



# **NetApp Storage Integrations Overview**

NetApp Solutions

NetApp  
August 05, 2022

This PDF was generated from [https://docs.netapp.com/us-en/netapp-solutions/containers/tanzu\\_with\\_netapp/vtwn\\_astra\\_register.html](https://docs.netapp.com/us-en/netapp-solutions/containers/tanzu_with_netapp/vtwn_astra_register.html) on August 05, 2022. Always check docs.netapp.com for the latest.

# Table of Contents

- NetApp storage integration overview . . . . . 1
  - NetApp Astra Control overview . . . . . 2
  - Astra Trident overview . . . . . 19

# NetApp storage integration overview

NetApp provides a number of products to help you orchestrate, manage, protect, and migrate stateful containerized applications and their data.



NetApp Astra Control offers a rich set of storage and application-aware data management services for stateful Kubernetes workloads powered by NetApp data protection technology. The Astra Control Service is available to support stateful workloads in cloud-native Kubernetes deployments. The Astra Control Center is available to support stateful workloads in on-premises deployments of Enterprise Kubernetes platforms like Red Hat OpenShift, Rancher, VMware Tanzu etc. For more information visit the NetApp Astra Control website [here](#).

NetApp Astra Trident is an open-source and fully-supported storage orchestrator for containers and Kubernetes distributions like Red Hat OpenShift, Rancher, VMware Tanzu etc. For more information, visit the Astra Trident website [here](#).

The following pages have additional information about the NetApp products that have been validated for application and persistent storage management in the VMware Tanzu with NetApp solution:

- [NetApp Astra Control Center](#)
- [NetApp Astra Trident](#)

[Next: NetApp Astra Control overview.](#)

## NetApp Astra Control overview

NetApp Astra Control Center offers a rich set of storage and application-aware data management services for stateful Kubernetes workloads deployed in an on-premises environment and powered by NetApp data protection technology.



NetApp Astra Control Center can be installed on a VMware Tanzu cluster that has the Astra Trident storage orchestrator deployed and configured with storage classes and storage backends to NetApp ONTAP storage systems.

For more information on Astra Trident, see [this document here](#).

In a cloud-connected environment, Astra Control Center uses Cloud Insights to provide advanced monitoring and telemetry. In the absence of a Cloud Insights connection, limited monitoring and telemetry (seven days worth of metrics) is available and exported to Kubernetes native monitoring tools (Prometheus and Grafana) through open metrics endpoints.

Astra Control Center is fully integrated into the NetApp AutoSupport and Active IQ ecosystem to provide support for users, provide assistance with troubleshooting, and display usage statistics.

In addition to the paid version of Astra Control Center, a 90-day evaluation license is also available. The

evaluation version is supported through email and the community Slack channel. Customers have access to these resources, other knowledge-base articles, and documentation available from the in-product support dashboard.

To understand more about the Astra portfolio, visit the [Astra website](#).

## Astra Control Center automation

Astra Control Center has a fully functional REST API for programmatic access. Users can use any programming language or utility to interact with Astra Control REST API endpoints. To learn more about this API, see the documentation [here](#).

If you are looking for a ready-made software development toolkit for interacting with Astra Control REST APIs, NetApp provides a toolkit with the Astra Control Python SDK that you can download [here](#).

If programming is not appropriate for your situation and you would like to use a configuration management tool, you can clone and run the Ansible playbooks that NetApp publishes [here](#).

## Astra Control Center installation prerequisites

Astra Control Center installation requires the following prerequisites:

- One or more Tanzu Kubernetes clusters, managed either by a management cluster or TKGS or TKGI. TKG workload clusters 1.4+ and TKGI user clusters 1.12.2+ are supported.
- Astra Trident must already be installed and configured on each of the Tanzu Kubernetes clusters.
- One or more NetApp ONTAP storage systems running ONTAP 9.5 or greater.



It's a best practice for each Tanzu Kubernetes install at a site to have a dedicated SVM for persistent storage. Multi-site deployments require additional storage systems.

