Überblick über NetApp Storage-Integrationen

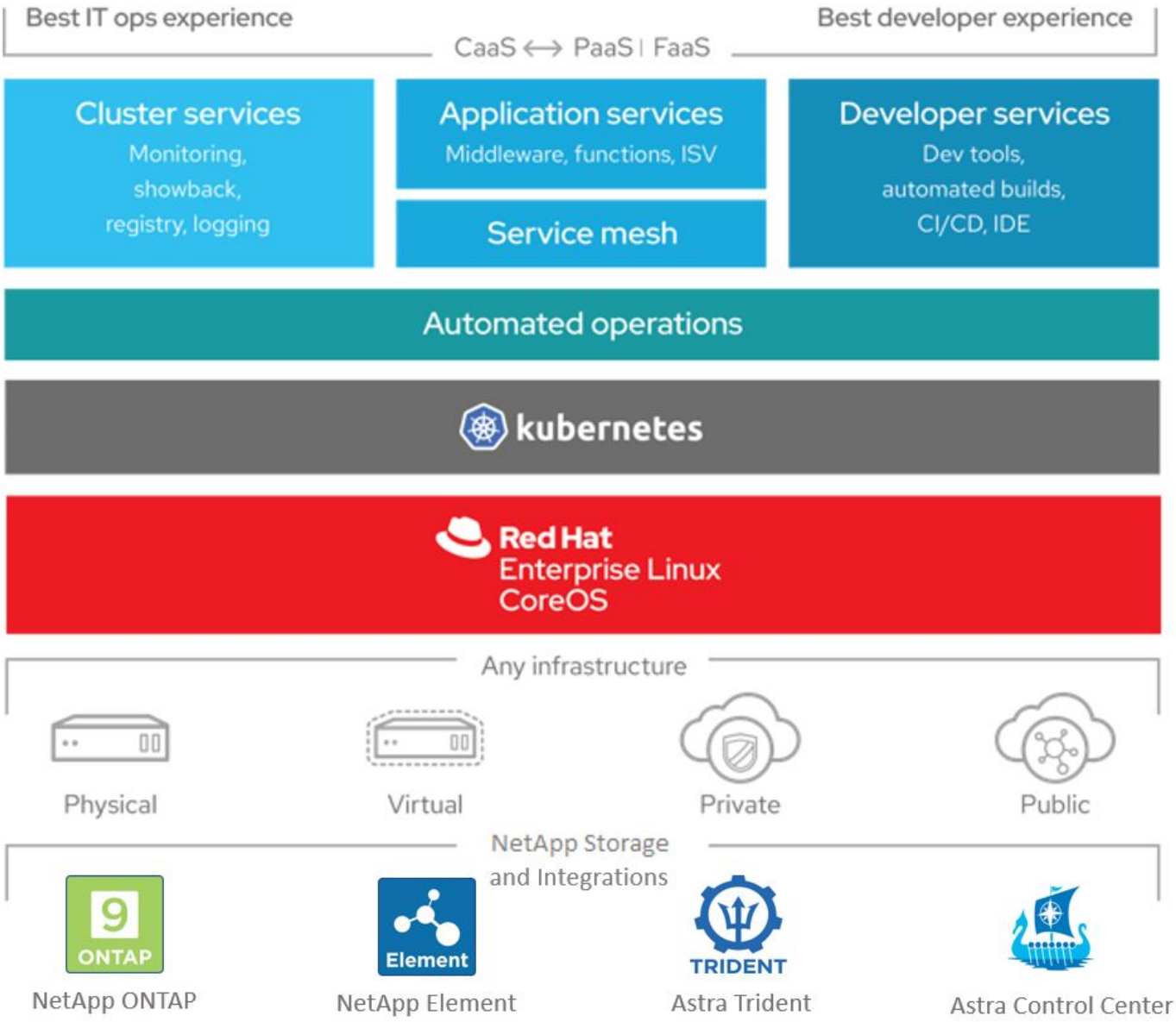
NetApp Solutions

NetApp
April 26, 2024

Inhalt

Überblick über die NetApp Storage-Integration

NetApp bietet verschiedene Produkte, die Sie bei der Orchestrierung und dem Management persistenter Daten in Container-basierten Umgebungen wie Red hat OpenShift unterstützen.



NetApp Astra Control bietet eine umfassende Auswahl an Storage- und applikationsspezifischen Datenmanagement-Services für zustandsorientierte Kubernetes Workloads auf Basis der NetApp Datensicherungstechnologie. Der Astra Control Service unterstützt statusorientierte Workloads in Cloud-nativen Kubernetes-Implementierungen. Das Astra Control Center unterstützt statusorientierte Workloads in lokalen Implementierungen wie Red hat OpenShift. Weitere Informationen finden Sie auf der NetApp Astra Control Website "Hier".

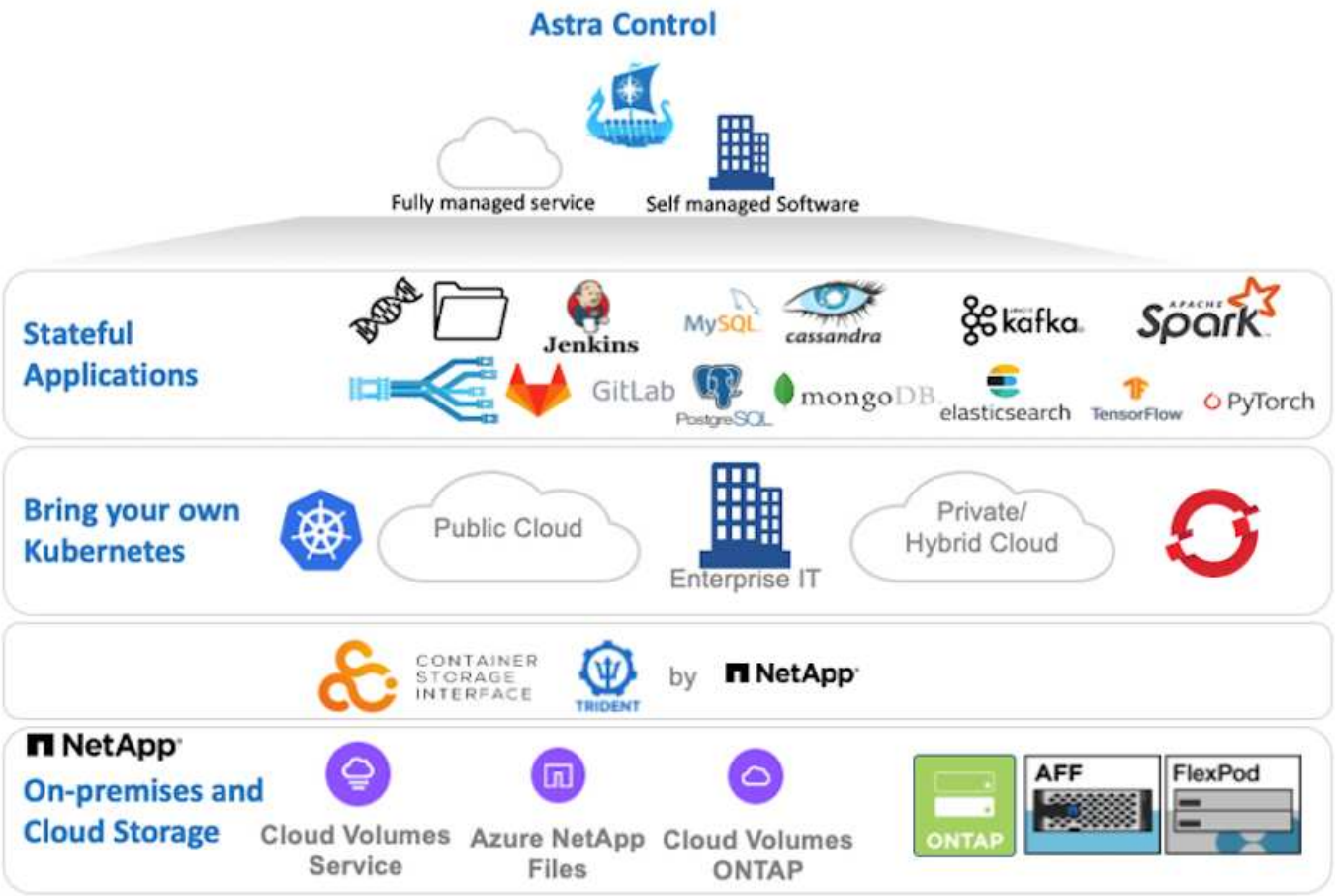
NetApp Astra Trident ist ein Open-Source- und vollständig unterstützter Storage-Orchestrator für Container und Kubernetes-Distributionen, einschließlich Red hat OpenShift. Weitere Informationen finden Sie auf der Astra Trident Website "Hier".

