 Trident 설정을 수행합니다. 운영자는 대규모 구축을 용이하게 하고 Trident 설치의 일부로 배포된 Pod의 자동 복구를 포함하여 추가 지원을 제공합니다.

21.01 릴리즈를 통해 Trident Operator의 설치를 용이하게 하는 제어 차트를 사용할 수 있게 되었습니다.

Helm을 사용하여 Trident 연산자를 배포합니다

1. 먼저 사용자 클러스터의 "kubeconfig" 파일 위치를 환경 변수로 설정하여 Trident에 이 파일을 전달할 수 있는 옵션이 없으므로 참조할 필요가 없습니다.

```
[netapp-user@rhel7]$ export KUBECONFIG=~/.kubeconfig
```

2. NetApp Astra Trident Helm 저장소를 추가합니다.

```
[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 리포지토리를 업데이트합니다.

```
[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. Trident를 설치할 새 네임스페이스를 만듭니다.

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

5. DockerHub 자격 증명으로 암호를 만들어 Astra Trident 이미지를 다운로드합니다.

```
[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. 관리 클러스터 구축이 포함된 TKGS(Tanzu가 설치된 vSphere) 또는 TKG에서 관리하는 사용자 또는 워크로드 클러스터의 경우 다음 절차에 따라 Astra Trident를 설치합니다.

- a. 로그인한 사용자에게 삼중임 네임스페이스에서 서비스 계정을 만들 수 있는 권한이 있는지, 삼중임 네임스페이스의 서비스 계정에 POD를 만들 수 있는 권한이 있는지 확인합니다.
- b. 생성된 네임스페이스에 Trident 연산자를 설치하려면 아래 helm 명령을 실행합니다.

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

7. TKG1 배포가 관리하는 사용자 또는 워크로드 클러스터의 경우 다음 helm 명령을 실행하여 생성된 네임스페이스에 Trident 연산자를 설치합니다.

```
[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 Pod가 가동되어 실행 중인지 확인합니다.

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        |
+-----+
```

스토리지 시스템 백엔드를 생성합니다

Astra Trident Operator 설치를 완료한 후에는 사용 중인 특정 NetApp 스토리지 플랫폼에 대한 백엔드를 구성해야 합니다. Astra Trident의 설정 및 구성을 계속하려면 아래 링크를 따라가십시오.

- ["NetApp ONTAP NFS 를 참조하십시오"](#)
- ["NetApp ONTAP iSCSI를 참조하십시오"](#)

NetApp ONTAP NFS 구성

NFS를 통해 NetApp ONTAP 스토리지 시스템과의 Trident 통합을 활성화하려면 스토리지 시스템과의 통신을 지원하는 백엔드를 생성해야 합니다. 이 솔루션에 기본 백엔드를 구성하지만, 보다 맞춤형된 옵션을 원할 경우 설명서를 참조하십시오 ["여기"](#).

ONTAP에서 SVM을 생성합니다

1. ONTAP System Manager에 로그인하고 스토리지 > 스토리지 VM으로 이동한 다음 추가를 클릭합니다.
2. SVM의 이름을 입력하고 NFS 프로토콜을 설정한 다음 Allow NFS Client Access 확인란을 선택하고, 워크로드 클러스터에서 볼륨을 PVS로 마운트하도록 허용하는 익스포트 정책 규칙에 작업자 노드가 있는 서브넷을 추가합니다.

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-T를 사용하여 NAT로 구축된 사용자 클러스터 또는 워크로드 클러스터를 사용하는 경우 송신 서브넷(TKGS0의 경우 또는 TKGI의 경우 부동 IP 서브넷)을 익스포트 정책 규칙에 추가해야 합니다.

3. 데이터 LIF와 SVM 관리 계정에 대한 세부 정보를 제공한 다음 저장을 클릭합니다.

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. SVM에 애그리게이트를 할당합니다. Storage > Storage VMs 로 이동하고 새로 생성된 SVM 옆에 있는 줄임표를 클릭한 다음 Edit 를 클릭합니다. Limit Volume Creation to Preferred Local Tiers 확인란을 선택하고 필요한 애그리게이트를 이 체크박스에 연결합니다.

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. Trident가 설치될 사용자 또는 워크로드 클러스터의 NAT가 구축된 경우 SNAT로 인해 비표준 포트에서 스토리지 마운트 요청이 도착할 수 있습니다. 기본적으로 ONTAP에서는 루트 포트에서 시작된 볼륨 마운트 요청만 허용합니다. 따라서 ONTAP CLI에 로그인하고 설정을 수정하여 비표준 포트의 마운트 요청을 허용합니다.

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

백엔드 및 **StorageClasses**를 생성합니다

1. NFS를 지원하는 NetApp ONTAP 시스템의 경우 backendName, managementLIF, dataLIF, svm, 사용자 이름을 사용하여 jumphost에 백엔드 구성 파일을 생성합니다. 암호 및 기타 세부 정보

```
{
  "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"
}
```



사용자 지정 backendName 값을 storageDriverName 과 NFS를 함께 사용하여 쉽게 식별할 수 있도록 하는 데이터 LIF를 함께 정의하는 것이 좋습니다.

2. 다음 명령을 실행하여 Trident 백엔드를 생성합니다.