- A Trident storage backend must be configured on each Tanzu Kubernetes cluster with an SVM backed by an ONTAP cluster.
- A default StorageClass configured on each Tanzu Kubernetes cluster with Astra Trident as the storage provisioner.
- A load balancer must be installed and configured on each Tanzu Kubernetes cluster for load balancing and exposing Astra Control Center if you are using ingressType `AccTraefik`.
- An ingress controller must be installed and configured on each Tanzu Kubernetes cluster for exposing Astra Control Center if you are using ingressType `Generic`.
- A private image registry must be configured to host the NetApp Astra Control Center images.
- You must have Cluster Admin access to the Tanzu Kubernetes cluster where Astra Control Center is being installed.
- You must have Admin access to NetApp ONTAP clusters.
- A RHEL or Ubuntu admin workstation.

## Install Astra Control Center

This solution describes an automated procedure for installing Astra Control Center using Ansible playbooks. If you are looking for a manual procedure to install Astra Control Center, follow the detailed installation and operations guide [here](#).

1. To use the Ansible playbooks that deploy Astra Control Center, you must have an Ubuntu/RHEL machine with Ansible installed. Follow this [procedure](#) for Ubuntu and this [procedure](#) for RHEL.
2. Clone the GitHub repository that hosts the Ansible content.

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

3. Log into the NetApp Support Site and download the latest version of NetApp Astra Control Center. To do so requires a license attached to your NetApp account. After you download the tarball, transfer it to the workstation.



To get started with a trial license for Astra Control, visit the [Astra registration site](#).

4. Create or obtain the kubeconfig file with admin access to the user or workload Tanzu Kubernetes cluster on which Astra Control Center is to be installed.
5. Change the directory to na\_astra\_control\_suite.

```
cd na_astra_control_suite
```

6. Edit the vars/vars.yml file and fill the variables with the required information.

```
#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. Run the playbook to deploy Astra Control Center. The playbook requires root privileges for certain

configurations.

Run the following command to run the playbook if the user running the playbook is root or has passwordless sudo configured.

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

If the user has password-based sudo access configured, then run the following command to run the playbook and then enter the sudo password.

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

## Post Install Steps

1. It might take several minutes for the installation to complete. Verify that all the pods and services in the `netapp-astra-cc` namespace are up and running.

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

2. Check the `acc-operator-controller-manager` logs to ensure that the installation is completed.

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



The following message indicates the successful installation of Astra Control Center.

```
{"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. The username for logging into Astra Control Center is the email address of the administrator provided in the CRD file and the password is a string `ACC-` appended to the Astra Control Center UUID. Run the following command:

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



In this example, the password is `ACC-345c55a5-bf2e-21f0-84b8-b6f2bce5e95f`.