Auf den folgenden Seiten finden Sie zusätzliche Informationen zu den NetApp Produkten, die für das Management von Applikationen und persistentem Storage in Red hat OpenShift mit NetApp validiert wurden:

- "NetApp Astra Control Center"
- "NetApp Astra Trident"

Übersicht über das NetApp Astra Control Center

Das NetApp Astra Control Center bietet umfassende Storage- und applikationsorientierte Datenmanagement-Services für statusorientierte Kubernetes Workloads in einer On-Premises-Umgebung mit NetApp Datensicherungstechnologie.



Das NetApp Astra Control Center kann auf einem Red hat OpenShift-Cluster installiert werden. Über den Astra Trident Storage-Orchestrator mit Storage-Klassen und Storage-Back-Ends für NetApp ONTAP Storage-Systeme implementiert und konfiguriert werden.

Informationen zur Installation und Konfiguration von Astra Trident zur Unterstützung des Astra Control Center finden Sie unter "Dieses Dokument hier einfügen".

In einer Umgebung mit Cloud-Anbindung sorgt Astra Control Center mithilfe von Cloud Insights für erweitertes Monitoring und Telemetrie. Liegt keine Cloud Insights-Verbindung vor, ist eingeschränktes Monitoring und Telemetrie (7 Tage mit Kennzahlen) verfügbar und über offene metrische Endpunkte in native Kubernetes-Monitoring-Tools (Prometheus und Grafana) exportiert.

Astra Control Center ist vollständig in das AutoSupport- und Active IQ-Ecosystem von NetApp integriert und bietet damit Support für Benutzer, Hilfestellung bei der Fehlerbehebung und Statistiken zur Anzeige der Nutzungsstatistik.

Zusätzlich zur kostenpflichtigen Version des Astra Control Center ist eine 90-Tage-Evaluierungslizenz

verfügbar. Die Evaluierungsversion wird durch die E-Mail und die Community (Slack-Kanal) unterstützt. Kunden haben Zugriff auf diese und andere Knowledge-Base-Artikel sowie auf die Dokumentation, die über das Produkt-Support-Dashboard verfügbar sind.

Besuchen Sie das NetApp Astra Control Center "Astra-Website".

Installationsvoraussetzungen für Astra Control Center

1. Ein oder mehrere Red hat OpenShift-Cluster. Die Versionen 4.6 EUS und 4.7 werden derzeit unterstützt.
2. Astra Trident muss bereits auf jedem Red hat OpenShift-Cluster installiert und konfiguriert sein.
3. Mindestens ein NetApp ONTAP Storage-System mit ONTAP 9.5 oder höher

   Es ist Best Practice für jede OpenShift-Installation an einem Standort, die über eine dedizierte SVM für persistenten Storage verfügt. Implementierungen an mehreren Standorten erfordern zusätzliche Storage-Systeme.

4. Auf jedem OpenShift-Cluster muss ein Trident-Storage-Back-End mit einer durch einen ONTAP-Cluster gesicherten SVM konfiguriert werden.
5. Eine Standard-StorageClass-Konfiguration auf jedem OpenShift-Cluster mit Astra Trident als Storage-provisionierung
6. Auf jedem OpenShift-Cluster muss ein Load Balancer installiert und konfiguriert werden, um den Lastausgleich zu ermöglichen und OpenShift Services offenzulegen.

   Siehe den Link "Hier" Weitere Informationen zu Load Balancer, die zu diesem Zweck validiert wurden.

7. Eine private Image-Registrierung muss konfiguriert sein, um die NetApp Astra Control Center Images zu hosten.

   Siehe den Link "Hier" So installieren und konfigurieren Sie eine private OpenShift-Registrierung zu diesem Zweck.

8. Sie benötigen Cluster-Admin-Zugriff auf das Red hat OpenShift-Cluster.
9. Sie müssen Administratorzugriff auf NetApp ONTAP Cluster haben.
10. Sie erhalten eine Admin-Workstation mit den Tools Docker oder Podman, tridentctl und oc oder kubectl, die Sie installiert und Ihrem €Pfad hinzugefügt haben.

    Bei Docker-Installationen muss die Docker-Version größer als 20.10 sein, und Podman-Installationen müssen über eine Podman-Version von mehr als 3.0 verfügen.

Installieren Sie Astra Control Center

OperatorHub wird verwendet

1. Melden Sie sich bei der NetApp Support-Website an und laden Sie die neueste Version des NetApp Astra Control Center herunter. Dazu ist eine Lizenz erforderlich, die an Ihr NetApp Konto angehängt ist. Nach dem Download des Tarballs, übertragen Sie es auf die Admin-Workstation.

   Um mit einer Testlizenz für Astra Control zu beginnen, besuchen Sie die "Astra: Anmelde-Website".

2. Entpacken Sie den tar-Ball und ändern Sie das Arbeitsverzeichnis in den resultierenden Ordner.

```
[netapp-user@rhel7 ~]$ tar -vxzf astra-control-center-
21.12.60.tar.gz
[netapp-user@rhel7 ~]$ cd astra-control-center-21.12.60
```

3. Bevor Sie mit der Installation beginnen, schieben Sie die Astra Control Center-Bilder in eine Bildregistrierung. Hierfür wählen Sie entweder Docker oder Podman. In diesem Schritt werden Anweisungen für beide angegeben.