```
[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. 백엔드가 생성되면 다음 번에 스토리지 클래스를 생성해야 합니다. 다음 샘플 저장소 클래스 정의에서는 필수 및 기본 필드를 강조 표시합니다. 매개 변수 'backendType'은 새로 생성된 Trident 백엔드의 스토리지 드라이버를 반영해야 합니다.

```

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

```

4. kubectl 명령을 실행하여 스토리지 클래스를 생성합니다.

```

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

```

5. 스토리지 클래스를 생성한 후 첫 번째 영구 볼륨 클레임(PVC)을 생성해야 합니다. 다음은 샘플 PVC 정의입니다. 'storageClassName' 필드가 방금 만든 스토리지 클래스의 이름과 일치하는지 확인합니다. 프로비저닝 워크로드에 따라 필요에 따라 PVC 정의를 추가로 사용자 지정할 수 있습니다.

```

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

```

6. kubectl 명령을 실행하여 PVC를 생성한다. 생성 중인 백업 볼륨의 크기에 따라 생성 시간이 다소 걸릴 수 있으므로 완료 시 프로세스를 확인할 수 있습니다.

```

[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 구성

iSCSI를 통해 영구 볼륨의 NetApp ONTAP 스토리지 시스템을 VMware Tanzu Kubernetes 클러스터와 통합하여 각 노드에 로그인하고 iSCSI 볼륨을 마운트하는 iSCSI 유틸리티 또는 패키지를 구성하여 노드를 준비합니다. 이렇게 하려면 여기에 설명된 절차를 따르십시오 ["링크"](#).



NetApp은 VMware Tanzu Kubernetes 클러스터의 NAT 구현 시 이 절차를 권장하지 않습니다.



TKGI는 변경 불가능한 구성 이미지를 실행하는 Tanzu Kubernetes 클러스터의 노드로 Bosh VM을 사용하며, Bosh VM에서 iSCSI 패키지를 수동으로 변경하는 경우 재부팅 후에도 지속되지 않습니다. 따라서 TKGI에서 구축 및 운영하는 Tanzu Kubernetes 클러스터의 영구 스토리지에는 NFS 볼륨을 사용하는 것이 좋습니다.

클러스터 노드가 iSCSI 볼륨에 대해 준비된 후에는 스토리지 시스템과의 통신을 지원하는 백엔드를 생성해야 합니다. 이 솔루션에 기본 백엔드를 구성했지만, 보다 맞춤화된 옵션을 원하는 경우 설명서를 참조하십시오 ["여기"](#).

ONTAP에서 SVM을 생성합니다

ONTAP에서 SVM을 생성하려면 다음 단계를 완료하십시오.

1. ONTAP System Manager에 로그인하고 스토리지 > 스토리지 VM으로 이동한 다음 추가를 클릭합니다.
2. SVM의 이름을 입력하고 iSCSI 프로토콜을 설정한 다음 데이터 LIF에 대한 세부 정보를 제공합니다.

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 관리 계정의 세부 정보를 입력한 다음 저장을 클릭합니다.

Storage VM Administration

☒ Manage administrator account

USER NAME

vsadmin

PASSWORD

.....

CONFIRM PASSWORD

.....

☐ Add a network interface for storage VM management.

Save

Cancel

4. SVM에 애그리게이트를 할당하려면 스토리지 > 스토리지 VM으로 이동하고, 새로 생성한 SVM 옆의 줄임표를 클릭한 다음 편집 을 클릭합니다. Limit Volume Creation to Preferred Local Tiers 확인란을 선택하고 필요한 애그리게이트를 이 체크박스에 연결합니다.

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

백엔드 및 **StorageClasses**를 생성합니다

1. NFS를 지원하는 NetApp ONTAP 시스템의 경우 backendName, managementLIF, datLIF, svm, 사용자 이름을 사용하여 jumphost에 백엔드 구성 파일을 생성합니다. 암호 및 기타 세부 정보

```
{
  "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. 다음 명령을 실행하여 Trident 백엔드를 생성합니다.

```
[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. 백엔드를 생성한 후 다음 번에 스토리지 클래스를 생성해야 합니다. 다음 샘플 저장소 클래스 정의에서는 필수 및 기본 필드를 강조 표시합니다. 매개 변수 'backendType'은 새로 생성된 Trident 백엔드의 스토리지 드라이버를 반영해야 합니다. 또한 이름 필드 값을 기록해 둡니다. 이 값은 나중에 참조해야 합니다.

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



이 파일에 정의된 fschType이라는 선택적 필드가 있습니다. iSCSI 백엔드에서 이 값을 특정 Linux 파일 시스템 유형(XFS, ext4 등)으로 설정하거나 삭제하여 Tanzu Kubernetes 클러스터가 사용할 파일 시스템을 결정할 수 있도록 할 수 있습니다.

4. kubeck 명령을 실행하여 스토리지 클래스를 생성합니다.

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

5. 스토리지 클래스를 생성한 후 첫 번째 영구 볼륨 클레임(PVC)을 생성해야 합니다. 다음은 샘플 PVC 정의입니다. 'storageClassName' 필드가 방금 만든 스토리지 클래스의 이름과 일치하는지 확인합니다. 프로비저닝 워크로드에 따라 필요에 따라 PVC 정의를 추가로 사용자 지정할 수 있습니다.

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

6. kubectl 명령을 실행하여 PVC를 생성한다. 생성 중인 백업 볼륨의 크기에 따라 생성 시간이 다소 걸릴 수 있으므로 완료 시 프로세스를 확인할 수 있습니다.

```
[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
RWO		ontap-iscsi	3s

저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.