

Using Astra Trident with Kubernetes



1 © 2022 NetApp, Inc. All rights reserved.

STRSW-ILT-UATWK

Kubernetes Version 1.23.1

Astra Trident 22.01.0

ONTAP 9.9.1

Welcome

Getting started

- Schedule (start, stop, breaks, and breakout sessions)
- Activities and participation
- Materials
- Equipment check
- Support

Classroom sessions

- Sign-in sheet
- Refreshments
- Phones to vibrate
- Alarm signal
- Evacuation procedure
- Electrical safety

Virtual sessions

- Collaboration tools
- Ground rules
- Phones and headsets

NetApp 2 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Review the schedule, class structure, and logistics details.

If students have not retrieved their courseware, direct them to the download site.

Use the Getting Started exercise in the module to do an equipment check of connectivity to the lab environment.

For a course delivered virtually, be sure to discuss the following topics.

- Collaboration tools: Explain the collaboration tools that are used during the class. If necessary, refer the students to a demonstration video or demonstrate the tool.
- Remind students to turn off distractions, be prepared to be called on, raise hand or use chat to ask questions, be patient waiting for responses to chat, and use the coffee cup icon to indicate temporary absence.
- Suggest appropriate phones and headset. Non-battery telephones have best results. Voice over IP (VoIP) might be spotty.
- Explain how breakout sessions work, if applicable.

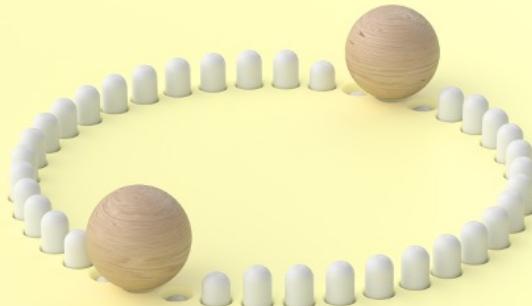
Student Notes

Set your phone to vibrate to prevent disturbing your fellow students. NetApp realizes that work does not always stop when you are in training. If you need to take a call, step outside the classroom.

Instructions

Tell everyone the following:

- Your name
- The company you work for
- Your Docker experience
- Your cloud experience (AWS, Google Cloud, Azure)
- Your Kubernetes experience
- Your NetApp Astra family experience



Instructor Notes

Use this slide to introduce yourself and to prompt students to state their names and their professions.

In VILT classes, you can have students introduce themselves through the chat window or through the conference connection. Remind students to unmute and mute their lines when speaking.

In VILT classes, you can insert the optional Map Your Location slide, from the Interactive Formats section, to personalize introductions and give students the opportunity to practice with the whiteboard tools.

Student Notes

Take time to meet and get to know one another. If you are participating in a NetApp Virtual Live class, your instructor asks you to use the chat window or a conference connection to speak. If you are using a conference connection, unmute your line to speak and be sure to mute again after you speak.

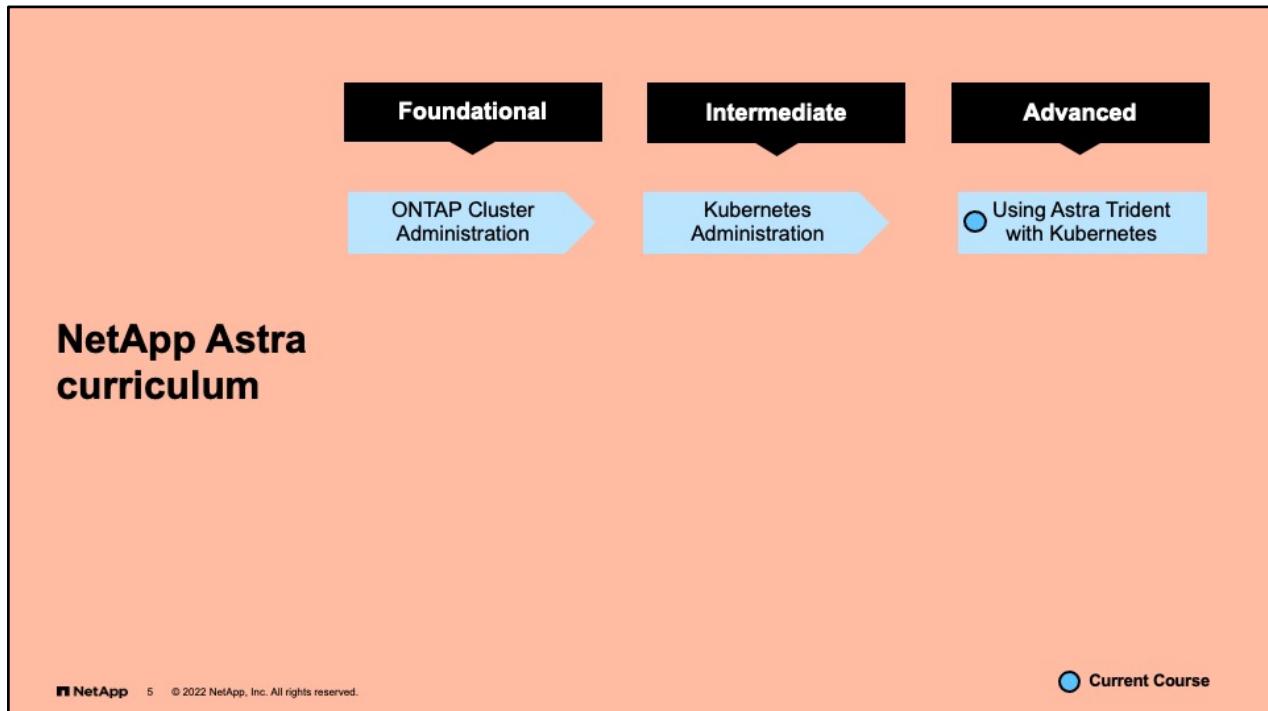
About this course

This course focuses on enabling you to do the following:

- Review how containers use persistent storage
- Describe the storage concepts that are available in Kubernetes
- Explain how NetApp Astra Trident makes managing persistent storage easier
- Install Astra Trident in a Kubernetes cluster
- Configure back ends, storage classes, and persistent volumes to use storage that is managed by Astra Trident
- Use Astra Trident to manage frequently seen use scenarios
- Monitor Astra Trident by using Prometheus and Grafana

Instructor Notes

Student Notes



Instructor Notes

Student Notes

What this course is not

A course on Kubernetes administration

Instructor Notes

Student Notes

Course modules

Morning

- Introduction
- Module 0: Welcome
- Module 1: Kubernetes storage overview
- Module 2: Astra Trident installation

Afternoon

- Module 3: Astra Trident configuration
- Module 4: Astra Trident use scenarios

Instructor Notes

Student Notes

This schedule is based on average completion times for modules. Each class includes learners with different backgrounds and experience levels. This situation means that some modules might take more or less time to complete. Your instructor will adjust the schedule accordingly for breaks, meals, and the start time of each module.

Course appendixes

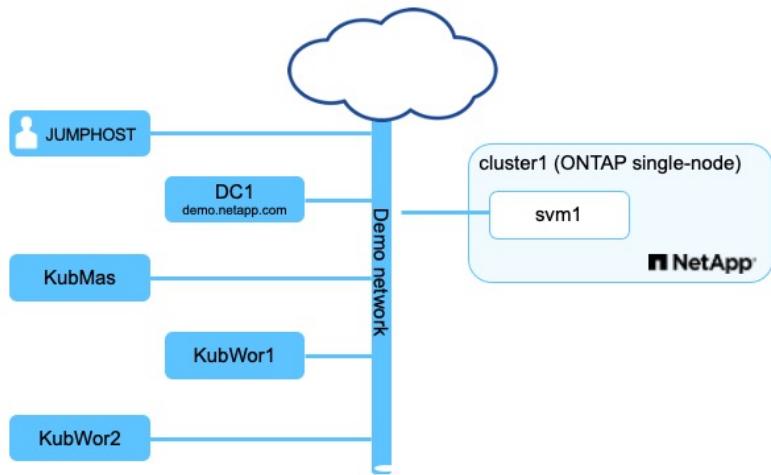
- Appendix 1: Kubernetes-related certifications
- Appendix 2: An introduction to operators
- Appendix 3: Astra Trident monitoring

Instructor Notes

Student Notes

Exercise equipment: Basic architecture

Location	User name	Password
Windows	Administrator	Netapp1!
KubMas	root	Netapp1!
KubWor1	root	Netapp1!
KubWor2	root	Netapp1!
ONTAP	admin	Netapp1!



Content: <https://github.com/NetApp-Learning-Services/STRSW-ILT-UATWK>

NetApp 9 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

For equipment-based offerings, provide an equipment kit and connection information to each participant. Explain each server in the exercise kit.

Verify that license codes are installed. Refer to the hidden Courseware Licenses slide at the end of this module.

Contact information for courseware licenses:

- Courses offered directly through NetApp University: nq-nau-labsupport@netapp.com
- Courses provided by Authorized Learning Partners: Contact your direct manager or supervisor or NetApp University ALP Program Manager, nq-netappu-nci@netapp.com.

Student Notes

Launch your exercise equipment kit from your laptop or from the classroom desktop. To connect to your exercise equipment, use Remote Desktop Connection or the NetApp Learning Services portal.

The Windows Server is your jump host to access the lab environment.



Complete an exercise

Module 0: Welcome

Checking the exercise equipment

- Task 1: Connect to your NetApp ONTAP cluster
- Task 2: Verify that required license codes are configured
- Task 3: Verify that the required software and tools are installed
- Task 4: Configure Kubernetes administrator access on the jump host
- Task 5: Add the courseware to the integrated development environment (IDE)
- Task 6: Work with YAML files in the IDE

Estimated duration:
60 minutes

 10 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

See the instructions in your Exercise Guide.



Share your experiences

Roundtable questions for the equipment-based exercises

Checking the exercise equipment

- Do you have questions about your lab equipment?
- Do you have an issue to report?

Instructor Notes

Display while students are completing the exercise(s)

Student Notes



Module 1

Kubernetes storage overview

 NetApp 1 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

About this module

This module focuses on enabling you to do the following:

- Introduce Kubernetes and the need for persistent storage
- Describe how containers use persistent storage
- Summarize the storage concepts of Kubernetes
- Explain how Astra Trident enables dynamic provisioning of Kubernetes persistent volumes

Instructor Notes

Student Notes



Persistent storage in Kubernetes

 3 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Modern IT is adopting Kubernetes

Benefits of a containerized environment



Agility

- Build and deploy apps faster
- Automate almost everything

Cost optimization

- Pay for what you use, not what you own
- Smaller apps > Fewer resources > Lower cost

Hybrid and reusable

- Reuse existing pipeline in the public, private, and hybrid cloud
- Build once, deploy multiple times anywhere

Stability

- Help developers deliver business value above all
- Focus on business outcomes, not technology outcomes



© 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

The missing piece

Data persistence

- Containers: ephemeral
- Applications: not ephemeral
- Need for persistence:
 - Data protection
 - Availability
 - Scalability
 - Performance

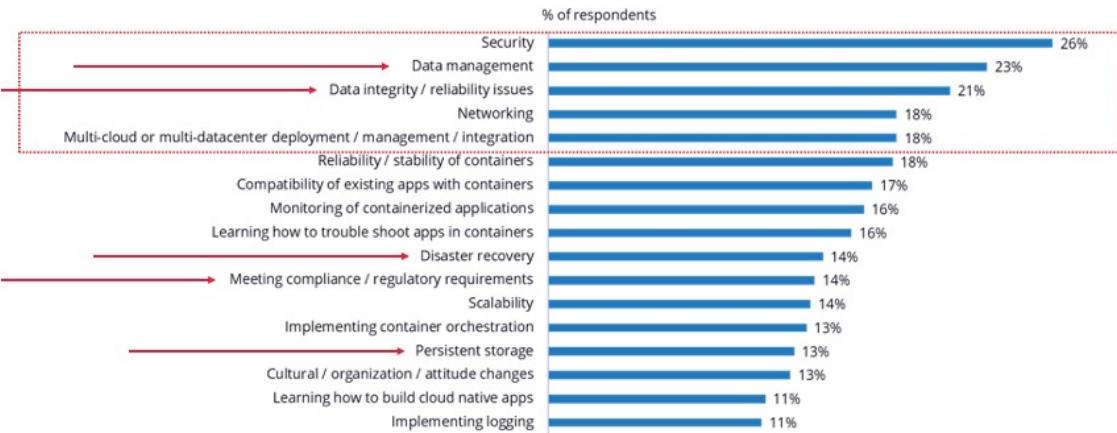


Instructor Notes

Student Notes

Top challenges when deploying containers

Data persistence, management, compliance, and disaster recovery are top challenges



Source: IDC's Enterprise Infrastructure Market Pulse, Q2 2019, IDC, April/May 2019

NetApp 6 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes



Container storage

 NetApp 7 © 2022 NetApp, Inc. All rights reserved.

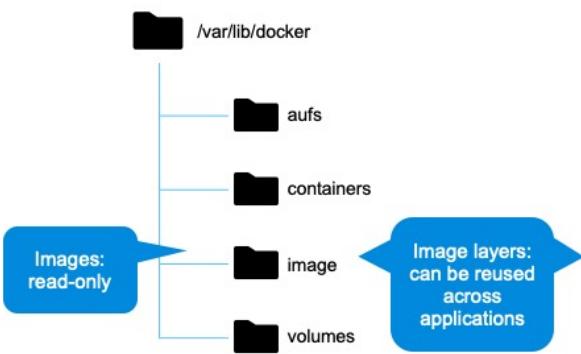
Instructor Notes

Student Notes

Container storage

Layers are generated from the container build file

- Containers store files in a file structure:



- Following is the layered architecture:

Container build file (dockerfile):

```
FROM Ubuntu
RUN apt-get update && apt-get -y install python
RUN pip install flask flask-mysql
COPY . /opt/code
ENTRYPOINT FLASK_APP=/opt/code/app.py flask run
```

- Build the container:

```
docker build dockerfile -t myapp
```

Layer 1: Base Ubuntu Layer	120 MB
Layer 2: APT update and python	306 MB
Layer 3: PIP packages installed	6.3 MB
Layer 4: Add custom app source	229 B
Layer 5: Change entrypoint	0 B

Instructor Notes

Student Notes

Container storage layers

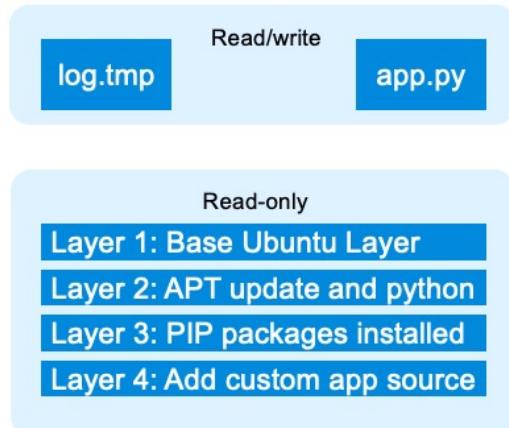
Read-only layers with a read/write container cache

- Images layers:

- Are predefined and read-only
- Write new files to a read/write "container" layer during the container's execution
- Write modified files from the read-only layers (for example, app.py) to the "container" layer (an operation called copy-on-write)

- Read/write (container) layer:

- Persists during the life of the container
- Is deleted when the container is stopped



 NetApp 9 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

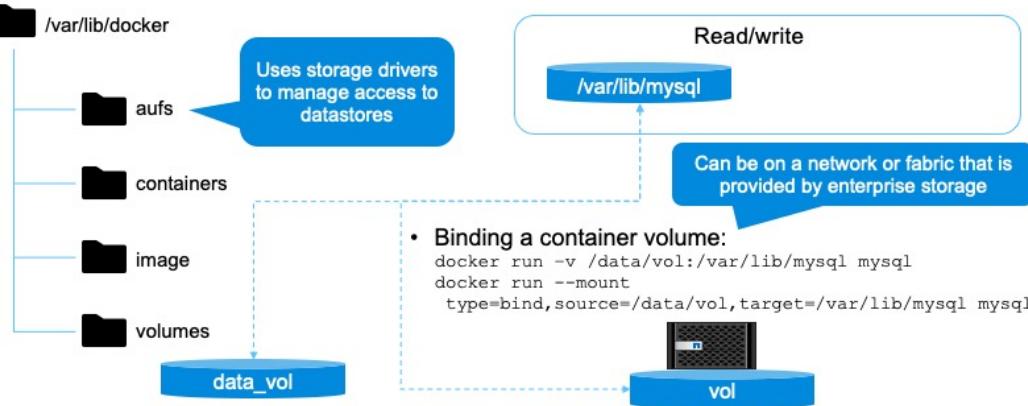
Student Notes

Container volumes

Mounting and binding

- Mounting a container volume:

```
docker run -v data_vol:/var/lib/mysql mysql
```



NetApp 10 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Kubernetes storage concepts

 NetApp 11 © 2022 NetApp, Inc. All rights reserved.

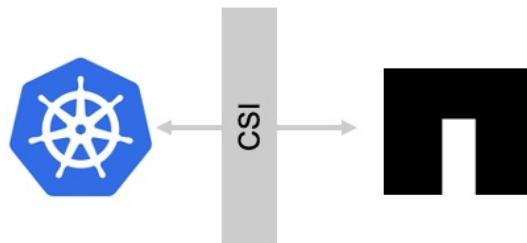
Instructor Notes

Student Notes

Kubernetes uses and extends container storage

Managing volumes with legacy drivers and the new CSI standard

- Provides “legacy” storage volumes:
 - Local
 - Azure File Storage
 - ConfigMap
 - GCE Docker
 - VMware vSphere Storage
 - iSCSI
 - NFS
 - FC
 - Amazon Elastic Block Store
 - Host Path
- Adopted new Container Storage Interface (CSI) standard:
 - Determines how to create, manage, and delete volumes
 - Has been adopted by NetApp



Instructor Notes

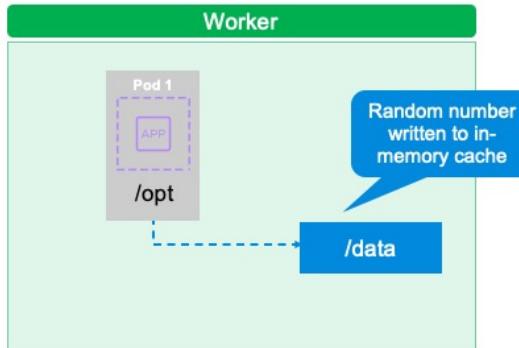
Student Notes

For more information about the legacy storage volumes, see
<https://kubernetes.io/docs/concepts/storage/volumes/>.