Podman

a. Exportieren Sie den FQDN der Registrierung mit dem Namen der Organisation/des Namespace/Projekts als Umgebungsvariable ‘registry’.

```
[netapp-user@rhel7 ~]$ export REGISTRY=astra-
registry.apps.ocp-vmw.cie.netapp.com/netapp-astra
```

b. Melden Sie sich bei der Registrierung an.

```
[netapp-user@rhel7 ~]$ podman login -u ocp-user -p password
--tls-verify=false astra-registry.apps.ocp-vmw.cie.netapp.com
```

> Wenn Sie verwenden `kubeadmin` Benutzer, um sich bei der privaten Registrierung anzumelden, dann Token statt Passwort verwenden - `podman login -u ocp-user -p token --tls-verify=false astra-registry.apps.ocp-vmw.cie.netapp.com`.

> Alternativ können Sie ein Service-Konto erstellen, dem Registry-Editor und/oder der Registry-Viewer-Rolle zuweisen (ob Sie Push/Pull-Zugriff benötigen) und sich mithilfe des Token des Service-Kontos bei der Registrierung anmelden.

c. Erstellen Sie eine Shell-Skriptdatei, und fügen Sie den folgenden Inhalt darin ein.

```
[netapp-user@rhel7 ~]$ vi push-images-to-registry.sh

for astraImageFile in $(ls images/*.tar) ; do
  # Load to local cache. And store the name of the loaded
image trimming the 'Loaded images: '
  astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image(s): //')
  astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
  # Tag with local image repo.
  podman tag ${astraImage} ${REGISTRY}/${astraImage}
  # Push to the local repo.
  podman push ${REGISTRY}/${astraImage}
done
```

> Wenn Sie nicht vertrauenswürdige Zertifikate für Ihre Registrierung verwenden, bearbeiten Sie das Shell-Skript und verwenden Sie es `--tls-verify=false` Für den Podman-Push-Befehl `podman push $REGISTRY/$(echo $astraImage | sed 's/[\/]\+\///') --tls-verify=false`.

d. Machen Sie die Datei ausführbar.

```
[netapp-user@rhel7 ~]$ chmod +x push-images-to-registry.sh
```

e. Das Shell-Skript ausführen.

```
[netapp-user@rhel7 ~]$ ./push-images-to-registry.sh
```

Docker

a. Exportieren Sie den FQDN der Registrierung mit dem Namen der Organisation/des Namespace/Projekts als Umgebungsvariable ‘registry’.

```
[netapp-user@rhel7 ~]$ export REGISTRY=astra-
registry.apps.ocp-vmw.cie.netapp.com/netapp-astra
```

b. Melden Sie sich bei der Registrierung an.

```
[netapp-user@rhel7 ~]$ docker login -u ocp-user -p password
astra-registry.apps.ocp-vmw.cie.netapp.com
```

> Wenn Sie verwenden `kubeadmin` Benutzer, um sich bei der privaten Registrierung anzumelden, dann Token statt Passwort verwenden - `docker login -u ocp-user -p token astra-registry.apps.ocp-vmw.cie.netapp.com`.

> Alternativ können Sie ein Service-Konto erstellen, dem Registry-Editor und/oder der Registry-Viewer-Rolle zuweisen (ob Sie Push/Pull-Zugriff benötigen) und sich mithilfe des Token des Service-Kontos bei der Registrierung anmelden.

c. Erstellen Sie eine Shell-Skriptdatei, und fügen Sie den folgenden Inhalt darin ein.

```
[netapp-user@rhel7 ~]$ vi push-images-to-registry.sh

for astraImageFile in $(ls images/*.tar) ; do
  # Load to local cache. And store the name of the loaded
image trimming the 'Loaded images: '
  astraImage=$(docker load --input ${astraImageFile} | sed
's/Loaded image: //')
  astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
  # Tag with local image repo.
  docker tag ${astraImage} ${REGISTRY}/${astraImage}
  # Push to the local repo.
  docker push ${REGISTRY}/${astraImage}
done
```

d. Machen Sie die Datei ausführbar.

```
[netapp-user@rhel7 ~]$ chmod +x push-images-to-registry.sh
```

e. Das Shell-Skript ausführen.

```
[netapp-user@rhel7 ~]$ ./push-images-to-registry.sh
```

4. Wenn Sie private Bildregistries verwenden, die nicht öffentlich vertrauenswürdig sind, laden Sie die TLS-Zertifikate der Bildregistrierung auf die OpenShift-Knoten hoch. Erstellen Sie dazu im Namespace openshift-config eine configmap mit den TLS-Zertifikaten und patchen Sie sie auf die Cluster-Image-Konfiguration, damit das Zertifikat vertrauenswürdig ist.

```
[netapp-user@rhel7 ~]$ oc create configmap default-ingress-ca -n
openshift-config --from-file=astra-registry.apps.ocp
-vmw.cie.netapp.com=tls.crt

[netapp-user@rhel7 ~]$ oc patch image.config.openshift.io/cluster
--patch '{"spec":{"additionalTrustedCA":{"name":"default-ingress-
ca"}}}' --type=merge
```

> Wenn Sie eine interne OpenShift-Registrierung mit Standard-TLS-Zertifikaten vom Ingress Operator mit einer Route verwenden, müssen Sie den vorherigen Schritt dennoch befolgen, um die Zertifikate auf den Routing-Hostnamen zu patchen. Um die Zertifikate aus dem Ingress Operator zu extrahieren, können Sie den Befehl verwenden `oc extract secret/router-ca --keys=tls.crt -n openshift-ingress-operator`.