4. Get the traefik service load balancer IP if the ingressType is AccTraefik.

```
[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. Add an entry in the DNS server pointing the FQDN provided in the Astra Control Center CRD file to the EXTERNAL-IP of the traefik service.

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. Log into the Astra Control Center GUI by browsing its FQDN.



7. When you log into Astra Control Center GUI for the first time using the admin email address provided in CRD, you need to change the password.



8. If you wish to add a user to Astra Control Center, navigate to Account > Users, click Add, enter the details of the user, and click Add.

**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

▼

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](#).

- Astra Control Center requires a license for all of its functionalities to work. To add a license, navigate to Account > License, click Add License, and upload the license file.

Account

Users

Credentials

Notifications

License

Connections

ASTRA CONTROL CENTER LICENSE

To get started with Astra Control Center, select Add license to manually upload the file.

Add license

ADD LICENSE

Select and add a license file.

License file

EvalNLF-AstraControlCenter-480Cores(vCPU)-100000002-ACC60f19...

⬆

✕

Cancel

Add



If you encounter issues with the install or configuration of NetApp Astra Control Center, the knowledge base of known issues is available [here](#).

Next: [Register your Tanzu Kubernetes clusters.](#)

## Register your VMware Tanzu Kubernetes Clusters with the Astra Control Center

To enable the Astra Control Center to manage your workloads, you must first register your Tanzu Kubernetes clusters.

## Register VMware Tanzu Kubernetes clusters

1. The first step is to add the Tanzu Kubernetes clusters to the Astra Control Center and manage them. Go to Clusters and click Add a Cluster, upload the kubeconfig file for the Tanzu Kubernetes cluster, and click Select Storage.

 **Add Kubernetes cluster**

STEP 1/3: CREDENTIALS

X

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 Center detects the eligible storage classes. Now select the way that storageclass provisions volumes using Trident backed by an SVM on NetApp ONTAP and click Review. In the next pane, verify the details and click Add Cluster.
3. When the cluster is added, it moves to the Discovering status while Astra Control Center inspects it and installs the necessary agents. The cluster status changes to `Healthy` after it is successfully registered.




## Clusters

Actions

+ Add Kubernetes cluster

Search

1-1 of 1 entries

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



All Tanzu Kubernetes clusters to be managed by Astra Control Center should have access to the image registry that was used for its installation as the agents installed on the managed clusters pull the images from that registry.

4. Import ONTAP clusters as storage resources to be managed as backends by Astra Control Center. When Tanzu Kubernetes clusters are added to Astra and a storageclass is configured, it automatically discovers and inspects the ONTAP cluster backing the storageclass but does not import it into the Astra Control Center to be managed.

**Backends**

+ Add

Search

★

🔍 1

1-1 of 1 entries

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

- To import the ONTAP clusters, navigate to Backends, click the dropdown, and select Manage next to the ONTAP cluster to be managed. Enter the ONTAP cluster credentials, click Review Information, and then click Import Storage Backend.

**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 →

- After the backends are added, the status changes to Available. These backends now have the information about the persistent volumes in the Tanzu Kubernetes cluster and the corresponding volumes on the ONTAP system.



## Backends

+ Add

Search

★ 🔍

1-1 of 1 entries < >

Name ↓	State	Capacity	Throughput	Type	Cluster	Cloud	Actions
<a href="#">K8s-Ontap</a>	✔ Available	Not available yet	Not available yet	ONTAP 9.9.1	Not applicable	Not applicable	⋮