For more information about the CSI standard, see <https://github.com/container-storage-interface>

Volumes in Kubernetes

Simple volume with emptyDir



Scratch space volume example:

```
apiVersion: v1
kind: Pod
metadata:
  name: Random-number
spec:
  containers:
  - name: alpine
    image: alpine
    command: ['/bin/sh', '-c']
    args: ['shuf -i 0-99 n1 >> /opt/number.out;']
  volumeMounts:
  - mountPath: /opt
    name: data_vol
  volumes:
  - name: data_vol
    emptyDir: {}
```

Data not persisted when another instance of the pod is created

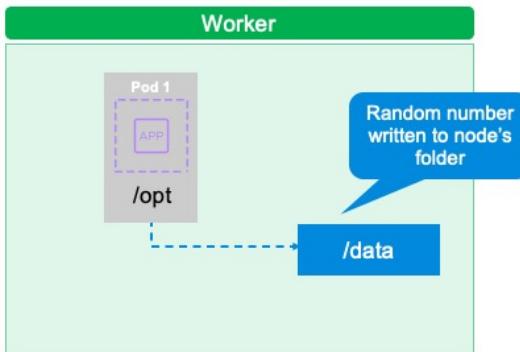
NetApp 13 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Volumes in Kubernetes

Simple volume with HostPath



A node's directory volume example:

```
apiVersion: v1
kind: Pod
metadata:
  name: Random-number
spec:
  containers:
  - name: alpine
    image: alpine
    command: ['shuf -i 0-99 n1 >> /opt/number.out;']
  volumeMounts:
  - mountPath: /opt
    name: data_vol
  volumes:
  - name: data_vol
    hostPath:
      path: /data
      type: Directory
```

Data not available
to all nodes in a
multinode cluster

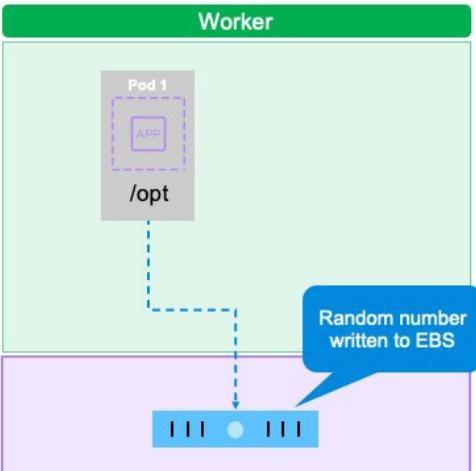
NetApp 14 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Volumes in Kubernetes

Simple volume with Amazon Elastic Block Store



Amazon Elastic Block Store (Amazon EBS) example:

```
apiVersion: v1
kind: Pod
metadata:
  name: Random-number
spec:
  containers:
  - name: alpine
    image: alpine
    command: ['bin/sh', '-c']
    args: ['shuf -i 0-99 n1 > /opt/number.out;']
  volumeMounts:
  - mountPath: /opt
    name: data_vol
  volumes:
  - name: data_vol
    awsElasticBlockStore:
      volumeId: ...
      fsType: ext4
```



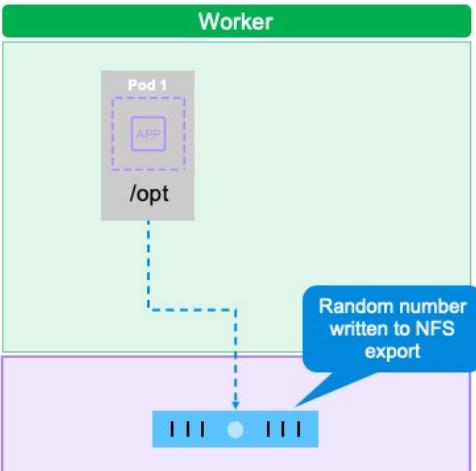
© 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Volumes in Kubernetes

Simple volume with NFS



NFS example:

```
apiVersion: v1
kind: Pod
metadata:
  name: Random-number
spec:
  containers:
  - name: alpine
    image: alpine
    command: ['"/bin/sh', '-c']
    args: ['shuf -i 0-99 n1 >> /opt/number.out;']
  volumeMounts:
  - mountPath: /opt
    name: data_vol
  volumes:
  - name: data_vol
    nfs:
      server: 192.168.0.31
      path: /nfs
      readOnly: false
```

NetApp 16 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Volumes versus persistent volumes

Overview



Kubernetes volumes

- Are tightly coupled with a pod definition
- Out live any container that is running in a pod (preserved across container restarts)
- Live for the lifetime of the pod
- Have the following persistence:
 - Some volume types, like emptyDir volumes, deleted when a pod is removed
 - Some volume types, like Amazon EBS volumes and NFS, persist after the pod removal

Persistent volumes (PV)

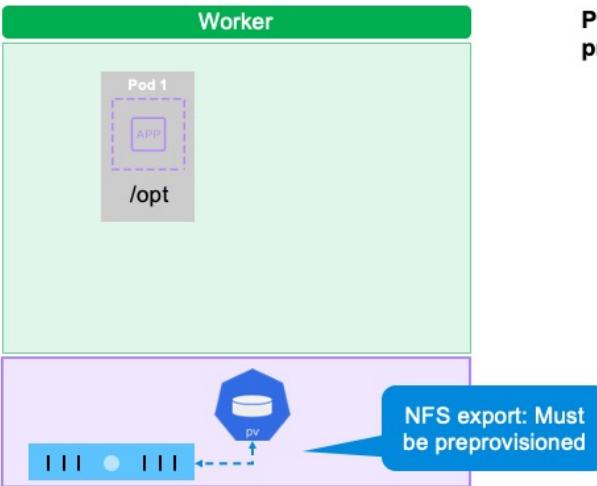
- Are loosely coupled with a pod definition
- Enable storage administrators to preprovision the storage for pods
- Are used by pods through a persistent volume claim (PVC)

Instructor Notes

Student Notes

Persistent volumes in Kubernetes

Simple static provisioned persistent volume with NetApp NFS



Persistent volumes (PVs) enable volumes to be pre-provisioned by administrators:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-data
spec:
  accessMode:
    - ReadWriteOnce
  nfs:
    server: 192.168.0.31
    path: /nfs
```

```
kubectl create -f pv.yml
```

Access mode
options:
ReadOnlyMany
ReadWriteOnce
ReadWriteMany



18 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

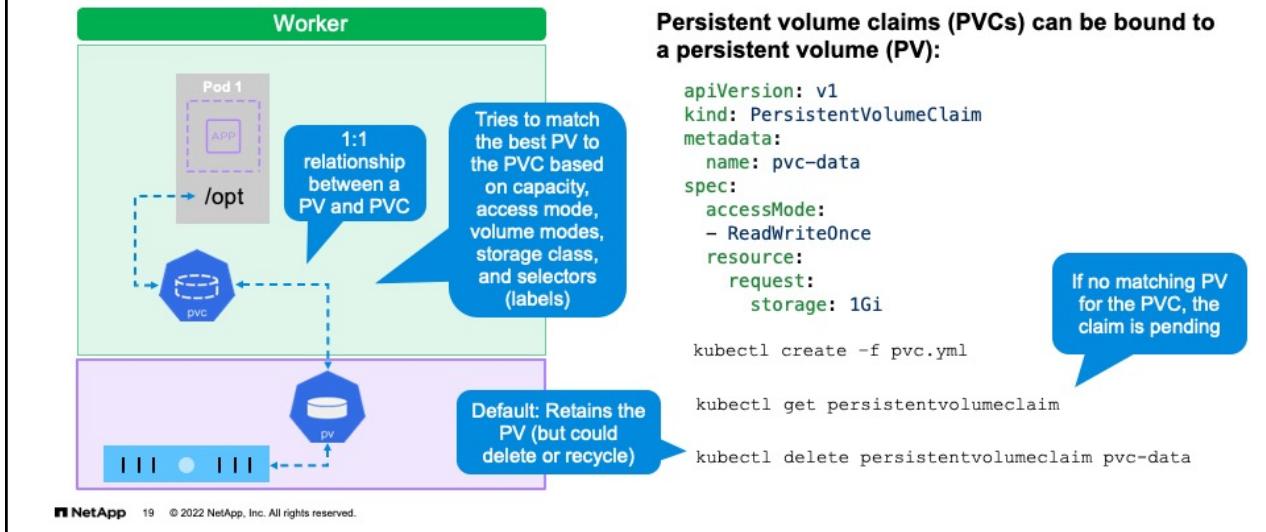
The access modes:

- **ReadWriteOnce:** The volume can be mounted as read/write by a single node.
- **ReadOnlyMany:** The volume can be mounted read-only by many pods on different nodes.
- **ReadWriteMany:** The volume can be mounted as read/write by many pods on different nodes.

For more information, see <https://kubernetes.io/docs/concepts/storage/persistent-volumes/>.

Persistent volumes claims in Kubernetes

Simple static provisioned persistent volume claim with NetApp NFS



Instructor Notes

Student Notes

If you delete a persistent volume claim, you have three choices what to do with the persistent volume. Use the `persistentVolumeReclaimPolicy` to retain, delete, or recycle (delete the data and recreate) the persistent volume.

Persistent volumes claims in Kubernetes

Using a simple static provisioned persistent volume claim with NetApp NFS

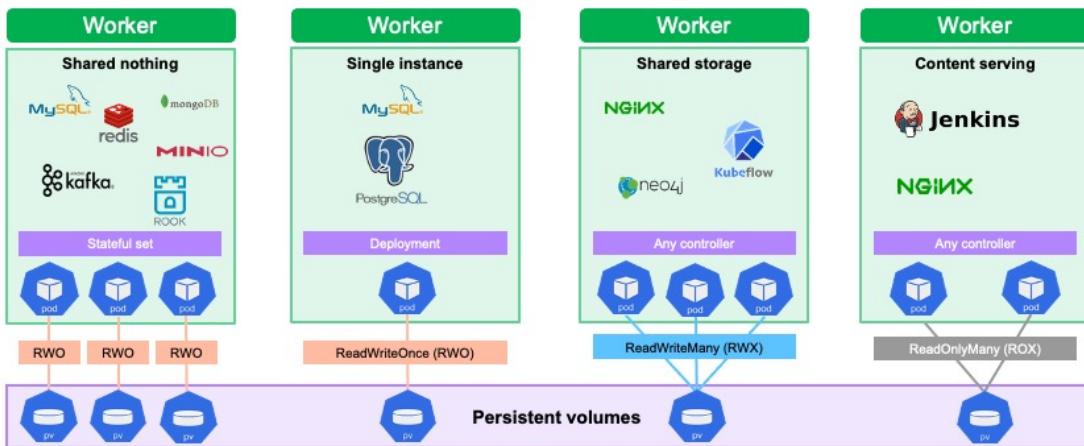


Instructor Notes

Student Notes

Persistent volume claim access modes

Different applications have different needs



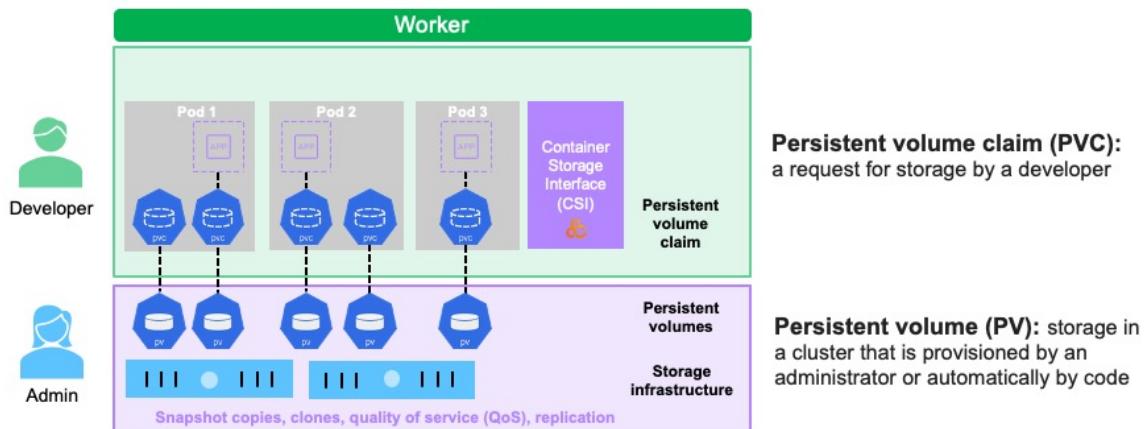
Instructor Notes

Student Notes

A volume can only be mounted by using one access mode at a time, even if it supports many. For example, a GCEPersistentDisk can be mounted as ReadWriteOnce by a single node or ReadOnlyMany by many nodes, but not both access modes at the same time.

Persistent storage integration

Two different users and roles



NetApp 22 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Static provisioning summary

Preconfigured storage method

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-data
spec:
  accessMode:
  - ReadWriteOnce
  nfs:
    server: 192.168.0.31
    path: /nfs
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-data
spec:
  accessMode:
  - ReadWriteOnce
  resource:
    request:
      storage: 1Gi
```

```
apiVersion: v1
kind: Pod
metadata:
  name: Random-number
spec:
  containers:
  - name: alpine
    image: alpine
    volumeMounts:
    - mountPath: /opt
      name: data_vol
  volumes:
  - name: data_vol
    persistentVolumeClaim:
      claimName: pvc-data
```

pv-data

pvc-data

mount

 NetApp 23 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes



Dynamic provisioning with Astra Trident

 NetApp 24 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Storage classes

Dynamically provisioning storage when requested



Dynamic provisioning:

- Uses a storage class to have the storage created when a persistent volume claim is requested
- No need to pre-create a PV
- PV created when a PVC is requested

Astra Trident is the provisioner for NetApp:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-trident
provisioner: csi.trident.netapp.io
parameters:
  <Additional Trident Parameters>
```

Instructor Notes

Student Notes

Dynamic provisioning basics

Provisioning storage by using a storage class

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-trident
provisioner: csi.trident.netapp.io
parameters:
  <Additional Trident Parameters>
```

Replace the PV
with a storage class

When requesting a
PVC, the storage class
creates a PV

sc-trident

pv

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-data
spec:
  accessMode:
    - ReadWriteOnce
  storageClassName: sc-trident
  resource:
    request:
      storage: 1Gi
```

Specify the storage
class to be used

pvc-data

```
apiVersion: v1
kind: Pod
metadata:
  name: Random-number
spec:
  containers:
    - name: alpine
      image: alpine
      volumeMounts:
        - mountPath: /opt
          name: data_vol
  volumes:
    - name: data_vol
      persistentVolumeClaim:
        claimName: pvc-data
```

mount

 NetApp 26 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Dynamic provisioning: Special case

StatefulSets with shared storage

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-trident
provisioner: csi.trident.netapp.io
parameters:
  <Additional Trident Parameters>
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-data
spec:
  accessMode:
    - ReadWriteOnce
  storageClassName: sc-trident
  resource:
    request:
      storage: 1Gi
```

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql-ss
  labels:
    type: db
spec:
  template:
    metadata:
      labels:
        type: db
    spec:
      containers:
        - name: mysql-ss
          image: mysql
          volumeMounts:
            - mountPath: /var/lib/mysql
              name: data-vol
        - name: data-vol
          persistedVolumeClaims:
            claimName: pvc-data
  replicas: 3
  selector:
    matchLabels:
      type: db
  serviceName: mysql-headless
```

By default, sharing the same data



Instructor Notes

Student Notes

Dynamic provisioning: Special case with separate PVCs

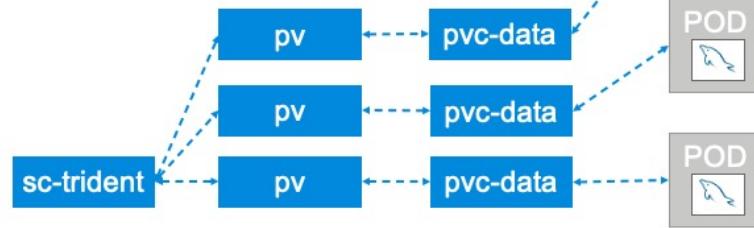
StatefulSets using volume claim templates

NOTE: If destroyed,
attaches the new
instance of the pod to
the existing PVC

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-data
spec:
  accessMode:
    - ReadWriteOnce
  storageClassName: sc-trident
  resource:
    request:
      storage: 1Gi
```