5. Erstellen Sie einen Namespace `netapp-acc-operator` Für Astra Control Center.

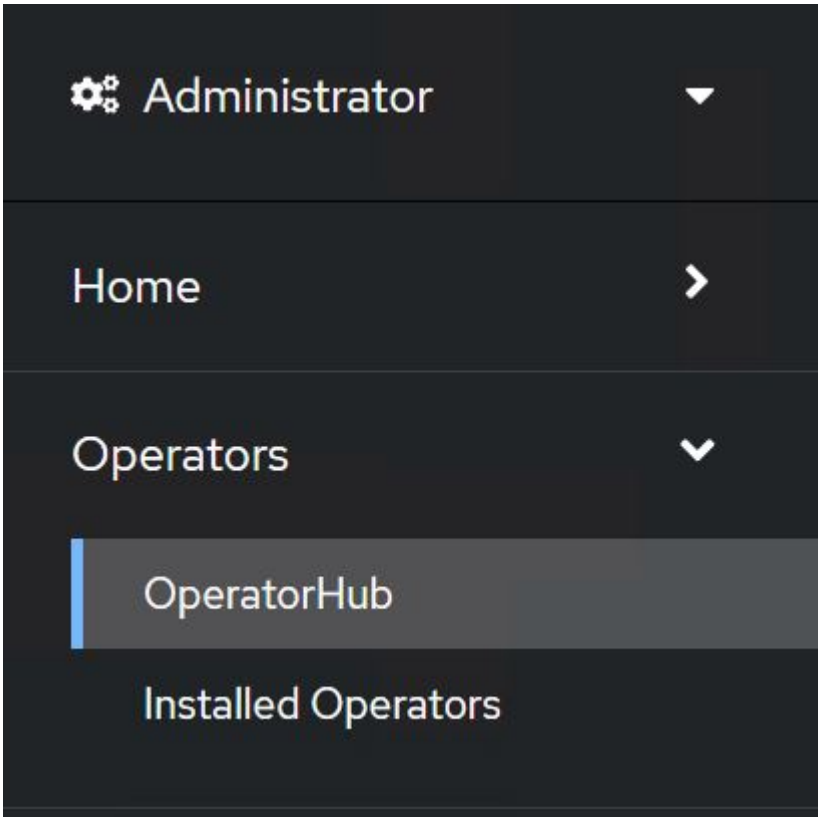
```
[netapp-user@rhel7 ~]$ oc create ns netapp-acc-operator

namespace/netapp-acc-operator created
```

6. Erstellen Sie ein Geheimnis mit Anmeldeinformationen, um sich in der Bildregistrierung anzumelden `netapp-acc-operator` Namespace.

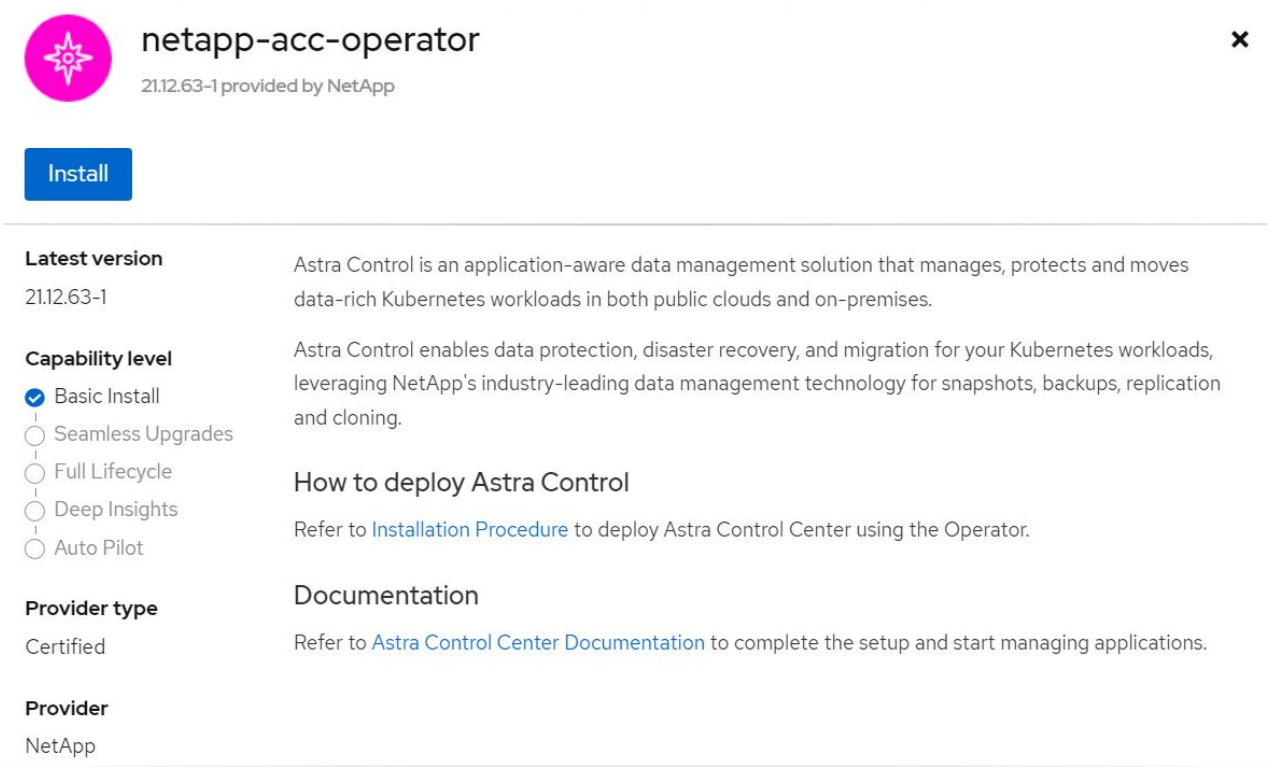
```
[netapp-user@rhel7 ~]$ oc create secret docker-registry astra-
registry-cred --docker-server=astra-registry.apps.ocp
-vmw.cie.netapp.com --docker-username=ocp-user --docker
-password=password -n netapp-acc-operator

secret/astra-registry-cred created
```

7. Melden Sie sich bei der Red hat OpenShift GUI-Konsole mit Zugriff auf Cluster-Administratoren an.
8. Wählen Sie in der Dropdown-Liste Perspektive den Eintrag Administrator aus.