7. For backup and restore across Tanzu Kubernetes clusters using Astra Control Center, you must provision an object storage bucket that supports the S3 protocol. Currently supported options are ONTAP S3, StorageGRID, AWS S3, and Microsoft Azure Blob storage. For the purpose of this installation, we are going to configure an AWS S3 bucket. Go to Buckets, click Add bucket, and select Generic S3. Enter the details about the S3 bucket and credentials to access it, click the checkbox Make this Bucket the Default Bucket for the Cloud, and then click Add.

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](#)

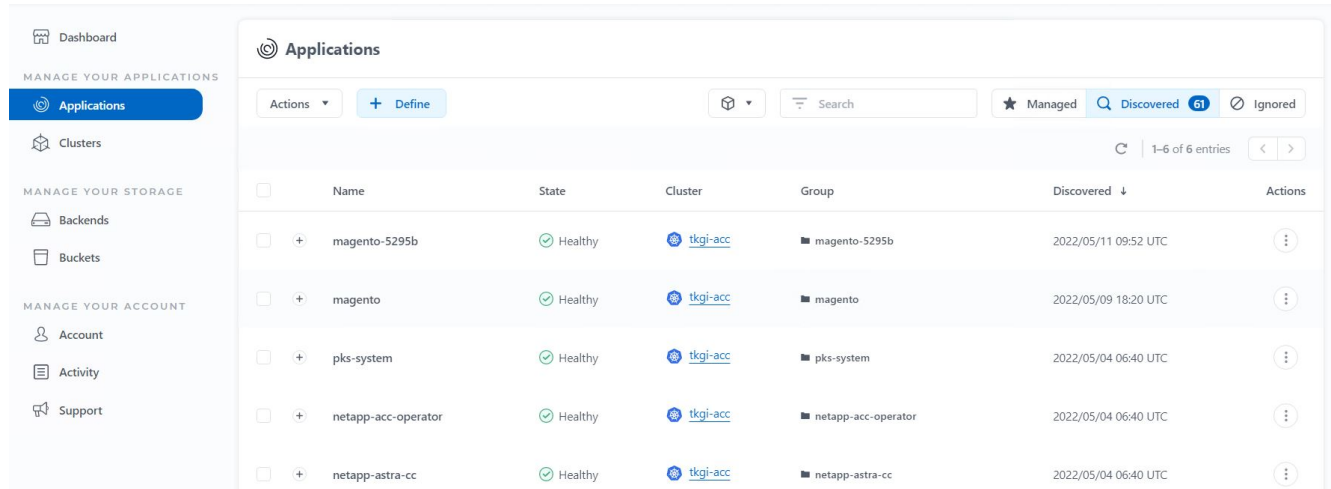
Next: Choose the Applications To Protect.

## Choose the applications to protect

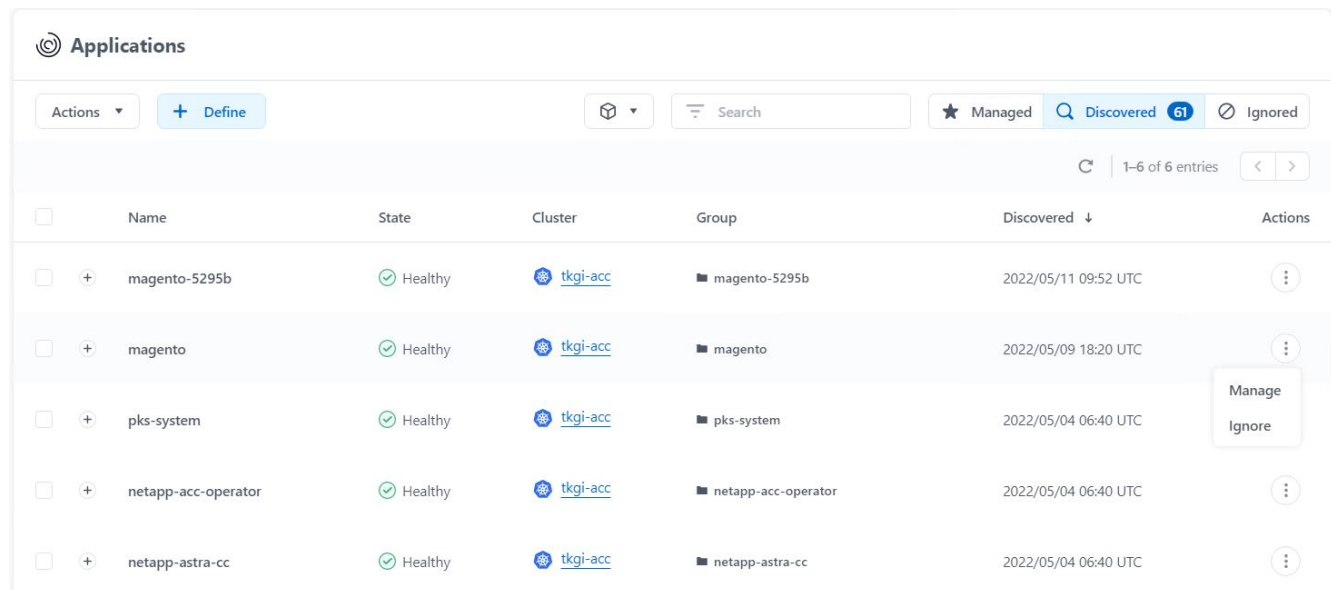
After you have registered your Tanzu Kubernetes clusters, you can discover the applications that are deployed and manage them via the Astra Control Center.

## Manage applications

1. After the Tanzu Kubernetes clusters and ONTAP backends are registered with the Astra Control Center, the control center automatically starts discovering the applications in all the namespaces that are using the storageclass configured with the specified ONTAP backend.



2. Navigate to Apps > Discovered and click the dropdown menu next to the application you would like to manage using Astra. Then click Manage.



3. The application enters the Available state and can be viewed under the Managed tab in the Apps section.

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"/>	<a href="#">magento</a>	<div>Healthy</div>	<div>Unprotected</div>	<div>tkgi-acc</div>	<div>magento</div>	2022/05/09 18:20 UTC	<div></div>

Next: [Protect Your applications.](#)

## Protect your applications

After application workloads are managed by Astra Control Center, you can configure the protection settings for those workloads.

### Create an application snapshot

A snapshot of an application creates an ONTAP Snapshot copy and a copy of the application metadata that can be used to restore or clone the application to a specific point in time based on that Snapshot copy.

1. To take a snapshot of the application, navigate to the Apps > Managed tab and click the application you would like to make a Snapshot copy of. Click the dropdown menu next to the application name and click Snapshot.

🔄
Actions ▾

**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](#)

Snapshot  
 Backup  
 Clone  
 Restore  
 Unmanage

2. Enter the snapshot details, click Next, and then click Snapshot. It takes about a minute to create the snapshot, and the status becomes Available after the snapshot is successfully created.



 Snapshot namespace application

STEP 1/2: DETAILS

X

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 →

## Create an application backup

1. For the backup and restore of managed applications in the Astra Control Center, you must configure superuser settings for the backing ONTAP systems as a prerequisite. To do so, enter the following commands.

2. To create a backup of the managed application in the Astra Control Center, navigate to the Apps > Managed tab and click the application that you want to take a backup of. Click the dropdown menu next to the application name and click Backup.

3. Enter the backup details, select the object storage bucket to hold the backup files, click Next, and, after reviewing the details, click Backup. Depending on the size of the application and data, the backup can take several minutes, and the status of the backup becomes Available after the backup is completed

successfully.

 **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-astro/na-astro-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 →

## Restoring an application

At the push of a button, you can restore an application to the originating namespace in the same cluster or to a remote cluster for application protection and disaster recovery purposes.

1. To restore an application, navigate to the Apps > Managed tab and click the app in question. Click the dropdown menu next to the application name and click Restore.


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
 

 tkg

2. Enter the name of the restore namespace, select the cluster you want to restore it to, and choose if you want to restore it from an existing snapshot or from a backup of the application. Click Next.

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"/> <div>magento-backup-20220516212730</div>	<div>Healthy</div>	<div>On-Demand</div>	<div>2022/05/16 21:27 UTC</div>