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mysql-ss
  labels:
    type: db
spec:
  template:
    metadata:
      labels:
        type: db
    spec:
      containers:
        - name: mysql-ss
          image: mysql
          volumeMounts:
            - mountPath: /var/lib/mysql
              name: data-vol
      volumes:
        - name: data-vol
          persistentVolumeClaims:
            claimName: pvc-data
  replicas: 3
  selector:
    matchLabels:
      type: db
  serviceName: mysql-headless
  volumeClaimTemplates:
```

To create a separate
PVC, designate the
Volume Claim Template



NetApp 28 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Astra Trident

The NetApp container storage provisioner

- Is a dynamic, automated storage provisioner
- Can abstract storage into pools of capabilities
 - Ability to differentiate storage
 - Ability for administrator to specify IOPS, compression, disk type, and so on
- Can map storage requests to storage pools
- Is open-source and fully supported:
 - <https://github.com/netapp/trident>
 - <https://docs.netapp.com/us-en/trident/index.html>

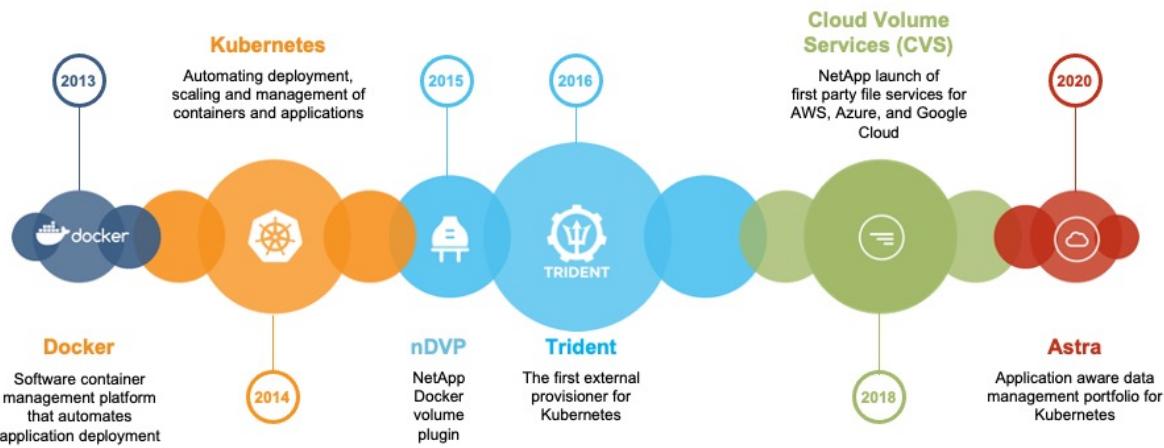


Instructor Notes

Student Notes

The NetApp contributions to the Kubernetes ecosystem

Astra Trident has been available since 2016: The first external vendor provisioner



NetApp 30 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Module summary

This module focused on enabling you to do the following:

- Introduce Kubernetes and the need for persistent storage
- Describe how containers use persistent storage
- Summarize the storage concepts of Kubernetes
- Explain how Astra Trident enables dynamic provisioning of Kubernetes persistent volumes

Instructor Notes

Student Notes



Complete an exercise

Module 1: Kubernetes storage overview

Working with Kubernetes storage volumes

- Task 1: Set up an emptyDir volume
- Task 2: Configure a hostPath volume
- Task 3: Deploy a pod with two containers that share storage
- Task 4: Configure an NFS server by using NetApp ONTAP software
- Task 5: Set up an NFS volume
- Task 6: Configure a PV and a PVC

Exercise duration:
60 minutes

 32 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

See the instructions in your Exercise Guide.



Share your experiences

Roundtable discussion

 NetApp 33 © 2022 NetApp, Inc. All rights reserved.

Working with Kubernetes storage volumes

- What is the difference between persistent volumes and persistent volume claims?
- Why would you create a StorageClass object?
- How would you ensure StatefulSet's pods are issued separate storage?

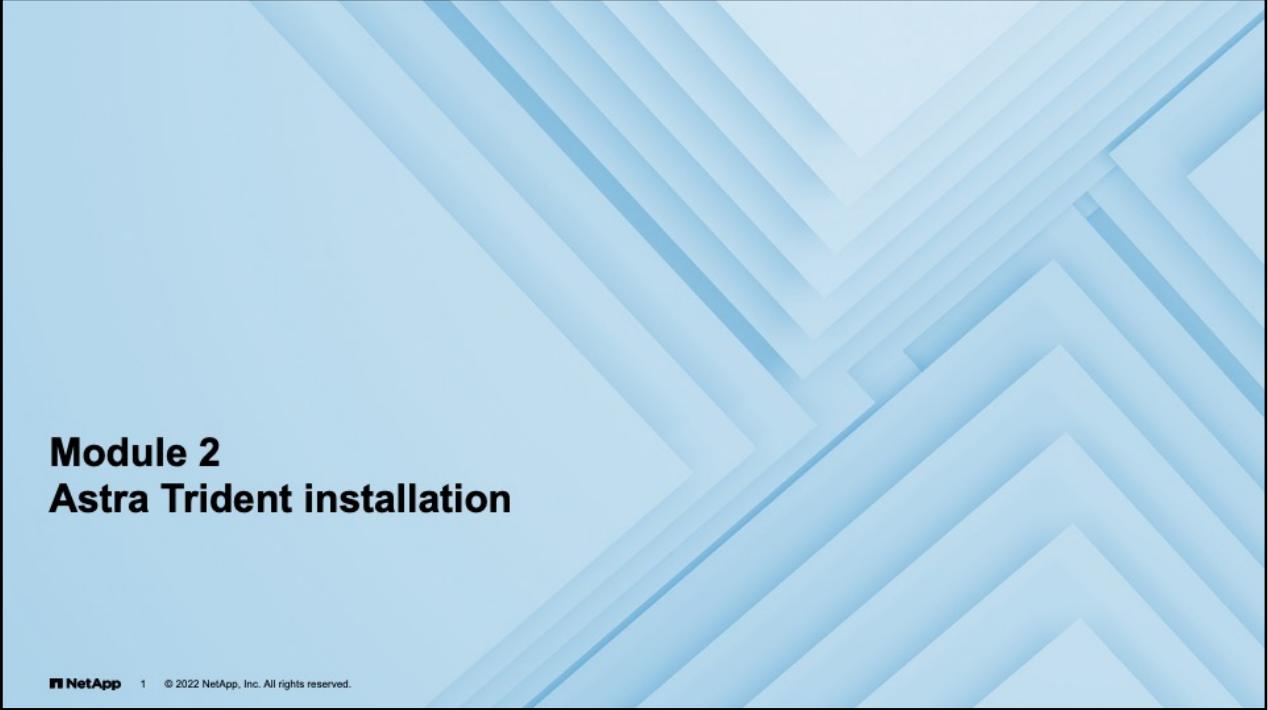
Instructor Notes

Display while students are completing the exercise(s)

Student Notes

Have a roundtable discussion with the class to answer these questions. Add any comments about experiences or “lessons learned” during the exercises that others might find helpful.

If you encounter an issue, notify your instructor immediately so that it can be resolved promptly.



Module 2

Astra Trident installation

 NetApp 1 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

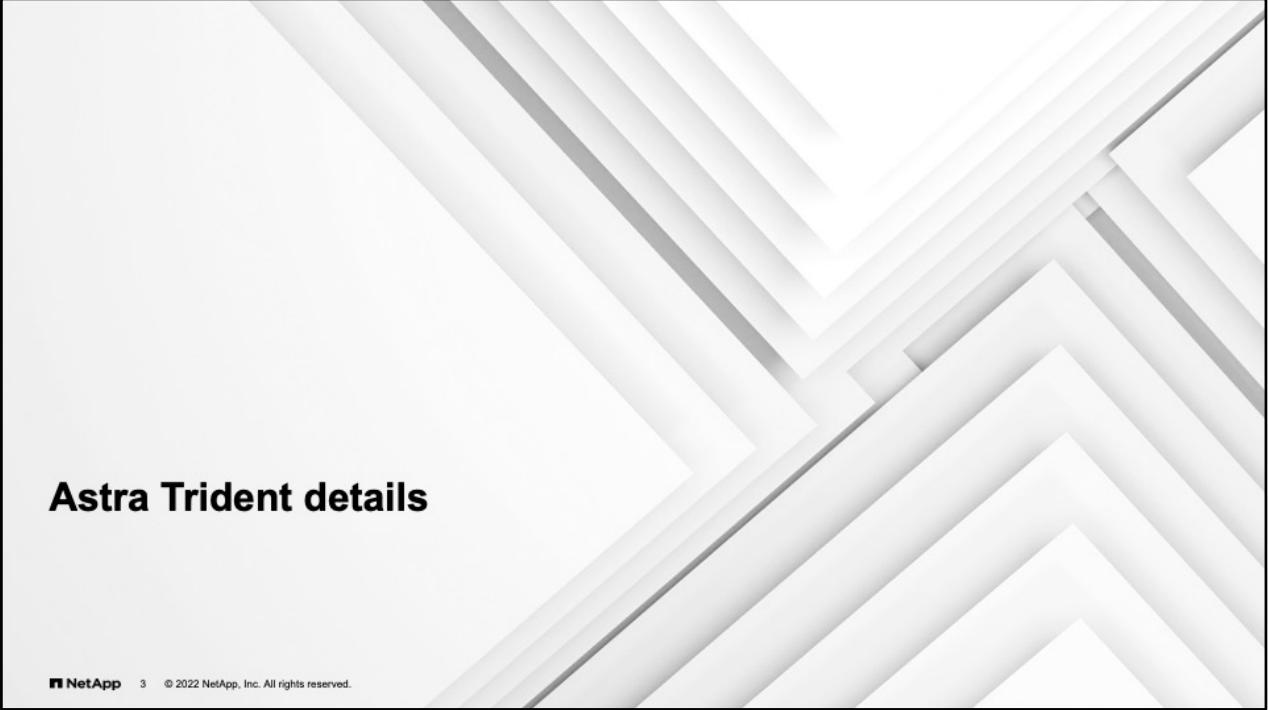
About this module

This module focuses on enabling you to do the following:

- Describe NetApp Astra Trident
- Explain how to install Astra Trident

Instructor Notes

Student Notes



Astra Trident details

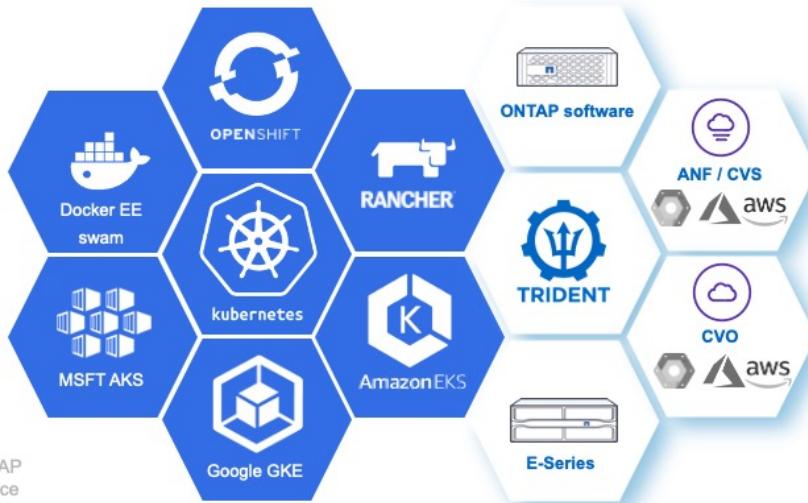
 NetApp 3 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Astra Trident

A cloud-native tool for NetApp storage



NetApp 4 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Typical software development workflow

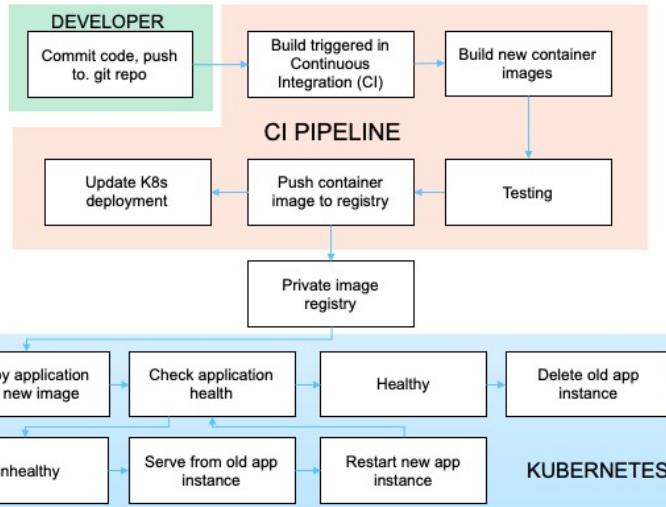
Two roles



- Developers
 - Focus on writing code
 - Work with Kubernetes manifests as little as needed to get test and deploy their code



- Administrators
 - Maintain the Kubernetes (K8s) clusters
 - Maintain storage clusters

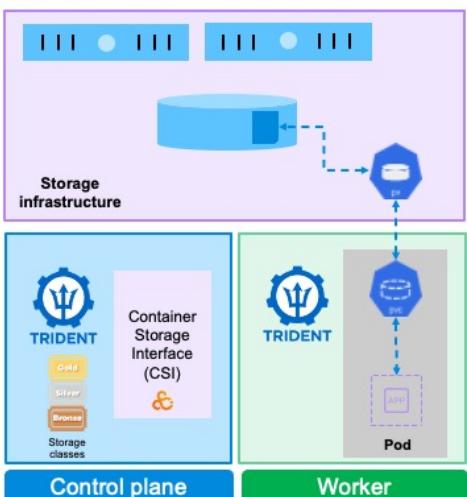


Instructor Notes

Student Notes

Astra Trident workflow

Creation of storage



Admin

1. Configures storage infrastructure
2. Installs Astra Trident in Kubernetes
3. Defines one or more back ends with virtual storage pool
4. Defines storage classes based on back ends



Developer

5. Creates a persistent volume claim (PVC) for a pod by using a storage class



TRIDENT

6. Detects creation of the PVC
7. Finds storage pool that satisfies the storage class
8. Creates a volume in an appropriate storage pool
9. Creates a persistent volume (PV) and binds the PVC with the PV
10. Mounts the volume with the container

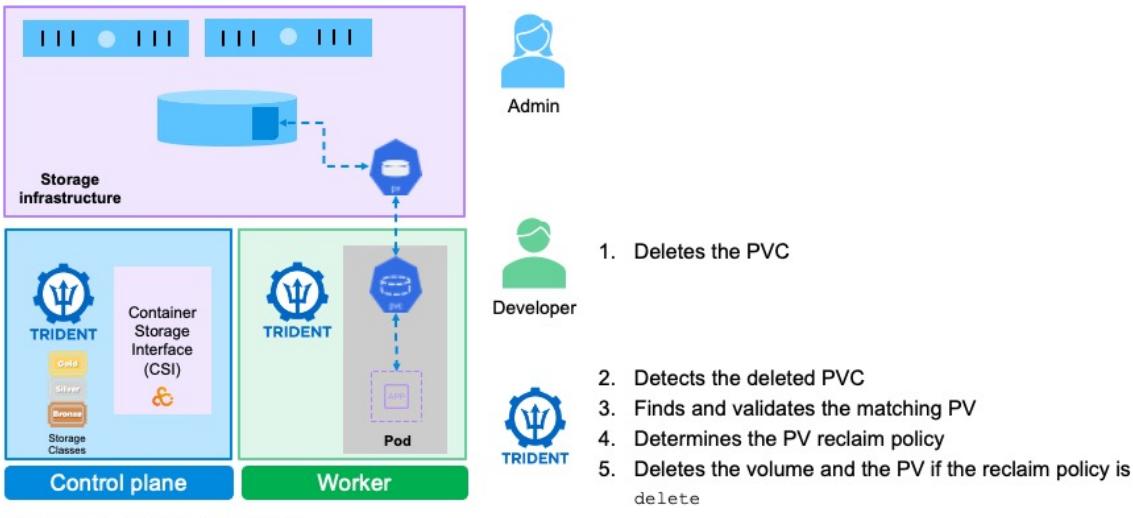
NetApp 6 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Astra Trident workflow

Deletion of storage



Instructor Notes

Student Notes

Support matrix

Broad compatibility

Operating system	Container platform	Storage back end
 CentOS	 Kubernetes 1.11-1.21.0 and later	 ONTAP software (FAS and AFF)
 Red Hat Enterprise Linux	 OpenShift 3.11, 4.2, 4.3, 4.4, 4.5, 4.6.8, 4.7 and later	 Element software (SolidFire)
 Ubuntu	 Docker Enterprise 2.1, 3.0, 3.1 and 3.2.11	 SANtricity software (E-Series, EF-Series)
 Red Hat Enterprise Linux CoreOS (RHCOS)	 Google Anthos 1.1, 1.2 and Anthos on-prem 1.15	 Azure NetApp Files
		 Cloud Volumes Service for AWS  Cloud Volumes Service for Google
		 Cloud Volumes ONTAP

 8 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

NOTE: Astra Trident should work with any distribution of Docker or Kubernetes that uses one of the supported versions as a base, such as Rancher and Tectonic.

Features

See <https://github.com/NetApp/trident/releases> for the latest

Feature	Astra Trident version	Kubernetes version	Feature gate required?
CSI Trident	19.07+	1.13+	Yes for 1.13
CSI Snapshots	20.01+	1.17+	No
PVC from CSI Snapshots	20.01+	1.17+	No
iSCSI PVC Resize	19.10+	1.16+	No
ONTAP bidirectional CHAP	20.04+	1.11+	No
Dynamic Export Policy	20.04+	1.13+	Requires CSI Astra Trident
Trident Operator	20.04+	1.14+	No
CSI Topology	20.10+	1.17	No
Auto-Worker Node Preparation (Beta)	20.10+	1.13+	Requires CSI Astra Trident

Instructor Notes

Student Notes

Astra Trident and ONTAP software

Protocols and access modes

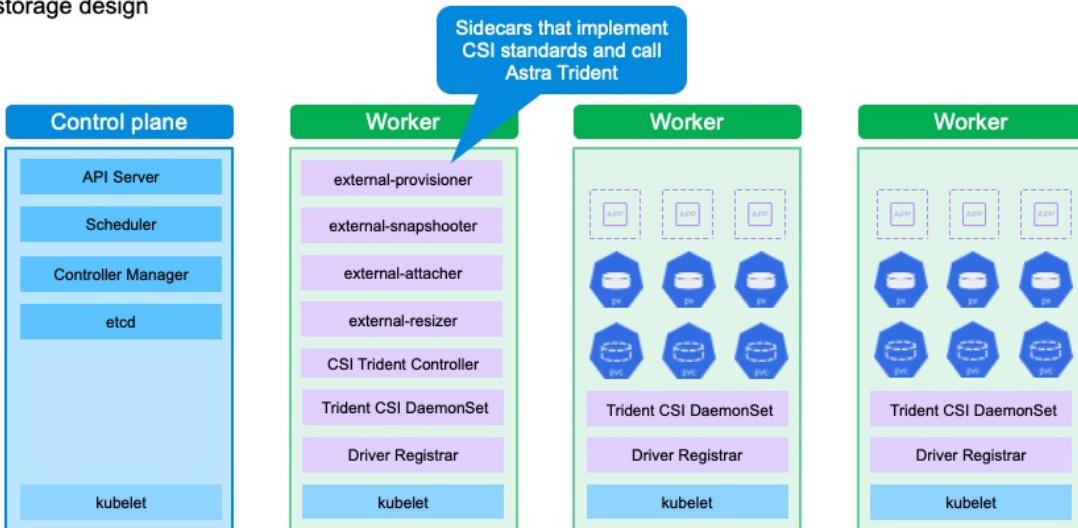
Driver	Protocol	Volume mode	Access mode	File systems supported
ontap-nas	NFS	Filesystem	RWO,RWX,ROX	" ", nfs
ontap-nas-economy	NFS	Filesystem	RWO,RWX,ROX	" ", nfs
ontap-nas-flexgroup	NFS	Filesystem	RWO,RWX,ROX	" ", nfs
ontap-san	iSCSI	Block	RWO,ROX,RWX	No filesystem, raw block device
ontap-san-economy	iSCSI	Block	RWO,ROX,RWX	No filesystem, raw block device
ontap-san	iSCSI	Filesystem	RWO,ROX	xfs, ext3, ext4
ontap-san-economy	iSCSI	Filesystem	RWO,ROX	xfs, ext3, ext4

Instructor Notes

Student Notes

Astra Trident storage architecture

CSI storage design



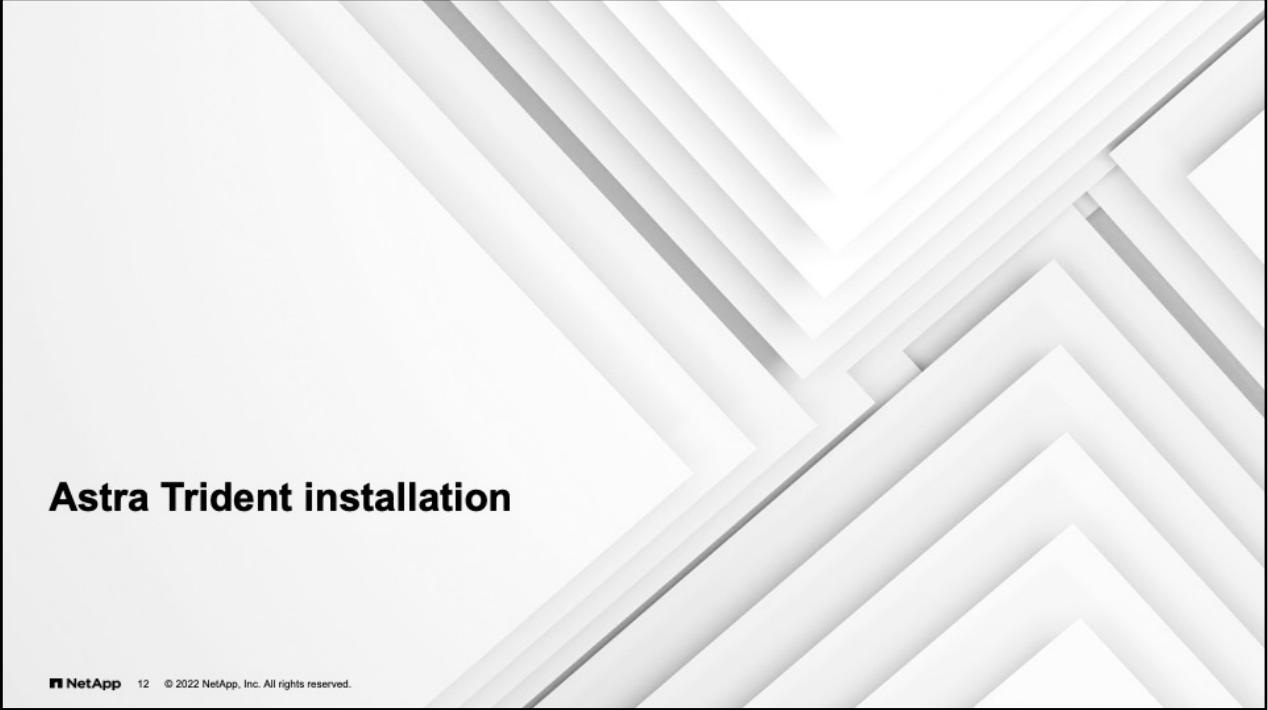
NetApp 11 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

The CSI storage standard that Astra Trident implements has various containers that are used to help to implement the standard. Some of these containers include the following:

- external-provisioner is a sidecar container that watches the Kubernetes API server for PersistentVolumeClaim objects. It requests CreateVolume to the CSI Astra Trident Controller to provision a new volume in the back-end storage. For more information, see <https://kubernetes-csi.github.io/docs/external-provisioner.html>.
- external-snapshotter is a sidecar container that watches the Kubernetes API server for VolumeSnapshot and VolumeSnapshotContent custom resources. It requests CreateSnapshot, DeleteSnapshot, and ListSnapshot to the CSI Astra Trident Controller to manage Snapshot copies in the back-end storage. For more information, see <https://kubernetes-csi.github.io/docs/external-snapshotter.html>.
- external-attacher is a sidecar container that watches the Kubernetes API server for VolumeAttachment objects and requests CSI Astra Trident Controller to run various volume operations. For more information, see <https://kubernetes-csi.github.io/docs/external-attacher.html>.
- external-resizer is a sidecar container that watches the Kubernetes API server for PersistentVolumeClaims object edits and requests CSI Astra Trident Controller to run ControllerExpandVolume operations in the storage back end. For more information, see <https://kubernetes-csi.github.io/docs/external-resizer.html>.



Astra Trident installation

 NetApp 12 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Astra Trident installation

Binary location

- Available at <https://github.com/NetApp/trident/releases>

- Executable and sample YAML files
- Source code

- Documented installation process at
<https://docs.netapp.com/us-en/trident/index.html>

- Methods to install:

- Use Binary (`tridentctl`)
- Use Operator

Introduced with Astra Trident 20.04
(approach used in the lab exercises)



Operator, Helm, and
GitOps methods
customizable

- Use a Helm chart

Introduced with Astra Trident 21.01

- Use GitOps

Introduced with Astra Trident 21.04

v22.01.0 Lastest

Changes since v21.10.0

IMPORTANT: If you are upgrading from any previous Trident release and use Azure NetApp Files, the `[location]` config parameter is now a mandatory, singleton field.

Fixes:

- Fixed issue where `azure-netapp-files` driver could be confused by multiple resources with the same name.
- ONTAP SAN IPv6 Data LIF's now work if specified with brackets.
- Kubernetes: Increase node registration backoff retry time for large clusters.
- Fixed issue where attempting to import an already imported volume returns EOF leaving PVC in pending state (issue #489).
- Fixed issue when Astra Trident performance slows down when > 32 snapshots are created on a SolidFire volume.
- Replaced SHA-1 with SHA-256 in SSL certificate creation.
- Fixed ANF driver to allow duplicate resource names and limit operations to a single location.

Enhancements:

- Added ability to limit `azure-netapp-files` driver to specific resource groups, NetApp accounts, capacity pools.
- Kubernetes: Added support for Kubernetes 1.23.
- Allow cross-region volumes in GCP driver (issue #633)
- Kubernetes: Add scheduling options for Trident pods when installed via Trident Operator or Helm (issue #657)
- Added support for `'unixPermissions'` option to ANF volumes. (issue #666)

Deprecations:

- Trident REST Interface can listen and serve only at 127.0.0.1 or `[-1]` addresses

Assets 3

- `trident-installer-22.01.0.tar.gz` 55.5 MB
- `Source code (.zip)`
- `Source code (.tar.gz)`

Instructor Notes

Student Notes

Astra Trident installation

Binary method overview



1. Create a namespace for Trident (`trident`)
2. Prepare worker nodes for NFS / iSCSI
3. Install trident: `./tridentctl install -n trident`
4. Verify `trident-csi-<generated id>` pod is running
5. Verify `./tridentctl -n trident version`

Instructor Notes

Student Notes

Astra Trident installation

Operator method overview



1. Create the Custom Resource Definitions
2. Apply `deploy/bundle.yaml` from the `tar.gz` file
 - Creates namespace
 - Creates a service account and security bindings
 - Installs the operator
 - Sets up a pod security policy
3. Create a `TridentOrchestrator` custom resource
Optionally, limit the nodes to deploy Astra Trident by using taints and tolerations or nodeSelectors
4. Prepare worker nodes for NFS / iSCSI
5. Verify that `trident-csi-<generated id>` pod is running

Instructor Notes

Student Notes

Astra Trident installation

Helm method overview



1. Add the Astra Trident helm repo:
`helm repo add netapp-trident \\\nhttps://netapp.github.io/trident-helm-chart`
2. Execute the helm install:
`helm install trident -n trident \\\nnetapp-trident/trident-operator --version 21.7.0`
3. Alternatively, download the helm chart and all
artifacts in a `tgz` file and install
4. Prepare worker nodes for NFS / iSCSI
5. Verify that `trident-csi-<generated id>` pod is
running

Instructor Notes

Student Notes

The Trident Helm chart can be installed directly from its public repository if the Kubernetes cluster has access to internet. Otherwise, the chart is also included in the Trident package that you can download from the NetApp GitHub page.

Astra Trident installation

GitOps method overview



See notes for details

1. Implement a continuous deployment (CD) application such as ArgoCD
2. Configure an ArgoCD application by using one of three methods:
 - Within the ArgoCD UI
 - Using `kubectl`
 - Using the "Ghost" method

Instructor Notes

Student Notes

With Trident 21.04, the Kubernetes administrator does not need to manage Astra Trident outside of Kubernetes. Now that Trident back ends can be created with `kubectl`, the whole Astra Trident lifecycle can be managed through continuous deployment (CD) tools, such as ArgoCD. For more information, see https://github.com/YvosOnTheHub/LabNetApp/tree/master/Kubernetes_v4/Scenarios/Scenario18.

Module summary

This module focused on enabling you to do the following:

- Describe Astra Trident
- Explain how to install Astra Trident

Instructor Notes

Student Notes



Complete an exercise

Module 2: Astra Trident installation

Installing Astra Trident

- Task 1: Download and set up the Astra Trident operator
- Task 2: Deploy instances of Astra Trident
- Task 3: Set up the `tridentctl` tool
- Task 4: Prepare worker nodes

Exercise duration:
60 minutes

 19 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

See the instructions in your Exercise Guide.



Share your experiences

Roundtable discussion

Installing Astra Trident

- Why would you use Astra Trident?
- Where does Astra Trident store its configuration data?
- What are methods to install Astra Trident?

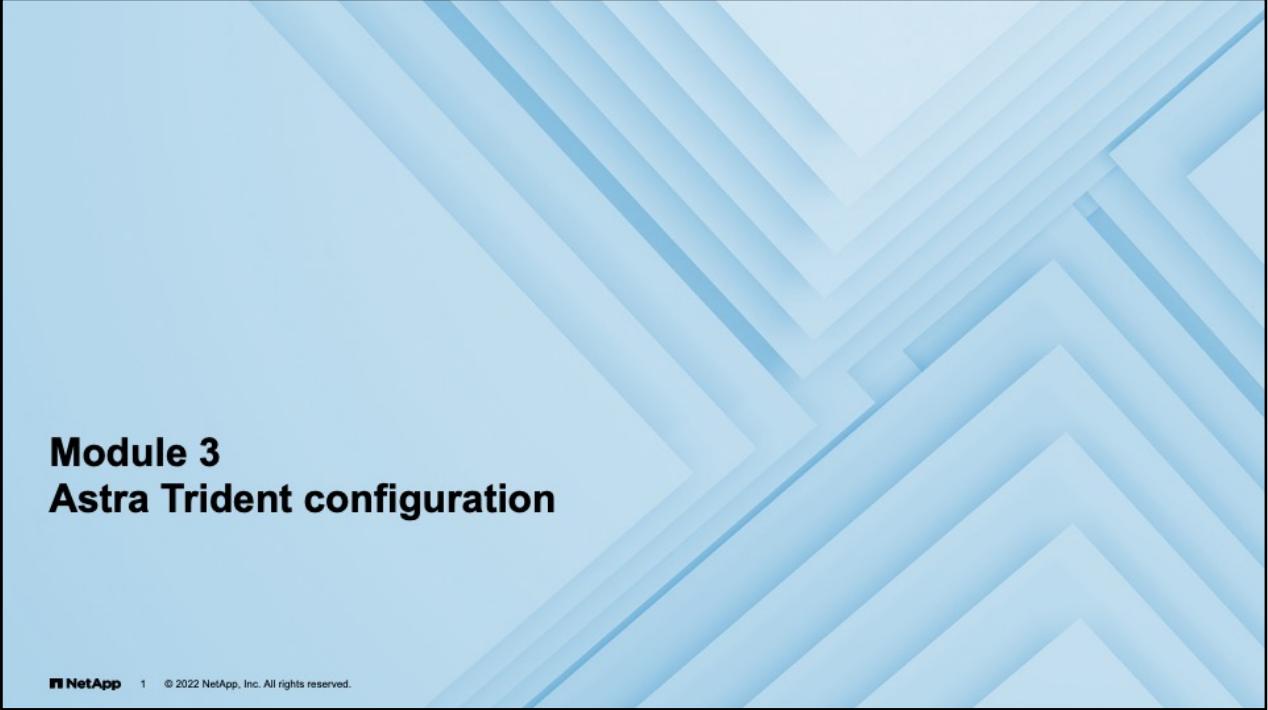
Instructor Notes

Display while students are completing the exercise(s)

Student Notes

Have a roundtable discussion with the class to answer these questions. Add any comments about experiences or “lessons learned” during the exercises that others might find helpful.

If you encounter an issue, notify your instructor immediately so that it can be resolved promptly.



Module 3

Astra Trident configuration

 1 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

About this module

This module focuses on enabling you to do the following:

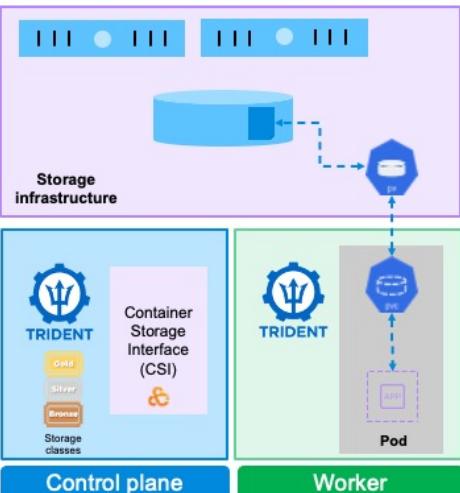
- Configure back ends
- Manage storage classes
- Create a persistent volume (PV) by using NetApp Astra Trident

Instructor Notes

Student Notes

Astra Trident terminology

Four basic concepts



Back ends

- Define the relationship between Astra Trident and a storage system
- Have parameters on how to communicate with the back-end storage
- Are either a YAML or JSON document stored in etcd



Virtual storage pools

- Are abstract layers that are defined in the back end
- Can be an aggregate or “local tier” in a NetApp ONTAP cluster (for example)



Storage classes

Use a particular back end



Persistent volume claims (PVCs)

- Use a particular storage class
- Create a persistent volume

Instructor Notes

Student Notes

Back ends: Back ends represent the storage providers on top of which Astra Trident provisions volumes. A single Astra Trident instance can manage any number of back ends.

Storage pools: Storage pools represent the distinct locations available for provisioning on each back end. For NetApp ONTAP software, these storage pools correspond to aggregates in storage VMs (storage virtual machines, also known as SVMs). For NetApp SolidFire, these storage pools correspond to administrator-specified quality-of-service (QoS) bands. For NetApp Cloud Volumes Service, these storage pools correspond to cloud provider regions. Each storage pool has a set of distinct storage attributes, which define its performance characteristics and data protection characteristics.

Configure back ends

Instructor Notes

Student Notes

Back ends

Overview

- Define the relationship between Astra Trident and a storage system
- Can be created by using one of the following:
 - A JSON and the `tridentctl` tool
 - A YAML definition file and the `kubectl` tool
- Are configured for various storage platforms, for example:
 - Azure NetApp Files
 - NetApp Cloud Volumes Service for AWS
 - NetApp Cloud Volumes Services for Google Cloud
 - Amazon FSx for NetApp ONTAP
 - NetApp Element software
 - NetApp ONTAP software or NetApp Cloud Volumes ONTAP
- Use only one storage driver at a time:
 - `ontap-nas`
 - `ontap-nas-economy`
 - `ontap-nas-flexgroup`
 - `ontap-san`
 - `ontap-san-economy`
 - `ontap-san`
 - `ontap-san-economy`
 - `azure-netapp-files`
 - `aws-cvs`
 - `gcp-cvs`
 - `solidfire-san`

Instructor Notes

Student Notes

Back-end creation

NFS example

NFS JSON example