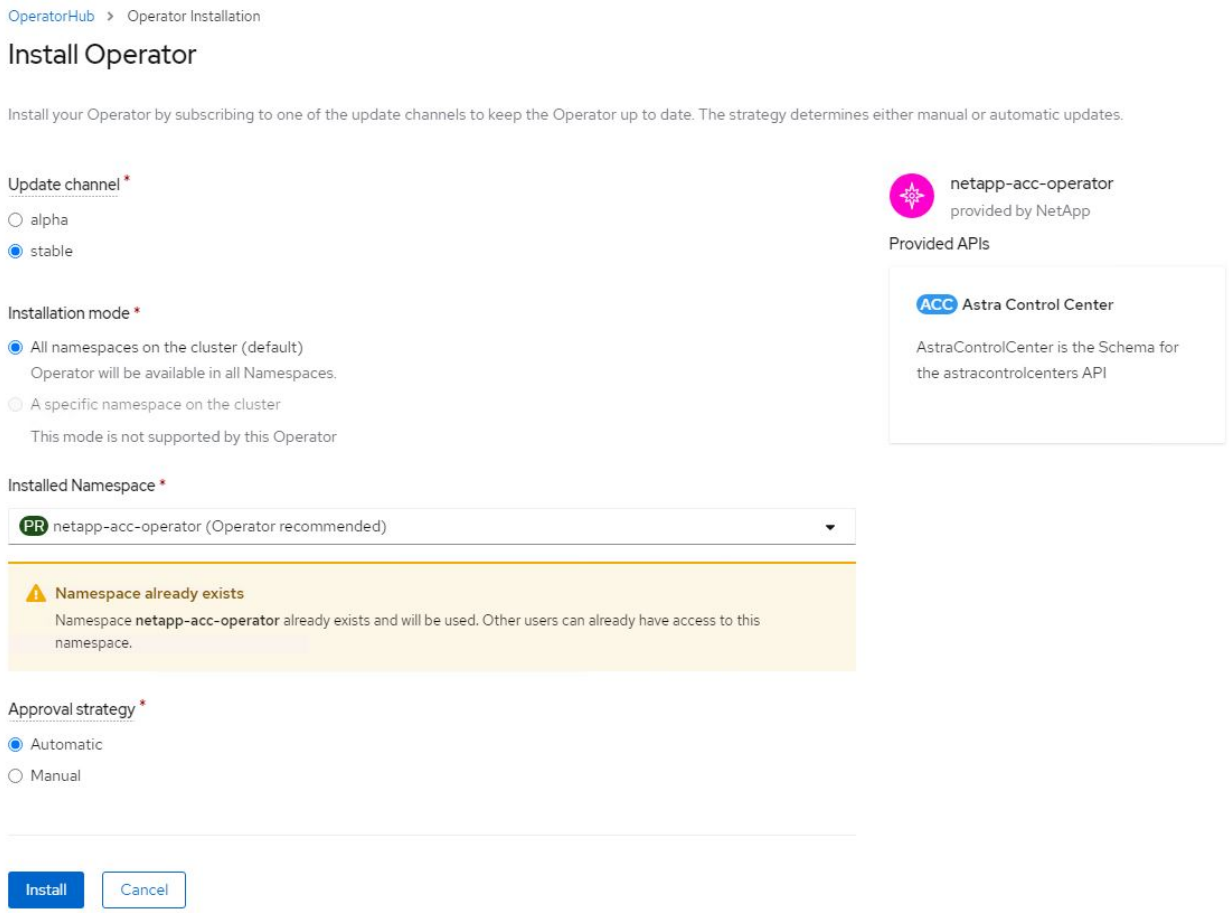
9. Navigieren Sie zu Operators > OperatorHub, und suchen Sie nach Astra.



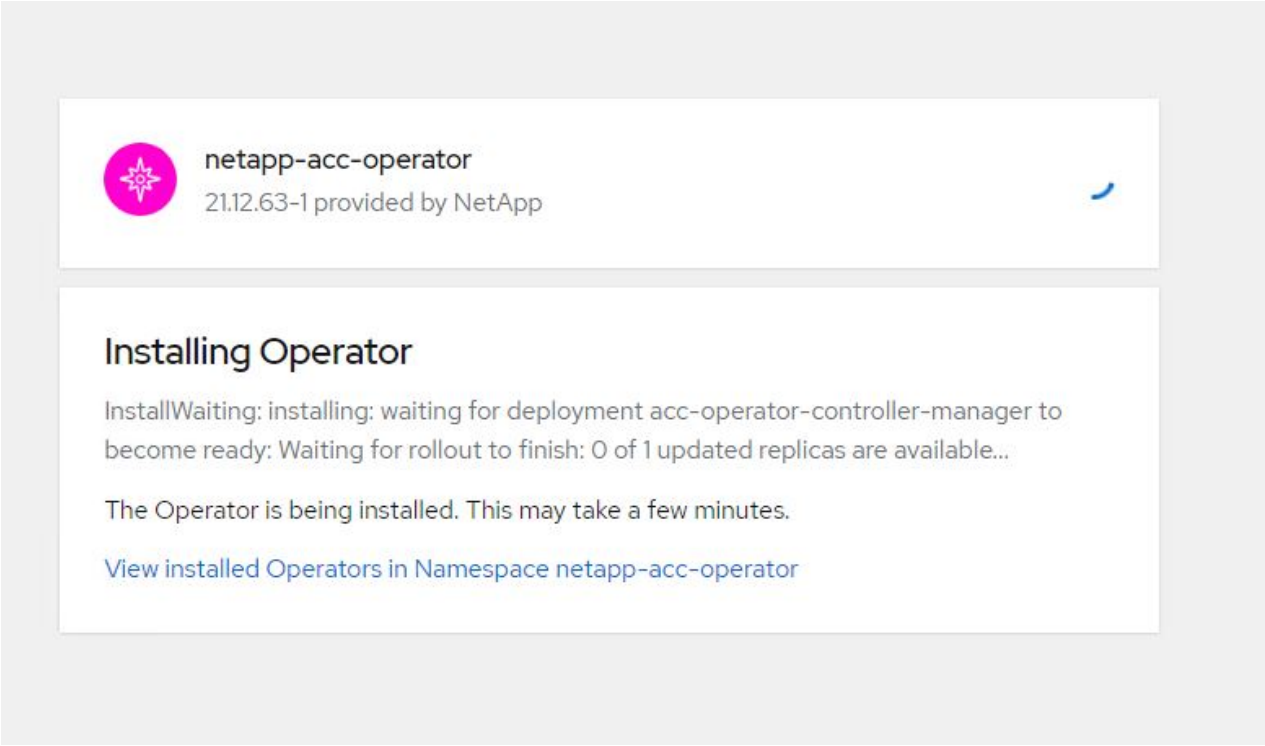
10. Wählen Sie `netapp-acc-operator` kachel und klicken Sie auf `Install`.



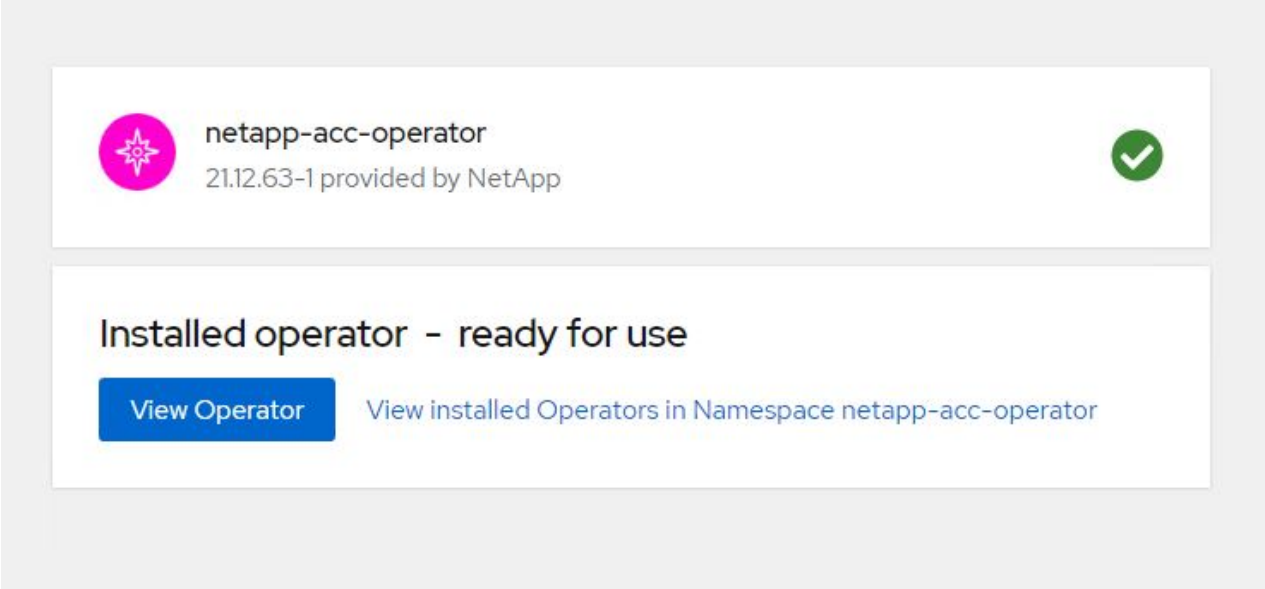
11. Übernehmen Sie im Bildschirm Operator installieren alle Standardparameter, und klicken Sie auf `Install`.