Cancel

Next →

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

- On the review pane, enter `restore` and click Restore after you have reviewed the details.

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 ✓

- The new application goes to the Restoring state while Astra Control Center restores the application on the selected cluster. After all the resources of the application are installed and detected by Astra, the application goes to the Available state.

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"/>	<a href="#">magento</a>	✓ Healthy	⚠ Unprotected	<a href="#">tkgi-acc</a>	📁 magento	2022/05/09 18:20 UTC	

## Cloning an application

You can clone an application to the originating cluster or to a remote cluster for dev/test or application protection and disaster recovery purposes. Cloning an application within the same cluster on the same storage backend uses NetApp FlexClone technology, which clones the PVCs instantly and saves storage space.

1. To clone an application, navigate to the Apps > Managed tab and click the app in question. Click the dropdown menu next to the application name and click Clone.

Actions

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

k8s

Snapshot  
Backup  
Clone  
Restore  
Unmanage

2. Enter the details of the new namespace, select the cluster you want to clone it to, and choose if you want to clone it from an existing snapshot, from a backup, or from the current state of the application. Click Next and then click Clone on the review pane after you have reviewed the details.

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 →

- The new application goes to the Discovering state while Astra Control Center creates the application on the selected cluster. After all the resources of the application are installed and detected by Astra, the application goes to the Available state.

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"/>	<a href="#">magento-bef7f</a>	✓ Healthy	⚠️ Unprotected	tkgi-acc	magento-bef7f	2022/05/16 21:31 UTC	⋮
<input type="checkbox"/>	<a href="#">magento</a>	✓ Healthy	ℹ️ Partially protected	tkgi-acc	magento	2022/05/09 18:20 UTC	⋮

Next: Videos and demos: VMware Tanzu with NetApp.

## Astra Trident overview

Astra Trident is an open-source, fully supported storage orchestrator for containers and Kubernetes distributions like Red Hat OpenShift, VMware Tanzu, Anthos by Google Cloud, Rancher etc. Trident works with the entire NetApp storage portfolio, including the NetApp ONTAP and Element storage systems, and it also supports NFS and iSCSI connections. Trident accelerates the DevOps workflow by allowing end users to provision and manage storage from their NetApp storage systems without requiring intervention from a storage administrator.

An administrator can configure a number of storage backends based on project needs and storage system

models that enable advanced storage features, including compression, specific disk types, or QoS levels that guarantee a certain level of performance. After they are defined, these backends can be used by developers in their projects to create persistent volume claims (PVCs) and to attach persistent storage to their containers on demand.



Astra Trident has a rapid development cycle and, like Kubernetes, is released four times a year.

The latest version of Astra Trident is 22.04 released in April 2022. A support matrix for what version of Trident has been tested with which Kubernetes distribution can be found [here](#).

Starting with the 20.04 release, Trident setup is performed by the Trident operator. The operator makes large scale deployments easier and provides additional support, including self healing for pods that are deployed as a part of the Trident install.

With the 21.01 release, a Helm chart was made available to ease the installation of the Trident Operator.

## Deploy Trident operator using Helm

1. First set the location of the user cluster's kubeconfig file as an environment variable so that you don't have to reference it, because Trident has no option to pass this file.

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

2. Add the NetApp Astra Trident helm repository.

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

3. Update the helm repositories.