```
{  
    "version": 1,  
    "storageDriverName": "ontap-nas",  
    "backendName": "nas-backend",  
    "managementLIF": "192.168.0.101",  
    "dataLIF": "192.168.0.31",  
    "svm": "svm0",  
    "username": "admin",  
    "password": "Netapp1!"  
}
```

NFS YAML example

```
apiVersion: trident.netapp.io/v1  
kind: TridentBackendConfig  
metadata:  
  name: nas-backend  
spec:  
  version: 1  
  backendName: ontap-nas-backend  
  storageDriverName: ontap-nas  
  managementLIF: 192.168.0.40  
  dataLIF: 192.168.0.41  
  svm: svm1  
  credentials:  
    name: nas-backend-secret  
  
    apiVersion: v1  
    kind: Secret  
    metadata:  
      name: nas-backend-secret  
    type: kubernetes.io/basic-auth  
    stringData:  
      username: vsadmin  
      password: Netapp1!
```

 NetApp 6 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Back-end creation

SAN example

SAN JSON example

```
{  
    "version": 1,  
    "storageDriverName": "ontap-san",  
    "managementLIF": "10.0.0.1",  
    "dataLIF": "10.0.0.2",  
    "svm": "trident_svm",  
    "username": "admin",  
    "password": "password",  
    "labels": {"k8scluster": "dev2", "backend": "dev2-sanbackend"},  
    "storagePrefix": "alternate-trident",  
    "igroupName": "custom",  
    "debugTraceFlags": {"api":false, "method":true},  
    "defaults": {  
        "spaceReserve": "volume",  
        "qosPolicy": "standard",  
        "spaceAllocation": "false",  
        "snapshotPolicy": "default",  
        "snapshotReserve": "10"  
    }  
}
```

 NetApp 7 © 2022 NetApp, Inc. All rights reserved.

SAN YAML example

```
apiVersion: trident.netapp.io/v1  
kind: TridentBackendConfig  
metadata:  
    name: san-backend  
spec:  
    version: 1  
    backendName: ontap-san-backend  
    storageDriverName: ontap-san  
    managementLIF: 192.168.0.40  
    dataLIF: 192.168.0.41  
    svm: svm1  
    credentials:  
        name: san-backend-secret
```

Instructor Notes

Student Notes

Back-end management

Examples

- Tridentctl approach

- Create:

```
tridentctl create backend -f <file> -n trident
```

- Delete:

```
tridentctl delete backend -f <file> -n trident
```

- View:

```
tridentctl get backend <backend> -o json -n trident
```

- Update:

```
tridentctl update backend <backend> -f <file> -n trident
```

- Identify the storage class that uses a back end:

```
tridentctl get backend -o json | jq '[.items[] | {backend: .name, storageClasses: [.storage[].storageClasses]|unique}]'
```

- Kubectl approach

- Create:

```
kubectl -n trident create -f <file>
```

- Delete:

```
kubectl -n trident tbc <backend>
```

- View:

```
kubectl -n trident get tbc [backend]  
kubectl -n trident describe tbc <backend>
```

- Update:

```
kubectl -n trident apply -f <file>
```

```
kubectl -n trident edit tbc <backend>
```

Requires JQuery tool

Instructor Notes

Student Notes



Manage storage classes

 9 © 2022 NetApp, Inc. All rights reserved.

Instructor Notes

Student Notes

Storage classes

Overview

Matched to any storage pool that meets the specific criteria:

- ```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
 name: performance
 provisioner: csi.trident.netapp.io
parameters:
 media: "ssd"
 provisioningType: "thin"
 snapshots: "true"

• Define the back end directly:
 parameters:
 backendType: ontap-nas

• Pick storage pools directly:
 parameters:
 StoragePool: "ontapnas_10.1.1.1:aggr1,aggr4,aggr10"
```

Overrides all other parameters

### Kubectl approach

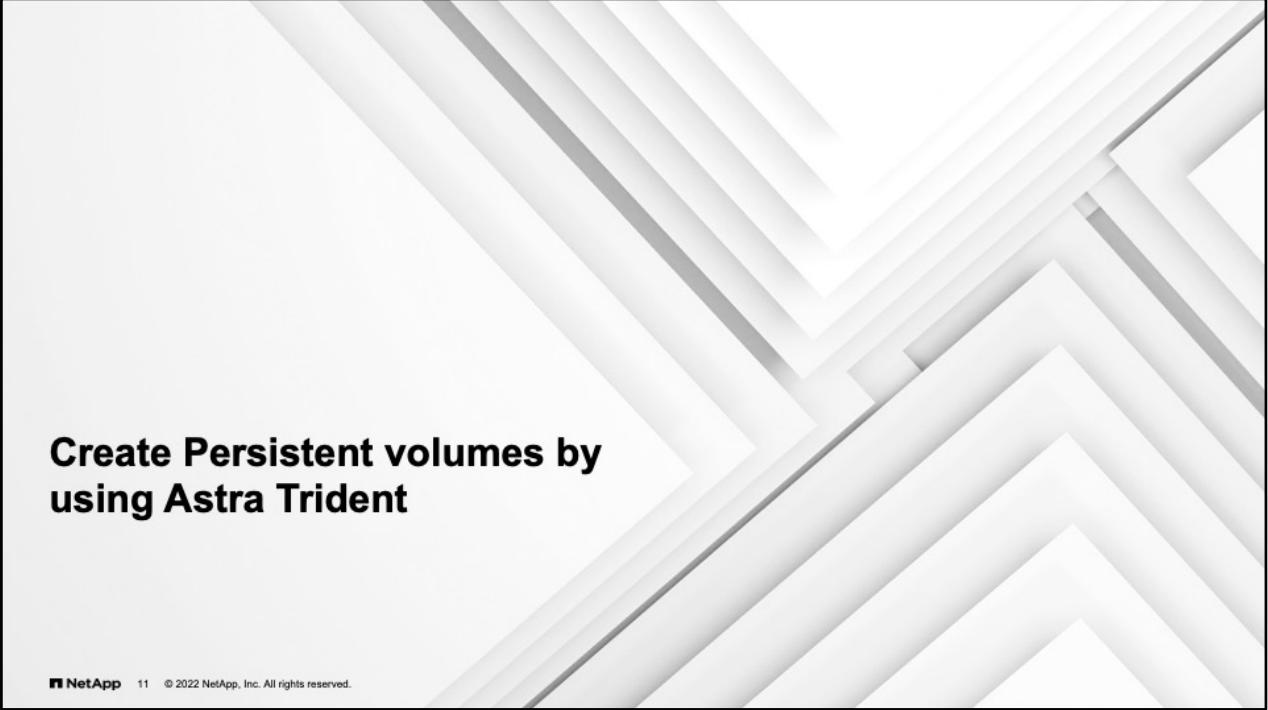
- **Create:**  
kubectl -n trident create -f <file>
- **Delete:**  
kubectl -n trident sc <storageclass>
- **View:**  
kubectl -n trident get sc [storageclass]  
kubectl -n trident describe sc <storageclass>
- **Update:**  
kubectl -n trident apply -f <file>  
kubectl -n trident edit sc <storageclass>
- **Change the default storage class:**  
kubectl patch storageclass <storage-class-name> -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
- **Remove the default storage class:**  
kubectl patch storageclass <storage-class-name> -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"false"}}}'

## Instructor Notes

### Student Notes

As of Kubernetes 1.16, you can specify a default storage class for all persistent volume classes. To make this selection, add the annotation `storageclass.kubernetes.io/is-default-class` to true.

Astra Trident storage pools also offer the capability to select volume destinations based on properties such as back-end type, media type, provisioning type (thin versus thick), encryption, and so on. The supported choice capabilities vary according to the storage driver that is being used.



## **Create Persistent volumes by using Astra Trident**

 NetApp 11 © 2022 NetApp, Inc. All rights reserved.

### **Instructor Notes**

### **Student Notes**

## A pod with storage

Astra Trident creates the storage for the pod

### Create a PVC

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

Specify the storage class used by Astra Trident

### A pod that uses the PVC

```
apiVersion: v1
kind: Pod
metadata:
 name: nfs-pod
spec:
 volumes:
 - name: nfs-storage
 persistentVolumeClaim:
 claimName: nfs-basic
 containers:
 - name: nfs-container
 image: nginx
 volumeMounts:
 - mountPath: "/usr/share/nginx/html"
 name: nfs-storage
```

## Instructor Notes

### Student Notes

When Astra Trident provisions a persistent volume, it selects which back end to use based on the associated storage class's "backendType" parameter, which contains the name of a storage driver. For example, storage classes that specify the "ontap\_nas" driver select a back end for ONTAP NFS storage. If you have multiple back ends of the desired type defined in your environment, Astra Trident randomly selects one on which to deploy the volume.

## Module summary

This module focused on enabling you to do the following:

- Configure back ends
- Manage storage classes
- Create a PV by using Astra Trident

### Instructor Notes

### Student Notes



## Complete an exercise

Module 3: Astra Trident configuration

### Working with Astra Trident

- Task 1: Create a NAS back end
- Task 2: Create a storage class for a NAS back end
- Task 3: Provision a persistent volume claim with a NAS back end
- Task 4: Mount the volumes in a pod
- Task 5: Perform back-end management by using the `tridentctl` tool
- Task 6: Perform back-end management with the `kubectl` tool

Exercise duration:  
**60 minutes**

 14 © 2022 NetApp, Inc. All rights reserved.

### Instructor Notes

### Student Notes

See the instructions in your Exercise Guide.



## Share your experiences

Roundtable discussion

### Working with Astra Trident

- Why do you need a back end?
- How do back ends connect to storage class objects?

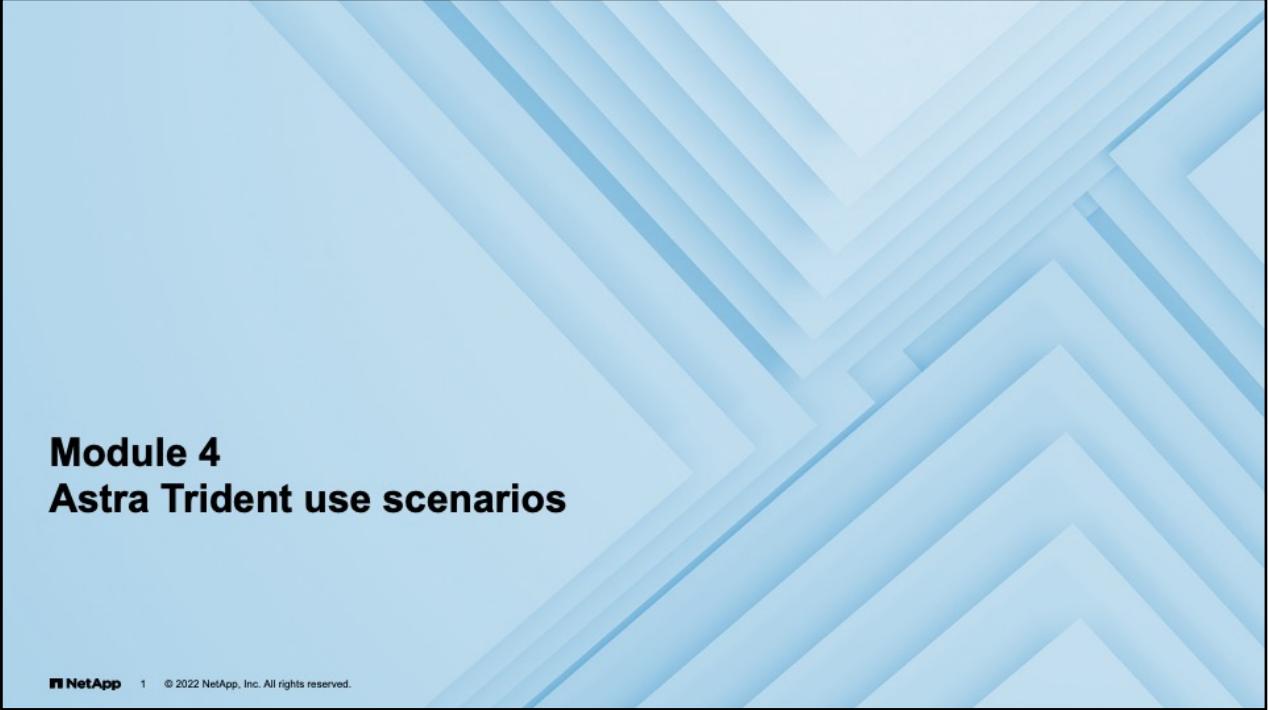
### Instructor Notes

Display while students are completing the exercise(s)

### Student Notes

Have a roundtable discussion with the class to answer these questions. Add any comments about experiences or “lessons learned” during the exercises that others might find helpful.

If you encounter an issue, notify your instructor immediately so that it can be resolved promptly.



## **Module 4**

### **Astra Trident use scenarios**

 NetApp 1 © 2022 NetApp, Inc. All rights reserved.

**Instructor Notes**

**Student Notes**

## About this module

This module focuses on enabling you to do the following:

- Manage Snapshot copies with NetApp Astra Trident
- Expand volumes by using Astra Trident
- Import volumes by using Astra Trident
- Deploy Astra Trident storage in multiple zones

### Instructor Notes

### Student Notes



## **Manage Snapshot copies with Astra Trident**

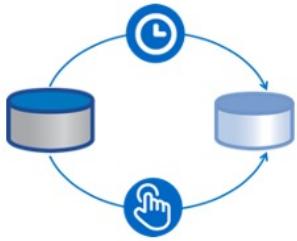
 NetApp 3 © 2022 NetApp, Inc. All rights reserved.

**Instructor Notes**

**Student Notes**

## CSI Snapshots

Available in Astra Trident 20.01 or later



- Available from the community
- A point in time copy of data and the state of a volume
- A great framework to use as part of a backup job
- Scheduled or on-demand execution
- Standard method with Kubernetes v1.20 and later
- Container Storage Interface (CSI)  
S snapshots not *consistent* snapshots

## Instructor Notes

### Student Notes

For more information about the Container Storage Interface (CSI) specification, see <https://github.com/container-storage-interface/spec/blob/master/spec.md>. *Warning:* CSI Snapshots are not *consistent* snapshots. Application integration must be achieved with third-party solutions, such as NetApp Astra Control.

## Resource definitions for Snapshots

Installed by using <https://github.com/kubernetes-csi/external-snapshotter>

- CRD: VolumeSnapshotClass
  - Defines the storage and storage parameters that are used in volume snapshots
  - Sets a deletion policy:
    - Delete: when deleting the volumesnapshot, the underlying data is also deleted
    - Retain: when deleting the volumesnapshot, the underlying data is not deleted
  - Is similar to storageclass
- CRD: VolumeSnapshot
  - Defines a request to create a snapshot of a volume
  - Is used by Astra Trident to create a NetApp Snapshot copy
  - Is associated with a single volumesnapshotclass
- CRD: VolumeSnapshotContents
  - Represents a snapshot taken from an already provisioned volume
  - Contains details about the unique snapshot
  - Has snapshotHandle attribute with the name of the persistent volume (PV) used

CRD: Custom Resource Definition

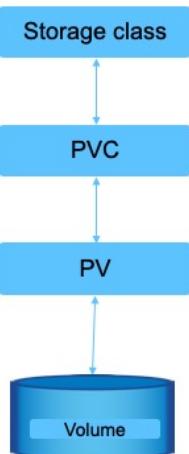
 NetApp 5 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

## Student Notes

## CSI Snapshot workflow

Astra Trident can create storage



PVC: Persistent volume claim

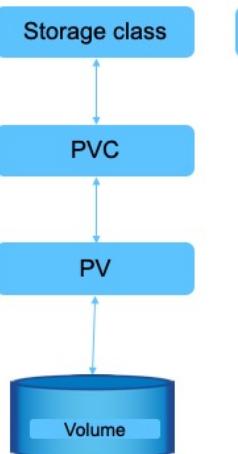


## Instructor Notes

## Student Notes

## CSI Snapshot workflow

Volume snapshot class creation



Volume snapshot class

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
 name: csi-snapclass
spec:
 driver: csi.trident.netapp.io
 deletionPolicy: Delete
```

Can be  
"Delete" or  
"Retain"

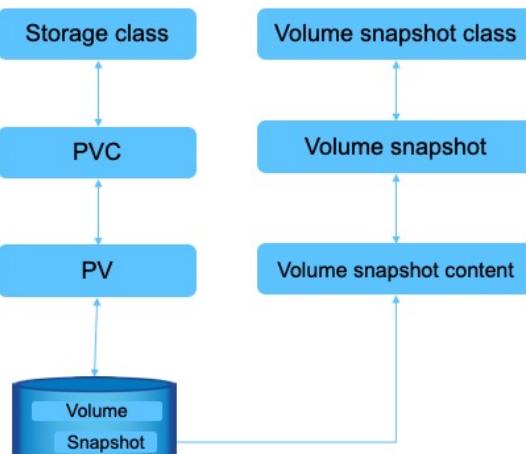
NetApp 7 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

## Student Notes

## CSI Snapshot workflow

Volume snapshot creation



```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
 name: volume-snapshot
spec:
 volumeSnapshotClassName: csi-snapclass
 source:
 persistentVolumeClaimName: pvc
```

Supported with the following drivers:

- ontap-nas
- ontap-san and ontap-san-economy
- solidfire-san
- aws-cvs
- gcp-cvs
- azure-netapp-files

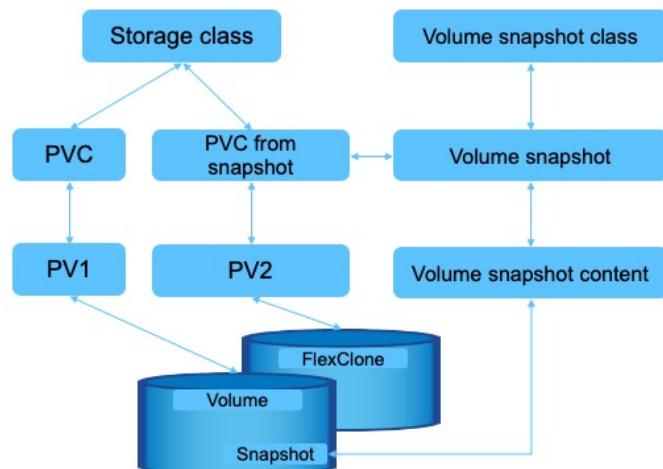
NetApp 8 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

## Student Notes

## CSI Snapshot workflow

Volume from snapshot creation



```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: pvc-from-snap
spec:
 accessModes:
 - ReadWriteMany
 resources:
 requests:
 storage: 1Gi
 storageClassName: sc-nas
 dataSource:
 name: volume-snapshot
 kind: VolumeSnapshot
 apiGroup: snapshot.storage.k8s.io
```

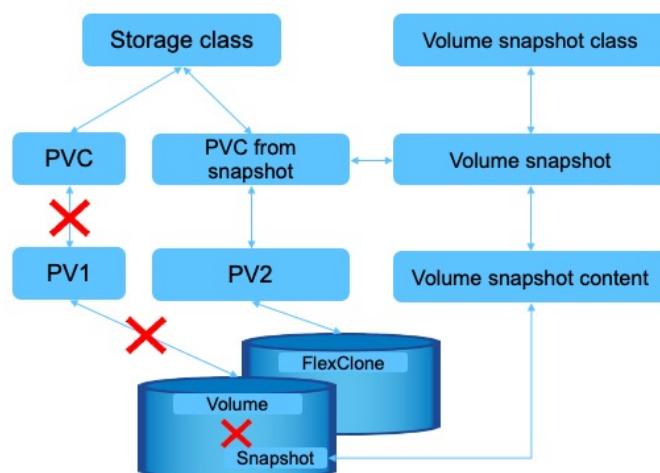
NetApp 9 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

## Student Notes

## CSI Snapshot workflow

Restore from a volume snapshot



```
apiVersion: v1
kind: Pod
metadata:
 name: pod
spec:
 volumes:
 - name: storage
 persistentVolumeClaim:
 claimName: pvc-from-snap
 containers:
 - name: container
 image: nginx
 resources:
 requests:
 memory: "64Mi"
 cpu: "250m"
 limits:
 memory: "128Mi"
 cpu: "500m"
 ports:
 - containerPort: 80
 name: "http-server"
 volumeMounts:
 - mountPath: "/usr/share/nginx/html"
 name: storage
```

To reassociate an existing pod to the new restored snapshot PVC

NetApp 10 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

## Student Notes

## Effect of SnapReserve

Using Astra Trident 21.07 or later

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
...
defaults:
 snapshotDir: 'true'
 snapshotPolicy: default
 snapshotReserve: '40'

apiVersion: v1
kind: PersistentVolumeClaim
...
spec:
 accessModes:
 - ReadWriteMany
 resources:
 requests:
 storage: 10Gi
...

```

Total FlexVol volume size  
16GB

10GB

Snapshot reserve size  
Requested PVC size

# kubectl exec pod -- df -h /data

| Filesystem               | Size | Used   | Available | Use% | Mounted on |
|--------------------------|------|--------|-----------|------|------------|
| 192.168.0.31:/nfs_pvc... | 1.0G | 192.0K | 11.0G     | 0%   | /data      |

 NetApp 11 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

### Student Notes

Astra Trident 21.07 sizes the available size of the FlexVol volume according to the PVC size. If a snapshot reserve is set, snapshot reserve is added to the available size of the FlexVol volume. The pod then sees the size requested in the PVC.



## **Expand volumes by using Astra Trident**

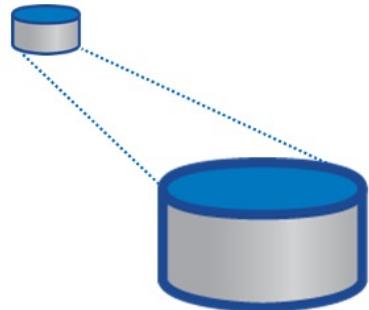
 NetApp 12 © 2022 NetApp, Inc. All rights reserved.

### **Instructor Notes**

### **Student Notes**

### **Volume resize**

Available in Astra Trident 20.01 or later



- Can expand volume size as needs grow
- Has no need for administrator intervention
- Supports NFS and iSCSI

## **Instructor Notes**

## **Student Notes**

## Resizing a volume

Using Astra Trident

To resize an existing PV:

- Change to the requested size in the PVC
- Astra Trident behavior:
  - If the PV is attached to a pod, expands the volume on the storage back end, rescans the device, and resizes the filesystem
  - When attempting to resize an unattached PV, expands the volume on only the storage back end

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: pvc
spec:
 accessModes:
 - ReadWriteMany
 resources:
 requests:
 storage: 2Gi
 storageClassName: nfs-basic
```

To change the PVC requested size

## Instructor Notes

## Student Notes



## **Import volumes by using Astra Trident**

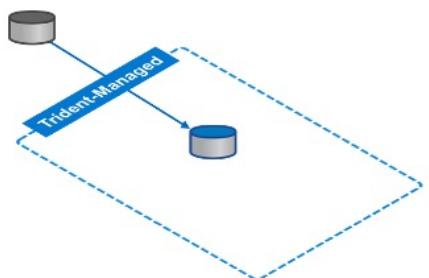
 NetApp 15 © 2022 NetApp, Inc. All rights reserved.