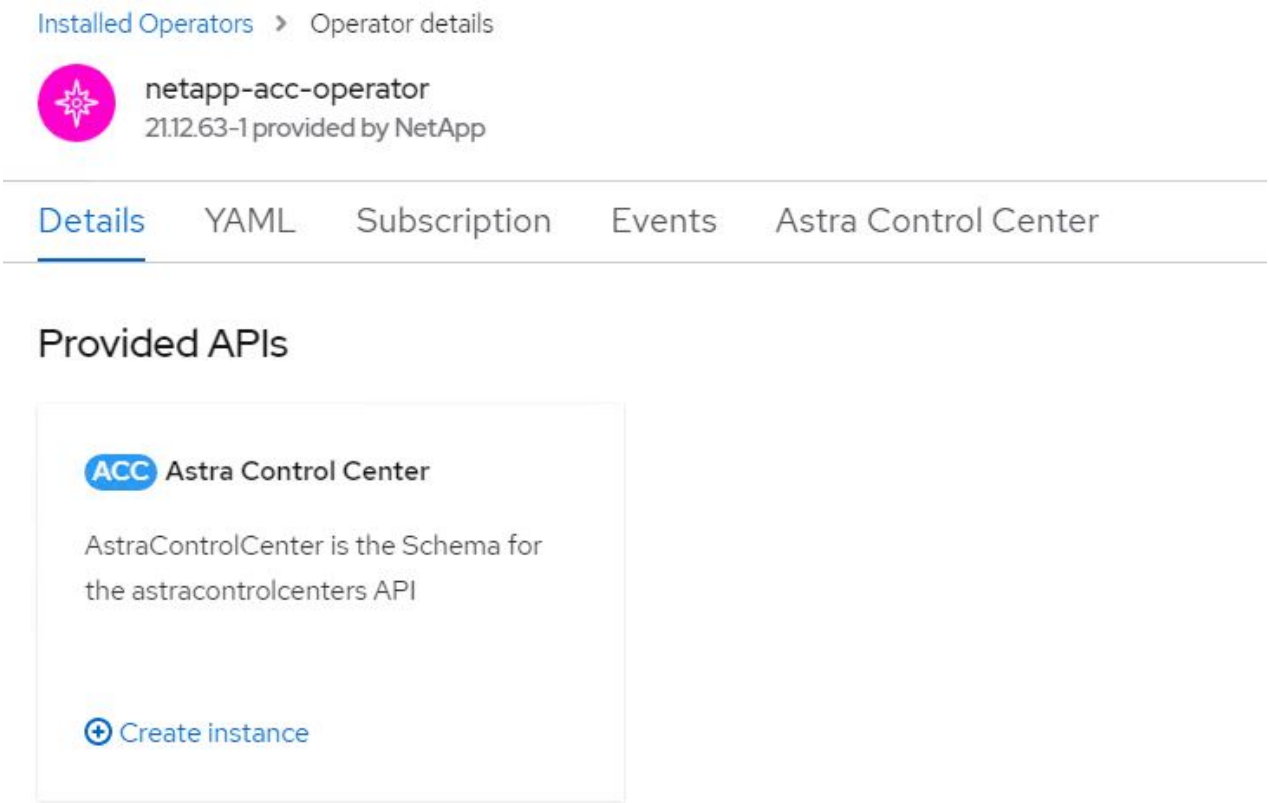
12. Warten Sie, bis die Installation des Bedieners abgeschlossen ist.



13. Sobald die Installation des Bedieners erfolgreich abgeschlossen ist, navigieren Sie zu, um auf zu klicken `View Operator`.



14. Klicken Sie dann auf `Create Instance` Im Astra Control Center Kachel im Operator.



15. Füllen Sie die aus `Create AstraControlCenter` Formularfelder und klicken Sie auf `Create`.

    a. Bearbeiten Sie optional den Instanznamen des Astra Control Center.
    b. Aktivieren oder deaktivieren Sie optional Auto Support. Es wird empfohlen, die Auto Support-Funktion beizubehalten.
    c. Geben Sie den FQDN für Astra Control Center ein.
    d. Geben Sie die Astra Control Center-Version ein. Die neueste wird standardmäßig angezeigt.

e. Geben Sie einen Kontonamen für das Astra Control Center und die Administratordetails wie Vorname, Nachname und E-Mail-Adresse ein.

f. Geben Sie die Richtlinie zur Rückgewinnung von Volumes ein. Die Standardeinstellung wird beibehalten.

g. Geben Sie in der Bildregistrierung den FQDN für Ihre Registrierung zusammen mit dem Namen der Organisation ein, den Sie erhalten haben, während Sie die Bilder in die Registrierung schieben (in diesem Beispiel `astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra`)

h. Wenn Sie eine Registrierung verwenden, für die eine Authentifizierung erforderlich ist, geben Sie den geheimen Namen im Abschnitt Image Registry ein.

i. Konfigurieren Sie Skalierungsoptionen für Astra Control Center Ressourceneinschränkungen.

j. Geben Sie den Namen der Speicherklasse ein, wenn PVCs in eine nicht-Standardspeicherklasse platziert werden sollen.

k. Definieren Sie die Einstellungen für die Verarbeitung von CRD.

Project: netapp-acc-operator

Name *

astra

Labels

app=frontend

Account Name *

HCG Solutions Engineering

Astra Control Center account name

Astra Address *

astra-control-center.cie.netapp.com

AstraAddress defines how Astra will be found in the data center. This IP address and/or DNS A record must be created prior to provisioning Astra Control Center. Example - "astra.example.com" The A record and its IP address must be allocated prior to provisioning Astra Control Center

Astra Version *

21.12.60

Version of AstraControlCenter to deploy. You are provided a Helm repository with a corresponding version. Example - 1.5.2, 1.4.2-patch

Email *

solutions_tme@netapp.com

EmailAddress will be notified by Astra as events warrant.

Auto Support *

AutoSupport indicates willingness to participate in NetApp's proactive support application, NetApp Active IQ. The default election is true and indicates support data will be sent to NetApp. An empty or blank election is the same as a default election. Air gapped installations should enter false.