```
[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. Create a new namespace for the installation of Trident.

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

5. Create a secret with DockerHub credentials to download the Astra Trident images.

```
[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. For user or workload clusters managed by TKGS (vSphere with Tanzu) or TKG with management cluster deployments, complete the following procedure to install Astra Trident:

- a. Ensure that the logged in user has the permissions to create service accounts in trident namespace and that the service accounts in trident namespace have the permissions to create pods.
- b. Run the below helm command to install Trident operator in the namespace created.

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

7. For a user or workload cluster managed by TKGI deployments, run the following helm command to install Trident operator in the namespace created.

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

8. Verify that the Trident pods are up and running.

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        |
+-----+
```

## Create storage-system backends

After completing the Astra Trident Operator install, you must configure the backend for the specific NetApp storage platform you are using. Follow the links below to continue the setup and configuration of Astra Trident.

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

Next: [Videos and demos: VMware Tanzu with NetApp.](#)

## NetApp ONTAP NFS configuration

To enable Trident integration with the NetApp ONTAP storage system via NFS, you must create a backend that enables communication with the storage system. We configure a basic backend in this solution, but if you are looking for more customized options, visit the documentation [here](#).

### Create an SVM in ONTAP

1. Log into ONTAP System Manager, navigate to Storage > Storage VMs, and click Add.
2. Enter a name for the SVM, enable the NFS protocol, check the Allow NFS Client Access checkbox, and add the subnets that your worker nodes are on in the export policy rules for allowing the volumes to be mounted as PVs in your workload clusters.



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



If you are using NAT'ed deployment of user clusters or workload clusters with NSX-T, you need to add the Egress subnet (in the case of TKGS0 or the Floating IP subnet (in the case of TKGI) to the export policy rules.

3. Provide the details for data LIFs and the details for SVM administration account, and then click Save.

## 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. Assign the aggregates to an SVM. Navigate to Storage > Storage VMs, click the ellipsis next to the newly created SVM and then click Edit. Check the Limit Volume Creation to Preferred Local Tiers checkbox and attach the required aggregates to it.

## 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. In case of NAT'ed deployments of user or workload clusters on which Trident is to be installed, the storage mount request might arrive from a non-standard port due to SNAT. By default, ONTAP only allows the volume mount requests when originated from root port. Thus, log into ONTAP CLI and modify the setting to

allow mount requests from non-standard ports.

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

## Create backends and StorageClasses

1. For NetApp ONTAP systems serving NFS, create a backend config file on the jumphost with the backendName, managementLIF, dataLIF, svm, username, password, and other details.

```
{
  "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"
}
```



It is a best practice to define the custom backendName value as a combination of the storageDriverName and the dataLIF that is serving NFS for easy identification.