### **Instructor Notes**

### **Student Notes**

## Volume import steps

Available in Astra Trident 20.01 or later



- Use cases include the following:
  - Inter-cluster data migration
  - Monolithic application migration to Kubernetes
  - Disaster recovery
  - Supports NFS and iSCSI

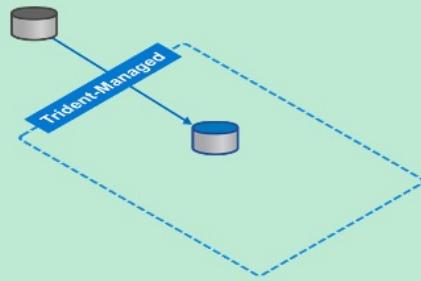
## Instructor Notes

## Student Notes

## Volume import steps

Using Astra Trident

1. Verify the availability of a volume not managed by Astra Trident
2. Create a PVC that is appropriate for the volume that you want to import
3. Select a back end to manage the volume
4. Use tridentctl to import the volume:  
`./tridentctl import volume [backend] [volume name] -f [YAML file of PVC] -n [namespace]`
5. Verify that the volume was imported and now appears as a PVC



## Instructor Notes

## Student Notes



## **Manage multiple-zone storage by using Astra Trident**

 NetApp 18 © 2022 NetApp, Inc. All rights reserved.

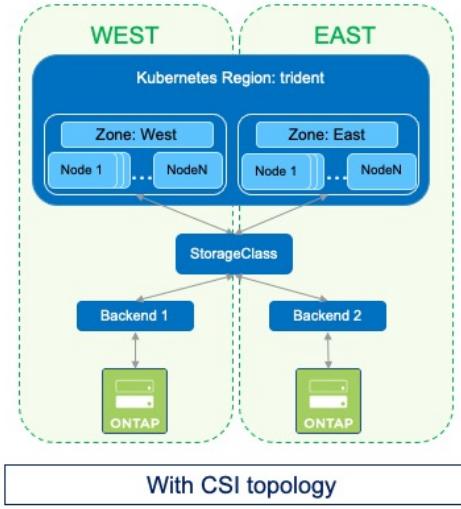
### **Instructor Notes**

### **Student Notes**

## CSI topologies

A community-adopted standard

- Are used to support different storage across different segments
  - Racks
  - Regions
  - Zones (or availability zones)
  - Domains
  - Functionality (dev versus test versus prod)
- Are used to provide storage available to the different segments
- Ensure that the volume is created in the same segment as the pod
- Can reduce the number of storage classes needed



With CSI topology

NetApp 19 © 2022 NetApp, Inc. All rights reserved.

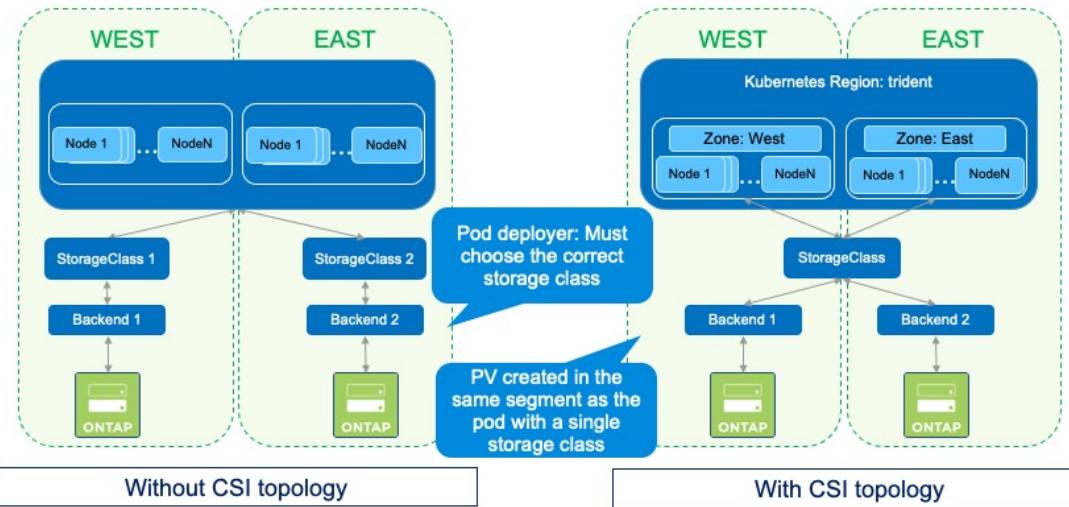
## Instructor Notes

### Student Notes

For more information about Kubernetes CSI topologies, see <https://kubernetes-csi.github.io/docs/topology.html>.

## CSI topologies simplify administration

No "geographic" specific storage classes with topologies



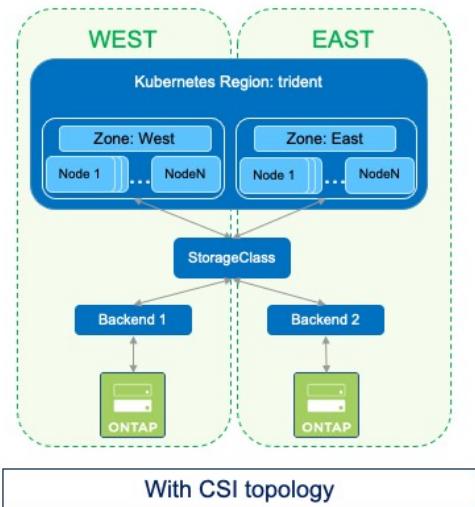
## Instructor Notes

### Student Notes

## CSI topologies steps

Requires Kubernetes 1.17 or later and Astra Trident 20.10 or later

1. Label the Kubernetes nodes before installing Astra Trident
  - topology.kubernetes.io/region
  - topology.kubernetes.io/zone
2. Kubernetes storage classes optional parameters:
  - allowedTopologies
  - volumeBindingMode: WaitForFirstConsumer
3. Specify Astra Trident back-end parameter  
supportedTopologies
4. Define a pod by using the storage class created in Step 3



NetApp 21 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

## Student Notes

## Module summary

This module focused on enabling you to do the following:

- Manage Snapshot copies with Astra Trident
- Expand volumes by using Astra Trident
- Import volumes by using Astra Trident
- Deploy Astra Trident storage in multiple zones

### Instructor Notes

### Student Notes



## Complete an exercise

Module 4: Astra Trident use scenarios

### Working with storage managed by Astra Trident

- Task 1: Managing Snapshot copies
- Task 2: Expanding volumes
- Task 3: Importing volumes
- Task 4: Deploying Astra Trident storage in multiple zones

Exercise duration:  
**60 minutes**

 NetApp 23 © 2022 NetApp, Inc. All rights reserved.

### Instructor Notes

### Student Notes

See the instructions in your Exercise Guide.



## Share your experiences

Roundtable discussion

### Working with storage managed by Astra Trident

- How is snapshot reserve accounted for by using Astra Trident?
- What is a common use case for importing a volume by using Astra Trident?
- Why would you use CSI topologies?

### Instructor Notes

Display while students are completing the exercise(s)

### Student Notes

Have a roundtable discussion with the class to answer these questions. Add any comments about experiences or “lessons learned” during the exercises that others might find helpful.

If you encounter an issue, notify your instructor immediately so that it can be resolved promptly.



## **Appendix 1**

### **Kubernetes-related certifications**

 NetApp 1 © 2022 NetApp, Inc. All rights reserved.

**Instructor Notes**

**Student Notes**

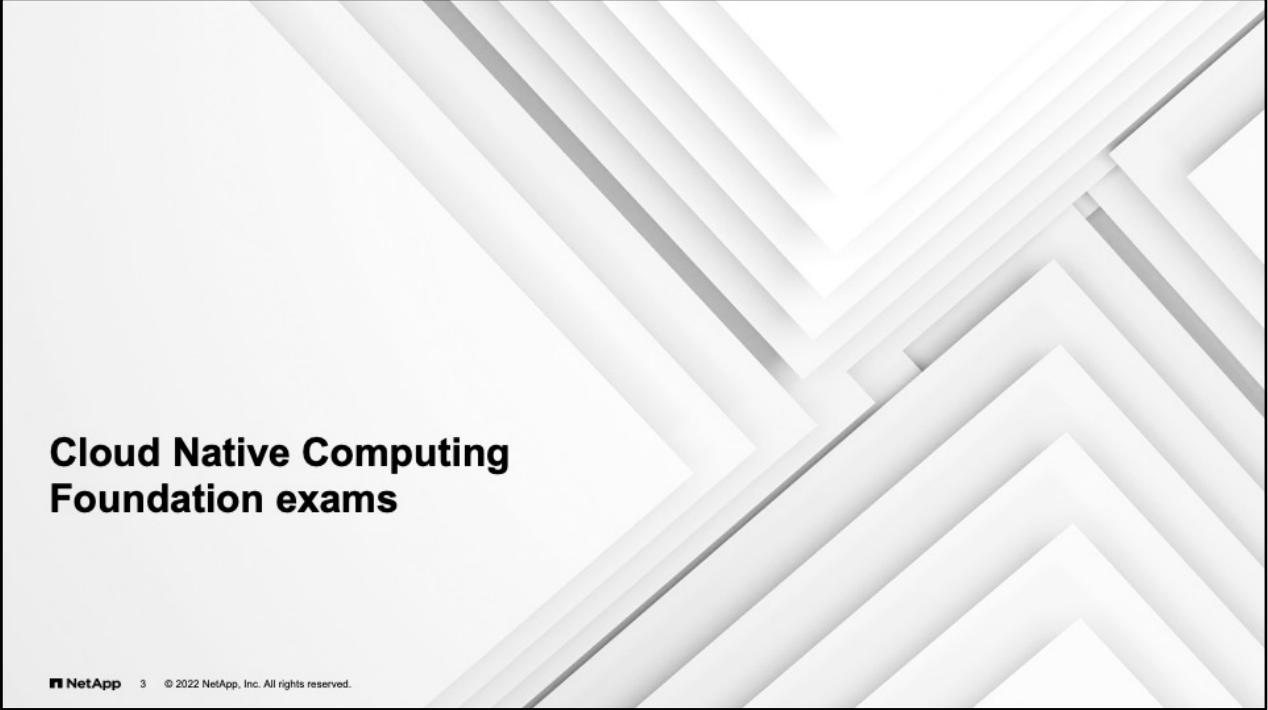
## About this module

This module focuses on enabling you to do the following:

- Describe Cloud Native Computing Foundation exams
- Describe NetApp Kubernetes-related exams

### Instructor Notes

### Student Notes



## **Cloud Native Computing Foundation exams**

 NetApp 3 © 2022 NetApp, Inc. All rights reserved.

**Instructor Notes**

**Student Notes**

**Cloud Native Computing Foundation (CNCF) and the Linux Foundation**

General information—subject to change



## Instructor Notes

## Student Notes

## **Cloud Native Associate**

General information—subject to change

- Cost: \$250 with one free retake
- Time limit: 90 minutes
- Passing score: 75%
- Validity: 3 years
- Languages: English
- Proctored: with audio, video, and screen-sharing feeds
- Multiple choice exam
- FAQ: <https://docs.linuxfoundation.org/tc-docs/certification/frequently-asked-questions-kcna>



## **Instructor Notes**

### **Student Notes**

<https://www.cncf.io/certification/kcna/>

## **Certified Kubernetes Administrator**

General information—subject to change

- Cost: \$375 with one free retake
- Time limit: 2 hours
- Passing score: 66%
- Validity: 3 years
- Languages: English, Simplified Chinese, and Japanese
- Proctored: with audio, video, and screen-sharing feeds
- Practical, command-line exam (with no multiple choice), access to some websites
- Updated frequently: currently etcd v3.5 and Kubernetes v1.22
- FAQ: <https://docs.linuxfoundation.org/tc-docs/certification/faq-cka-ckad-cks>



## **Instructor Notes**

### **Student Notes**

<https://www.cncf.io/certification/cka/>

## Certified Kubernetes Application Developer

General information—subject to change

- Cost: \$375 with one free retake
- Time limit: 2 hours
- Passing score: 66%
- Validity: 3 years
- Languages: English, Simplified Chinese, and Japanese
- Proctored: with audio, video, and screen-sharing feeds
- Practical, command-line exam (with no multiple choice), access to some websites
- Updated frequently: currently etcd v3.5 and Kubernetes v1.22
- FAQ: <https://docs.linuxfoundation.org/tc-docs/certification/faq-cka-ckad-cks>



## Instructor Notes

## Student Notes

<https://www.cncf.io/certification/ckad/>

## Certified Kubernetes Security Specialist

General information—subject to change

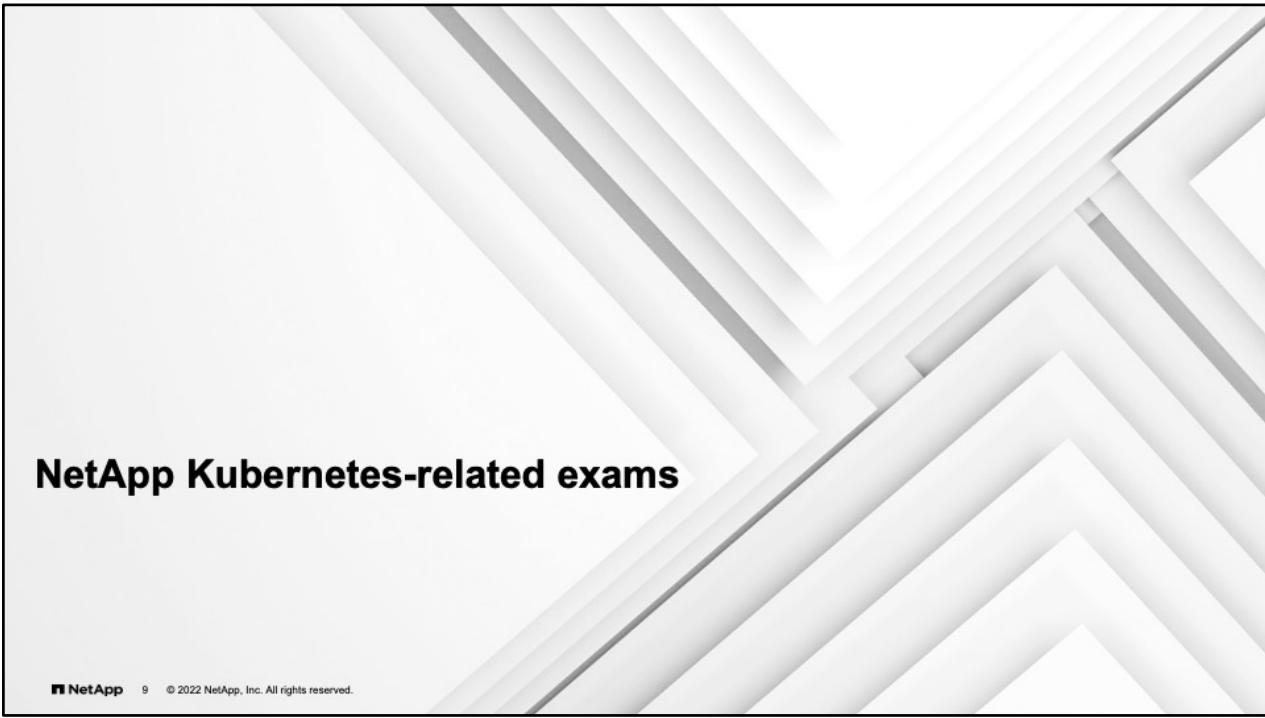
- Cost: \$375 with one free retake
- Time limit: 2 hours
- Passing score: 67%
- Validity: 2 years
- Languages: English, Simplified Chinese, and Japanese
- Proctored: with audio, video, and screen-sharing feeds
- Practical, command-line exam (with no multiple choice), access to some websites
- Updated frequently: currently etcd v3.5 and Kubernetes v1.22
- FAQ: <https://docs.linuxfoundation.org/tc-docs/certification/faq-cka-ckad-cks>



## Instructor Notes

## Student Notes

<https://www.cncf.io/certification/cks/>



## **NetApp Kubernetes-related exams**

 NetApp 9 © 2022 NetApp, Inc. All rights reserved.