First Name

HCG

The first name of the SRE supporting Astra.

Last Name

Admin

The last name of the SRE supporting Astra.

Image Registry

The container image registry that is hosting the Astra application images, ACC Operator and ACC Helm Repository.

Name

astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra

The name of the image registry. For example "example.registry/astra". Do not prefix with protocol.

Secret

astra-registry-cred

The name of the Kubernetes secret that will authenticate with the image registry.

Volume Reclaim Policy

Retain

Reclaim policy to be set for persistent volumes

Astra Resources Scaler

Default

Scaling options for AstraControlCenter Resource limits.

Storage Class

The storage class to be used for PVCs. If not set, default storage class will be used.

Crds

Options for how ACC should handle CRDs.

Create Cancel

Automatisiert [Ansible]

1. Um Astra Control Center mit Ansible-Playbooks zu implementieren, benötigen Sie eine Ubuntu/RHEL-Maschine mit installiertem Ansible. Befolgen Sie die Anweisungen "Hier" Für Ubuntu und RHEL.
2. Klonen Sie das GitHub Repository, das Ansible-Inhalte hostet.

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

3. Melden Sie sich bei der NetApp Support-Website an und laden Sie die neueste Version des NetApp Astra Control Center herunter. Dazu ist eine Lizenz erforderlich, die an Ihr NetApp Konto angehängt ist. Nach dem Download des Tarballs, übertragen Sie es auf die Workstation.

> Um mit einer Testlizenz für Astra Control zu beginnen, besuchen Sie die "Astra: Anmelde-Website".

4. Erstellen oder beziehen Sie die kubeconfig-Datei mit Administratorzugriff auf das OpenShift-Cluster, auf dem Astra Control Center installiert werden soll.

5. Ändern Sie das Verzeichnis in die na_astra_control_Suite.

```
cd na_astra_control_suite
```

6. Bearbeiten Sie das `vars/vars.yml` Datei, und füllen Sie die Variablen mit den erforderlichen Informationen.

```
#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. Nutzen Sie das Playbook zur Implementierung des Astra Control Center. Für bestimmte Konfigurationen sind Root-Berechtigungen erforderlich.

   Wenn der Benutzer, der das Playbook ausführt, root ist oder eine passwortlose sudo-Konfiguration hat, führen Sie den folgenden Befehl aus, um das Playbook auszuführen.

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

   Wenn der Benutzer passwortbasierten sudo-Zugriff konfiguriert hat, führen Sie den folgenden Befehl aus, um das Playbook auszuführen, und geben Sie dann das sudo-Passwort ein.

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

Schritte Nach Der Installation

1. Die Installation kann einige Minuten dauern. Überprüfen Sie, ob alle Pods und Services im enthalten sind `netapp-astra-cc` Der Namespace ist betriebsbereit.

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

2. Prüfen Sie die `acc-operator-controller-manager` Protokolle, um sicherzustellen, dass die Installation abgeschlossen ist.

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

Die folgende Meldung zeigt die erfolgreiche Installation des Astra Control Centers an.