2. Create the Trident backend by running the following command.

```
[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. With the backend created, you must next create a storage class. The following sample storage class definition highlights the required and basic fields. The parameter backendType should reflect the storage driver from the newly created Trident backend.

```

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

```

4. Create the storage class by running the kubectl command.

```

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

```

5. With the storage class created, you must then create the first persistent volume claim (PVC). A sample PVC definition is given below. Make sure that the `storageClassName` field matches the name of the storage class just created. The PVC definition can be further customized as required depending upon the workload to be provisioned.

```

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

```

6. Create the PVC by issuing the kubectl command. Creation can take some time depending on the size of the backing volume being created, so you can watch the process as it completes.

```

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

Next: [Videos and demos: VMware Tanzu with NetApp.](#)

## NetApp ONTAP iSCSI configuration

To integrate NetApp ONTAP storage system with VMware Tanzu Kubernetes clusters for persistent volumes via iSCSI, the first step is to prepare the nodes by logging into each node and configuring the iSCSI utilities or packages to mount iSCSI volumes. To do so, follow the procedure laid out in this [link](#).



NetApp does not recommend this procedure for NAT'ed deployments of VMware Tanzu Kubernetes clusters.



TKGI uses Bosh VMs as nodes for Tanzu Kubernetes clusters that run immutable configuration images, and any manual changes of iSCSI packages on Bosh VMs do not remain persistent across reboots. Therefore, NetApp recommends using NFS volumes for persistent storage for Tanzu Kubernetes clusters deployed and operated by TKGI.

After the cluster nodes are prepared for iSCSI volumes, you must create a backend that enables communication with the storage system. We configured a basic backend in this solution, but, if you are looking for more customized options, visit the documentation [here](#).

### Create an SVM in ONTAP

To create an SVM in ONTAP, complete the following steps:

1. Log into ONTAP System Manager, navigate to Storage > Storage VMs, and click Add.
2. Enter a name for the SVM, enable the iSCSI protocol, and then provide details for the data LIFs.

# 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	SUBNET MASK	GATEWAY	BROADCAST DOMAIN
10.61.181.231	24	10.61.181.1	Defa...

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

IP ADDRESS	SUBNET MASK	GATEWAY	BROADCAST DOMAIN
10.61.181.232	24	10.61.181.1	Defa...

3. Enter the details for the SVM administration account, and then click Save.

---

## Storage VM Administration

☒ Manage administrator account

USER NAME

vsadmin

PASSWORD

.....

CONFIRM PASSWORD

.....

☐ Add a network interface for storage VM management.

Save

Cancel

4. To assign the aggregates to the SVM, navigate to Storage > Storage VMs, click the ellipsis next to the newly created SVM, and then click Edit. Check the Limit Volume Creation to Preferred Local Tiers checkbox, and attach the required aggregates to it.



## 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

### Create backends and StorageClasses

1. For NetApp ONTAP systems serving NFS, create a backend config file on the jumphost with the backendName, managementLIF, dataLIF, svm, username, password, and other details.

```
{
  "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. Create the Trident backend by running the following command.

```
[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. After you create a backend, you must next create a storage class. The following sample storage class definition highlights the required and basic fields. The parameter `backendType` should reflect the storage driver from the newly created Trident backend. Also note the name-field value, which must be referenced in a later step.

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



There is an optional field called `fsType` that is defined in this file. In iSCSI backends, this value can be set to a specific Linux filesystem type (XFS, ext4, and so on) or can be deleted to allow Tanzu Kubernetes clusters to decide what filesystem to use.

4. Create the storage class by running the kubectl command.

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

5. With the storage class created, you must then create the first persistent volume claim (PVC). A sample PVC definition is given below. Make sure that the `storageClassName` field matches the name of the storage class just created. The PVC definition can be further customized as required depending upon the workload to be provisioned.

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

6. Create the PVC by issuing the kubectl command. Creation can take some time depending on the size of the backing volume being created, so you can watch the process as it completes.

```
[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

[Next: Solution validation/use cases.](#)

## Copyright Information

Copyright © 2022 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system-without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## Trademark Information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.