**Instructor Notes**

**Student Notes**

## NetApp Certified Hybrid Cloud Implementation Engineer exam

Current version: NS0-403

- Number of test questions: **60**
- Time limit: **90 minutes** (additional 30 minutes allowed for non-English-speaking individuals)
- Passing score: **unpublished**
- Validity: **3 years**
- Languages: **English**
- Recommended field experience: **6–12 months**
- For more information: <https://www.netapp.com/support-and-training/netapp-university-training-and-certification/certification/implementation-engineer/>



NetApp 10 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

## Student Notes

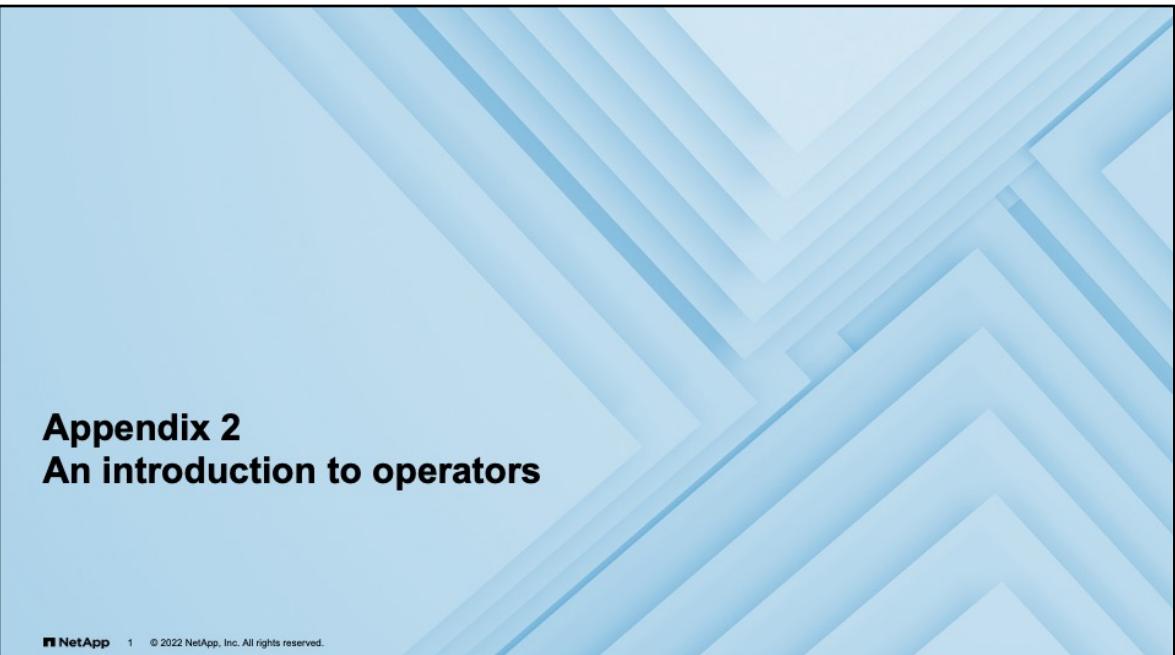
## Module summary

This module focused on enabling you to do the following:

- Describe Cloud Native Computing Foundation exams
- Describe NetApp Kubernetes-related exams

### Instructor Notes

### Student Notes



## **Appendix 2**

### **An introduction to operators**

 NetApp 1 © 2022 NetApp, Inc. All rights reserved.

**Instructor Notes**

**Student Notes**

## About this module

This module focuses on enabling you to do the following:

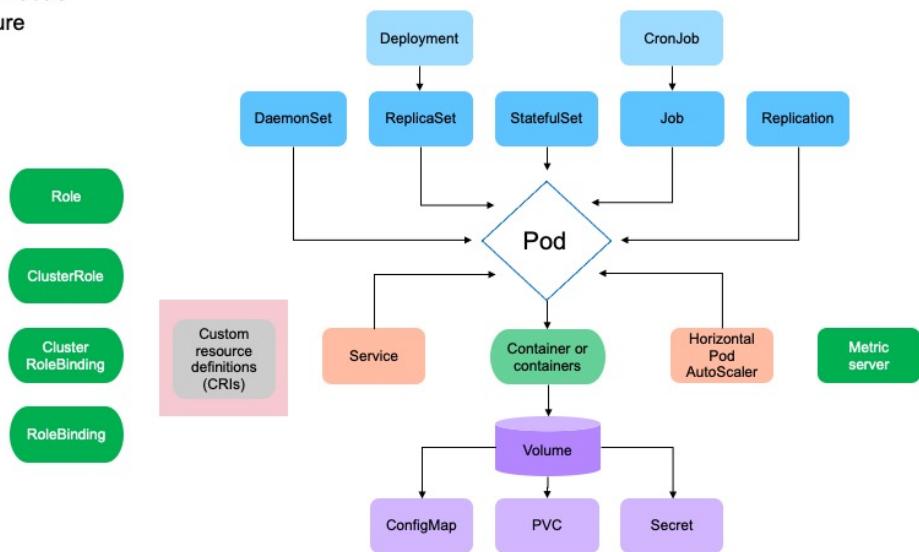
- Describe the frequently seen design patterns in Kubernetes
- Introduce the operator design pattern
- Review an example of how to create an operator for Kubernetes

### Instructor Notes

### Student Notes

## Kubernetes

Big picture



NetApp 3 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

## Student Notes



## **Design patterns for extending Kubernetes**

 NetApp 4 © 2022 NetApp, Inc. All rights reserved.

**Instructor Notes**

**Student Notes**

## Design patterns classifications

Design patterns are classified in three ways

| Category            | Focus area    | Examples                                                                                                                                |
|---------------------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Creational patterns | Instantiation | Factory method: Defines a generic interface for object creation<br>Object pools: Creates a pool of reusable objects                     |
| Structural patterns | Composition   | Façade: Encapsulates interfaces of a complex system<br>Proxy: Uses placeholder objects that represent other objects                     |
| Behavioral patterns | Communication | Iterator: Provides access to multiple elements of a complex object<br>Observer: Notifies another interested object when a state changes |

## Instructor Notes

## Student Notes

## Creational patterns

Instantiate objects within the environment based on various rules

- Sidecar pattern:
  - Is used to add monitoring, logging, configuration, testing, and networking
  - Adds support features to the main application (container)
  - Example: Istio sidecar
- Initialization pattern:
  - Is used during the startup of a pod
  - Provides "Separation of Concerns" within a pod
  - Example: initContainer—Executed for completion serially and expected to exit successfully
- Singleton pattern: Running only one instance (could be run as a StatefulSet with replica count of 1)

- Many examples natively:

- Deployments / ReplicaSets
- StatefulSets
- DaemonSet
- Job / Cronjob

## Instructor Notes

## Student Notes

## **Behavioral patterns**

Focuses on the communication and interaction between objects based on state changes

- Introspective pattern: Applications that are required to run in known environments
  - Goes against ephemeral nature of microservice architectures
  - Example: downwardAPI within a pod spec can expose Kubernetes data (mounted as a volume) to a container
- Controllers (and specifically operators pattern)

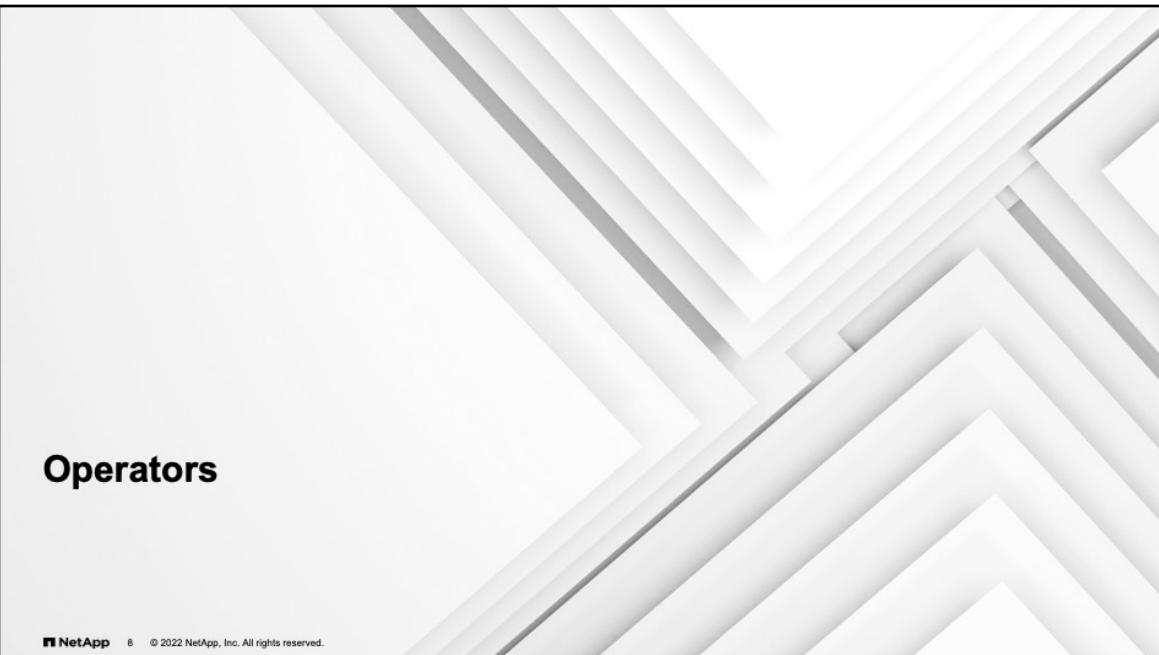
- Typically:
  - Responsibility assignment
  - Encapsulation of behaviors
  - Delegation of requests
- Native examples:
  - Schedulers
  - Kubelet metadata communication to API server (master and worker nodes)

## **Instructor Notes**

### **Student Notes**

For more information about downwardAPI, see

<https://kubernetes.io/docs/tasks/inject-data-application/downward-api-volume-expose-pod-information/>.



## **Operators**

 NetApp 8 © 2022 NetApp, Inc. All rights reserved.

**Instructor Notes**

**Student Notes**

## Recall: Kubernetes architecture

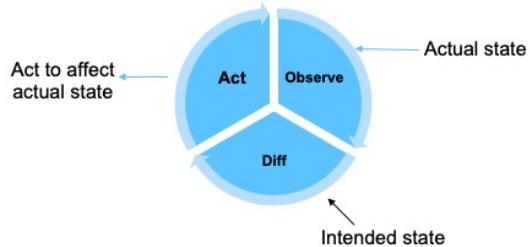
### Controllers



### Controllers

Have a control loop: observe, analyze, act

Example control loop ListWatcher: "Watches" for changes in etcd store and then remediates the situation



NetApp 9 © 2022 NetApp, Inc. All rights reserved.

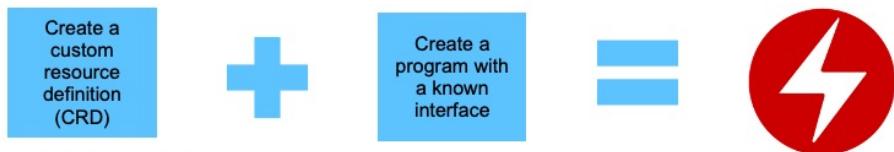
## Instructor Notes

## Student Notes

## Extending Kubernetes

Overview of operators

- Definition: A custom controller watching a custom resource (CR) type and taking application-specific actions
- To create an operator:



- Operators used for the following:
  - Infrastructure applications: Storage, caching, databases, basic functionality (such as "PodSets")
  - Main applications: A method for deploying a specific application
- For a community of operators, see <https://operatorhub.io/>

NetApp 10 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

### Student Notes

Another list of awesome operators: <https://github.com/operator-framework/awesome-operators>



## **Example operator implementation**

 NetApp 11 © 2022 NetApp, Inc. All rights reserved.

**Instructor Notes**

**Student Notes**

## Example scenario

### Sample workflow

- A slightly complex example (visitorapp):



- Manifests:

- Front end = deployment and service (has a configurable title)
- Back end = deployment and service (must know the database service name)
- Database = deployment and service
- Secret = credentials for the database

- The basic philosophy:

- "SRE for every application."
- "If a human operator needs to touch your system during normal operations, you have a bug."
- "Operators are the bug-fixes."

- Use operators to do the following:

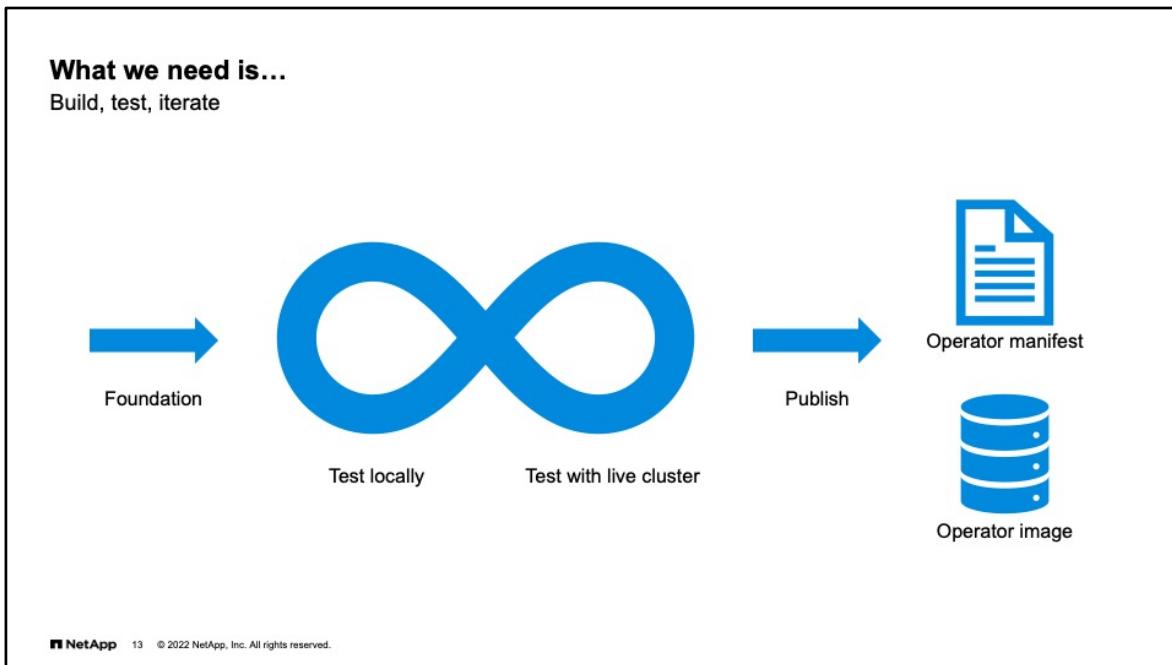
- Run
- Upgrade
- Respond to errors
- Slow performance

## Instructor Notes

### Student Notes

The example provided is from *Kubernetes Operators* (O'Reilly), by Jason Dobies and Joshua Wood.

The quotes are from *Site Reliability Engineering* (O'Reilly), by Betsy Beyer et al. (eds.).



## Instructor Notes

## Student Notes

The easiest way to create, test, and package an operator

- Found: <https://sdk.operatorframework.io/>

- Packages in three formats:

- Ansible
- Helm
- Go: The focus of this module

- Install:

```
$ RELEASE_VERSION=v0.18.0 # Set the release version variable
$ curl -LO https://github.com/operator-framework/operator-sdk/releases/download/\${RELEASE_VERSION}/operator-sdk-\${RELEASE_VERSION}-x86_64-linux-gnu # Linux
$ chmod +x operator-sdk-${RELEASE_VERSION}-x86_64-linux-gnu &&
 sudo mkdir -p /usr/local/bin/ &&
 sudo cp operator-sdk-${RELEASE_VERSION}-x86_64-linux-gnu /usr/local/bin/operator-sdk &&
 rm operator-sdk-${RELEASE_VERSION}-x86_64-linux-gnu
```

- Tool:

```
$ operator-sdk [commands]
```

Operator software development kit (SDK) provides supporting code, enabling you to focus on actual business logic.

## Instructor Notes

### Student Notes

For more information, see <https://sdk.operatorframework.io/docs/>.

## **Operator SDK project**

### Sample workflow

- Create an operator project:

- \$ mkdir -p \$HOME/projects
- \$ cd \$HOME/projects
- \$ operator-sdk new visitorapp-operator
- \$ cd visitorapp-operator

- Determine whether the operator is namespace-scoped or cluster-scoped

- Defaults to namespace-scoped
- To change it to a cluster-scoped:
  - Set the WATCH\_NAMESPACE to "" in deploy/operator.yaml
  - Change the kind to ClusterRole in deploy/role.yaml
  - Change the kind to ClusterRoleBinding, under roleRef change the kind to ClusterRole, update the subjects with key namespace with the appropriate value in deploy/role\_binding.yaml
  - Update the CRD spec to have scope of Cluster (see next step)
  - Update the \_types.go file by adding // +genclient:nonNamespaced to the struct (see next step)

 NetApp 15 © 2022 NetApp, Inc. All rights reserved.

## **Instructor Notes**

## **Student Notes**

## Custom resource definitions creation

### Sample workflow

- Define the spec and status of the CR in `pkg/apis/example/v1/visitorapp_types.go`:

```
type visitorappSpec struct {
 Size int32 `json:"size"`
 Title string `json:"title"`
}

type visitorappStatus struct {
 BackendImage string `json:"backendImage"`
 FrontendImage string `json:"frontendImage"`
}
```

- Add a new CRD:

```
$ operator-sdk add api --api-
version=example.com/v1
 --kind=VisitorApp
...
INFO[0000] Created
deploy/crds/example_v1_visitorapp_cr.yaml
INFO[0001] Created
deploy/crds/example_v1_visitorapp_crd.yaml ...
```

## Instructor Notes

## Student Notes

## Custom resource definitions verification

Need to have Kubernetes v1.16.x and later to run

- Verify and update the CRD file with the appropriate spec (and status) definitions:

```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
 name: visitorapp.example.com
spec:
 group: example.com
 names:
 kind: VisitorApp
 listKind: VisitorAppList
 plural: visitorapps
 singular: visitorapp
 ...
```

```
spec:
 type: object
 properties:
 size:
 type: integer
 title:
 type: string
 required:
 - size
status:
 type: object
 properties:
 backendImage:
 type: string
 frontendImage:
 type: string
```

Removed title from  
"required"

## Instructor Notes

## Student Notes

## **Controller creation**

Create a controller

- Add a new controller stub code:

```
$ operator-sdk add controller --api-version=example.com/v1 --kind=VisitorApp
```

- Modify pkg/controller/visitorapp/visitorapp\_controller.go to perform the following logic:

- Create a visitorapp Deployment if it does not exist
- Ensure that the Deployment size is the same as specified by the visitorapp CR spec
- Update the visitorapp CR status by using the status writer with the names of the visitorapp pods

- Review the watch for the primary resources as the add/update/delete events occur:

```
err := c.Watch(&source.Kind{Type: &examplev1.VisitorApp{}}, &handler.EnqueueRequestForObject{})
```

- Review the watch for the subresources as the add/update/delete events occur:

```
err := c.Watch(&source.Kind{Type: &appsv1.Deployment{}}, &handler.EnqueueRequestForOwner{
 IsController: true,
 OwnerType: &examplev1.VisitorApp{},
})
...
```

 NetApp 18 © 2022 NetApp, Inc. All rights reserved.

## **Instructor Notes**

## **Student Notes**

## Reconcile function

Core declarative implementation

- Verify the Reconcile() method that implements the reconcile loop:

```
func (r *ReconcileVisitorApp) Reconcile(request reconcile.Request) (reconcile.Result, error) {
 // Lookup the visitorapp instance for this reconcile request
 instance := &examplev1.VisitorApp{}
 err := r.client.Get(context.TODO(), request.NamespacedName, instance)
 ...
}
```

Makes sure that all the objects are created for a visitorapp

- Results of the function:

- return reconcile.Result{}, nil //successful
- return reconcile.Result{}, err //error - requeue
- return reconcile.Result{Requeue: true}, nil //failed for another reason -requeue
- return reconcile.Result{RequeueAfter: time.Second\*5}, nil //requeue after 5 secs

 NetApp 19 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

## Student Notes

## Build and run the operator

Putting it all together

- Register the CRD:

```
$ kubectl create -f deploy/crds/example.com_visitorapps_crd.yaml
```

- Build the operator:

```
$ operator-sdk build [account]/visitorapp-operator:v0.0.1
```

- Push to Docker Hub:

```
$ docker login
```

```
$ docker push [account]/visitorapp-operator:v0.0.1
```

- Deploy the operator:

```
$ kubectl create -f deploy/service_account.yaml
$ kubectl create -f deploy/role.yaml
$ kubectl create -f deploy/role_binding.yaml
$ kubectl create -f deploy/operator.yaml
```

- Verify the running operator:

```
$ kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
visitorapp-operator 1/1 1 1 1m
```

You can test your operator:

```
operator-sdk run local --
watch-namespace [ns]
```

Optionally, use Operator Lifecycle Manager to deploy operators

Best practice: Modify role-based access control (RBAC) resources created by the operator SDK to grant only necessary permissions to run operator

Best practice: Aggregate all YAML files into a single YAML file

 NetApp 20 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

## Student Notes

## Test the operator

Create a VisitorApp type

- Create a definition file for an instance of the CR:

```
$ cat deploy/crds/example.com_v1_visitorapp_cr.yaml
apiVersion: example.com/v1
kind: visitorapp
metadata:
 name: example-visitorapp
spec:
 size: 1
 title: "My New Operator Dashboard"
```

- Create an instance of the CR:

```
$ kubectl apply -f deploy/crds/example.com_v1_visitorapp_cr.yaml
```

- Verify that the custom operator created the desired deployment:

```
$ kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
example-visitorapp-backend 1/1 1 1 5m
example-visitorapp-frontend 1/1 1 1 5m
example-visitorapp-mysql 1/1 1 1 5m
visitorapp-operator 1/1 1 1 5m
```

 NetApp 21 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

## Student Notes

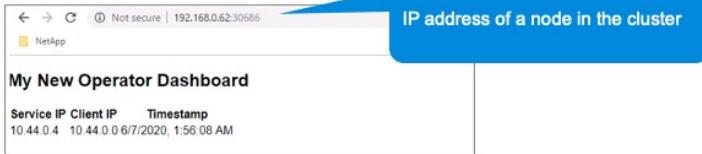
## SRE for every app

Everything works

- Verify that the custom operator created the desired services:

```
$ kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
example-visitorapp-backend-service NodePort 10.104.164.255 <none> 8000:30685/TCP 68s
example-visitorapp-frontend-service NodePort 10.107.28.229 <none> 3000:30686/TCP 68s
example-visitorapp-mysql-service ClusterIP None <none> 3306/TCP 70s
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 271d
visitorapp-operator-metrics ClusterIP 10.103.18.72 <none> 8383/TCP,8686/TCP 11m
```

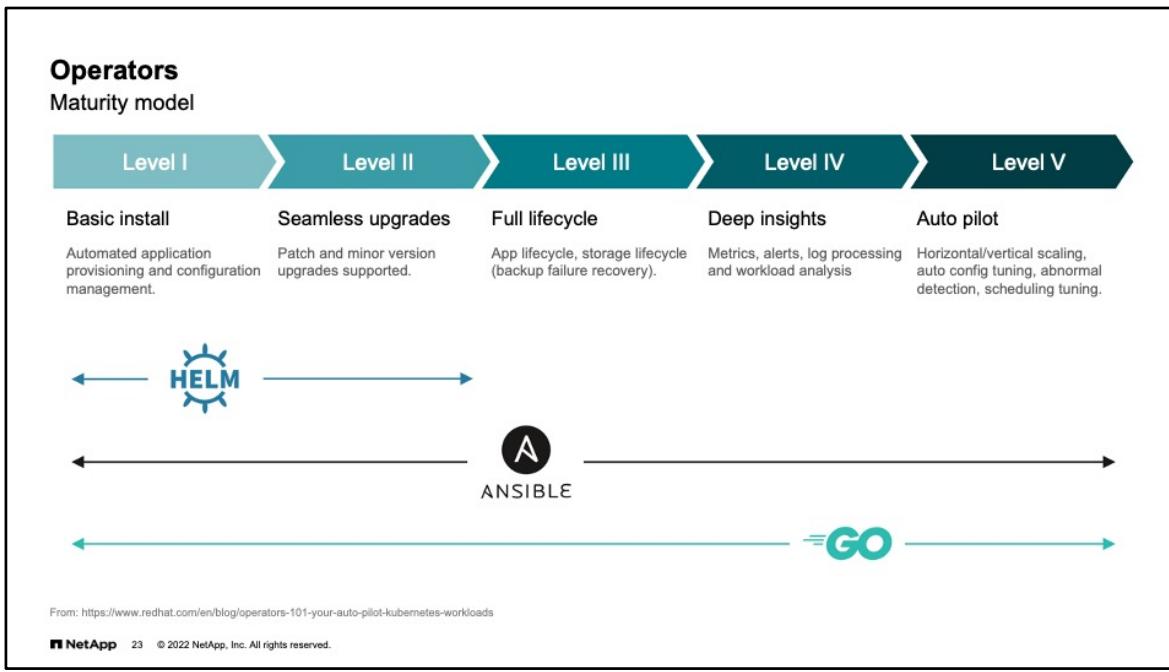
- Access the front-end image:



NetApp 22 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

## Student Notes



## Instructor Notes

## Student Notes

## Four golden signals sent to software

As the operator matures, verify that these metrics are captured

|                   | Definition                                 | Operator                                            |
|-------------------|--------------------------------------------|-----------------------------------------------------|
| <b>Latency</b>    | How long it takes to do something          | Measure application-specific latencies              |
| <b>Traffic</b>    | How frequently a service is requested      | Measure transactions occurring with the application |
| <b>Errors</b>     | The frequency and type of failed requests  | Useful HTTP codes                                   |
| <b>Saturation</b> | How much consumption of a limited resource | Could be CPU, memory, networking, or I/O            |

 NetApp 24 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

### Student Notes

Four golden signals are taken from *Site Reliability Engineering* (O'Reilly), by Betsy Beyer et al. (eds.).

## **Best practices of operators**

A partial list

- Should run as a single Kubernetes deployment
- Should run with the minimalist permissions required
- Should define new customer resource types
- Should use existing Kubernetes types whenever possible for child resources
- Should take care when naming child resources (must be unique)
- Should use “finalizers” to delete child resources not owned by a CR
- Termination should not affect the operand
  - NOTE: Removing a CRD does affect the operand application; deleting a CRD deletes a CR or CRs

## **Instructor Notes**

### **Student Notes**

Finalizers: “Finalizers allow controllers to implement asynchronous pre-delete hooks. Let’s say you create an external resource (such as a storage bucket) for each object of your API type, and you want to delete the associated external resource on object’s deletion from Kubernetes, you can use a finalizer to do that.”

For more information, see <https://book.kubebuilder.io/reference/using-finalizers.html>.

The example does not use a finalizer because it did not create any “outside” resources that were not “owned” by the CR.

## Module summary

- This module focused on enabling you to do the following:
- Describe the frequently seen design patterns in Kubernetes
- Introduce the operator design pattern
- Review an example of how to create an operator for Kubernetes

### Instructor Notes

### Student Notes



## **Appendix 3**

### **Astra Trident monitoring**

 NetApp 1 © 2022 NetApp, Inc. All rights reserved.

**Instructor Notes**

**Student Notes**

## About this module

This module focuses on enabling you to do the following:

- List the available options for monitoring NetApp Astra Trident
- Use Prometheus to monitor Astra Trident to ensure storage availability

### Instructor Notes

### Student Notes



## **Available options for monitoring Astra Trident**

 NetApp 3 © 2022 NetApp, Inc. All rights reserved.

**Instructor Notes**

**Student Notes**

**NetApp Cloud Insights for Kubernetes**  
A powerful tool for exploring Kubernetes metrics

The dashboard features four main circular icons on the left:

- Monitor**: A blue circle with a monitor icon.
- Analyze**: A grey circle with a bar chart icon.
- Optimize**: A green circle with a gear icon.
- Show-back**: A purple circle with a film strip icon.

The central area displays various metrics and charts:

- Object Counts**: Shows 38 Nodes, 366 Pods, 346 Healthy, 20 Alerting, and 0 Pending.
- Capacity**: A large circular gauge showing usage across three time periods: 12:00 PM, 1: Day, and 12:00 PM, 1: Day. It indicates 3% of allocatable capacity used.
- Resource Usage**: Three donut charts for CPU, Memory, and Storage, each showing 3%, 17%, and 57% of allocatable capacity respectively.
- Highest CPU Demand by Pod**: Lists pods with their CPU demand:
 

| Pod                                | CPU Demand |
|------------------------------------|------------|
| 301.12m elasticsearch-gold-0002... | 8.83 GB    |
| 383.26m telegraf-ri-jblrg...       | 8.83 GB    |
| 377.1m elasticsearch-gold-0001...  | 8.83 GB    |
| 235.56m master-api-ip-10-30-23...  | 8.23 GB    |
| 59.45m oci-service-tenants-5d4...  | 4.58 GB    |
- Highest Memory Demand by Pod**: Lists pods with their memory demand:
 

| Pod                        | Memory Demand |
|----------------------------|---------------|
| elasticsearch-gold-0001... | 300 GB        |
| elasticsearch-gold-0002... | 300 GB        |
| elasticsearch-gold-0003... | 300 GB        |
| elasticsearch-gold-0004... | 300 GB        |
| elasticsearch-gold-0005... | 300 GB        |
- Highest Storage Demand by Pod**: Lists pods with their storage demand:
 

| Pod                       | Storage Demand |
|---------------------------|----------------|
| taskmanager-ct70f54546... | 300 GB         |
| kafka-0                   | 300 GB         |
| jobmanager-0              | 300 GB         |
| jobmanager-1              | 300 GB         |
- Containers**: Summary of container status: 386 Containers (373 Ready, 5 Waiting, 8 Terminated, 0 Unknown).

At the bottom left: **NetApp** 4 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

## Student Notes



## **Monitor Astra Trident with Prometheus**

 NetApp 5 © 2022 NetApp, Inc. All rights reserved.

### **Instructor Notes**

### **Student Notes**

## A popular option

Prometheus and Grafana



[Prometheus](#)

- Cloud Native Computing Foundation project started at the same time as Kubernetes
- Open-source monitoring solution



[Grafana](#)

- Observability stack (monitoring, dashboarding, alerting ...)

## Instructor Notes

## Student Notes

Harvest is not discussed in this course.

### **Prometheus setup overview**

To install: <https://github.com/prometheus-operator/kube-prometheus>

- Create a monitoring namespace (default namespace for Prometheus)
- Install the custom resource definitions for Prometheus
- Apply the manifests to install Prometheus and related tools



### **Instructor Notes**

### **Student Notes**

## Prometheus tools

Usual pods deployed in a Prometheus setup

- Prometheus Operator (one pod): is used to manage all other Prometheus pods (deploys two Prometheus instances)
- Prometheus k8s (two pods): is used to aggregate metrics
- Alertmanager (three pods): is used to communicate Prometheus data with external services
- Kube State Metrics (one pod): is used to scrape telemetry data from your Kubernetes control plane services
- Node Explorer (one for each node in your cluster): is used to scrape telemetry data from your Kubernetes nodes
- Grafana (one pod): is used to provide a front-end UI for the Prometheus data
- Blackbox Exporter (one pod): enables black-box probing of endpoints over HTTP, HTTPS, DNS, TCP, and Internet ControlMessage Protocol (ICMP)
- Prometheus Adapter (two pods): contains an implementation of the Kubernetes resource metrics, custom metrics, and external metrics APIs to use with Horizontal Pod Autoscaler

## Instructor Notes

## Student Notes

## Integration with Prometheus

Supported in Astra Trident 20.01 or later



### Kubernetes objects:

- Resource Quota: PVC limit
- Resource Quota: Capacity limit
- Limit Range: PVC maximum size

### Astra Trident back-end parameters:

- limitVolumeSize
- limitAggregateUsage
- qtreesPerFlexvol and lunsPerFlexvol
- qosPolicy & adaptiveQosPolicy

### ONTAP parameters:

- Maximum number of volumes per tenant
- Dedicated aggregates

NetApp 9 © 2022 NetApp, Inc. All rights reserved.

## Instructor Notes

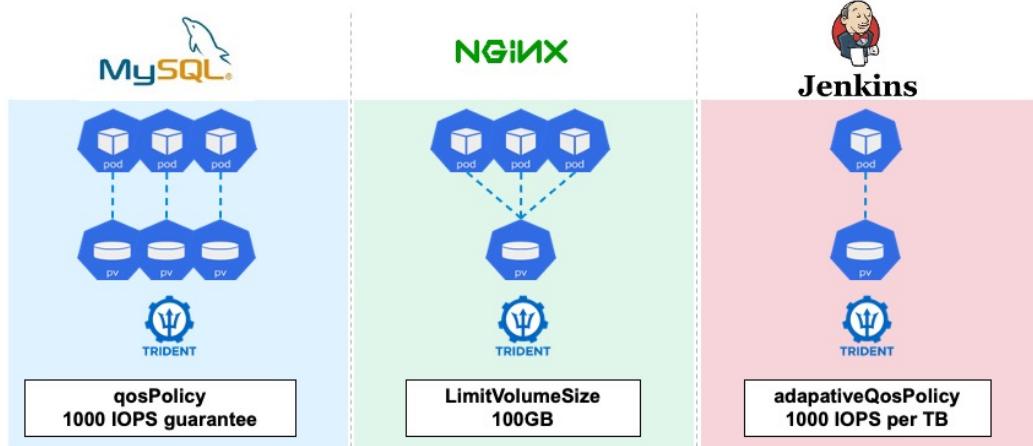
### Student Notes

For more information,

see <https://github.com/NetApp/trident/blob/master/core/metrics.go> and  
<https://netapp.io/2020/02/20/a-primer-on-prometheus-trident>.

## Examples

Collecting data with Astra Trident



NetApp 10 © 2022 NetApp, Inc. All rights reserved.

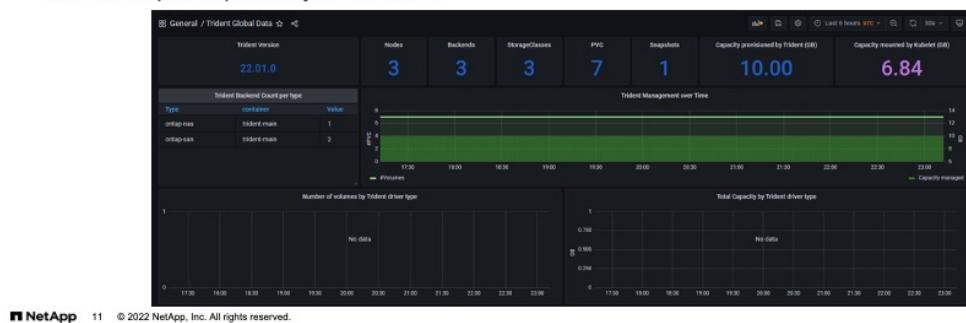
## Instructor Notes

## Student Notes

## Monitoring Astra Trident as a Prometheus target

Using Prometheus and Grafana

1. Create a role and rolebinding to allow Prometheus to monitor the namespace used for Astra Trident
2. Create a Prometheus servicemonitor to point to the metrics port exposed by Astra Trident
3. Create a Grafana dashboard to review the Astra Trident data



## Instructor Notes

## Student Notes

## Module summary

This module focused on enabling you to do the following:

- List the available options for monitoring Astra Trident
- Use Prometheus to monitor Astra Trident to ensure storage availability

### Instructor Notes

### Student Notes



## Complete an exercise

Appendix 3: Astra Trident monitoring

### Monitoring Astra Trident

- Task 1: Host Prometheus and supporting applications in Kubernetes
- Task 2: Configure a sample app as a Prometheus target
- Task 3: Configure Astra Trident as a Prometheus target

Exercise duration:  
**60 minutes**

## Instructor Notes

### Student Notes

See the instructions in your Exercise Guide.