```
{"level":"info","ts":1624054318.029971,"logger":"controllers.AstraContro
lCenter","msg":"Successfully Reconciled AstraControlCenter in
[seconds]s","AstraControlCenter":"netapp-astra-
cc/astra","ae.Version":"[21.12.60]"}
```

3. Der Benutzername für die Anmeldung beim Astra Control Center ist die E-Mail-Adresse des Administrators in der CRD-Datei und das Passwort ist eine Zeichenfolge `ACC-` An die Astra Control Center UUID angehängt. Führen Sie den folgenden Befehl aus:

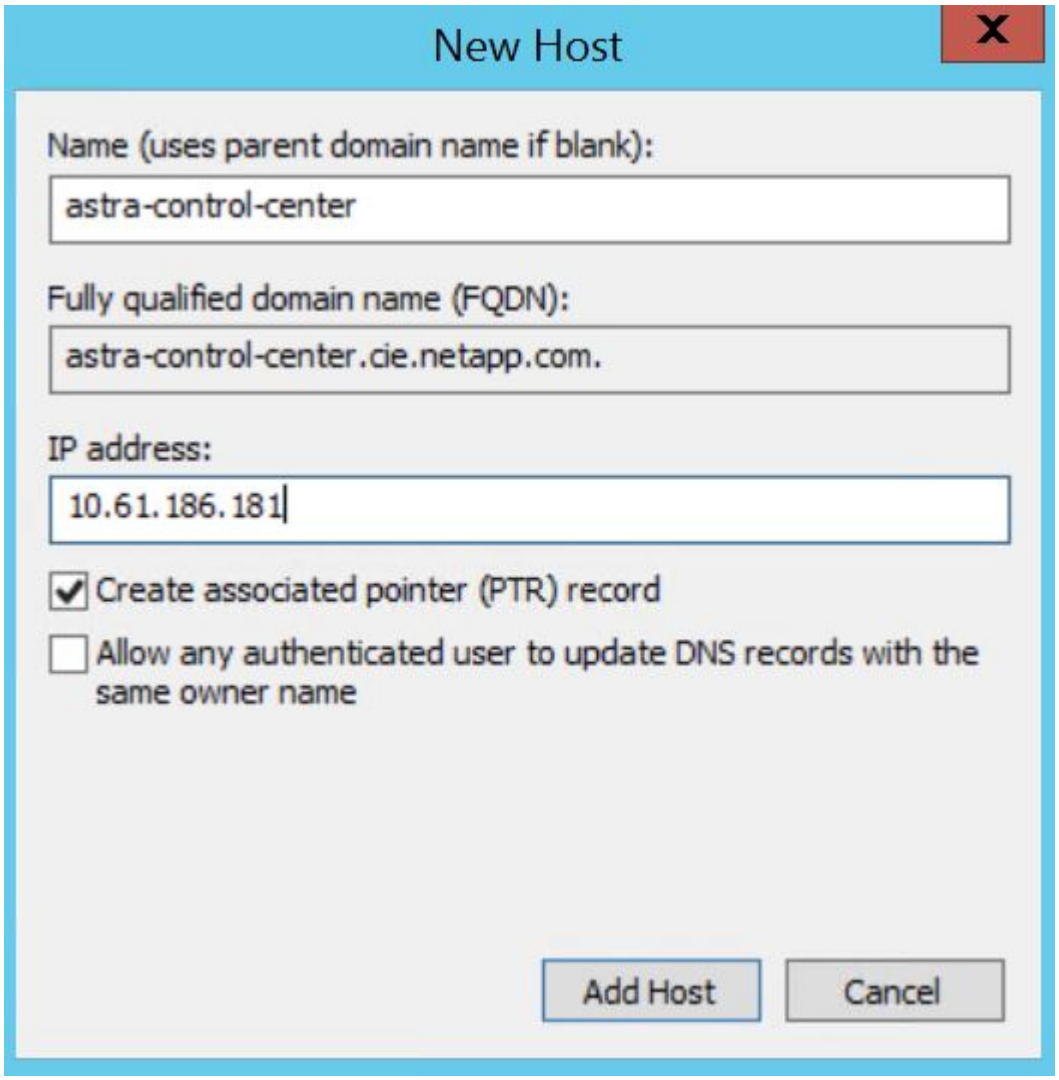
```
[netapp-user@rhel7 ~]$ oc get astracontrolcenters -n netapp-astra-cc
NAME    UUID
astra   345c55a5-bf2e-21f0-84b8-b6f2bce5e95f
```

In diesem Beispiel lautet das Passwort `ACC-345c55a5-bf2e-21f0-84b8-b6f2bce5e95f`.

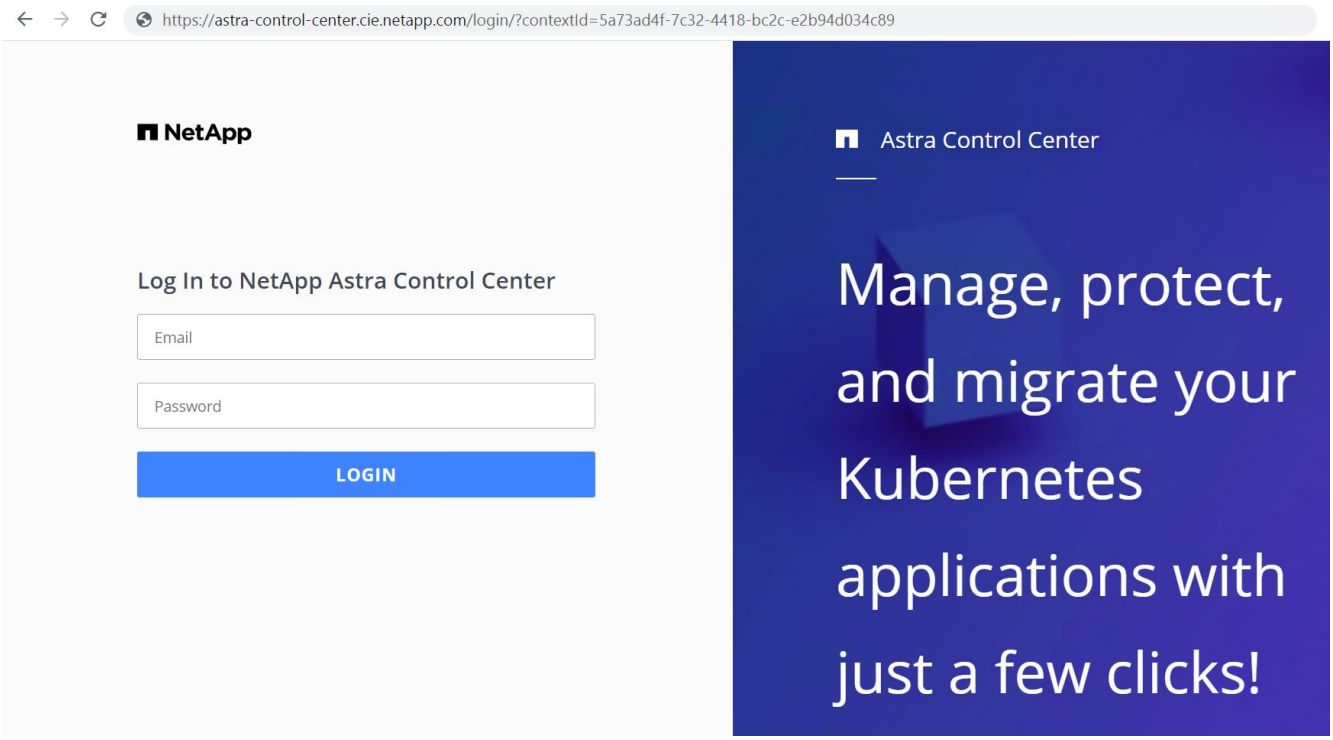
4. Holen Sie die LastausgleichsIP für den Traefik-Dienst ab.

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

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

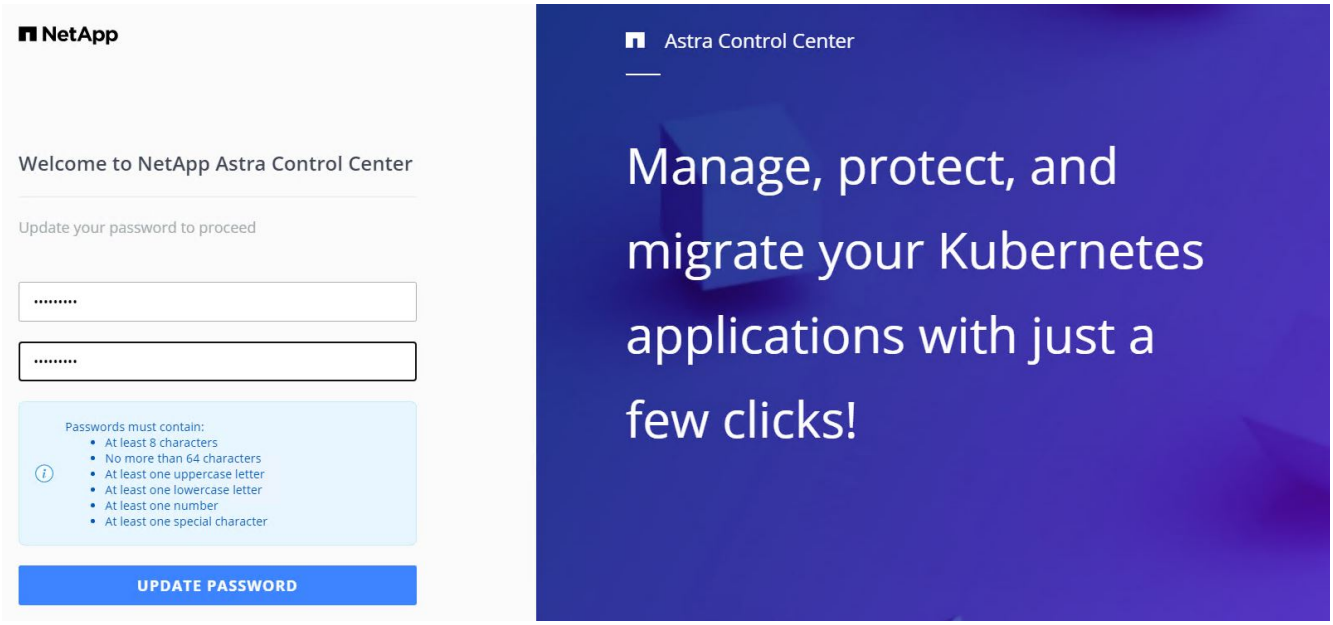
5. Fügen Sie einen Eintrag im DNS-Server hinzu, der auf den in der Astra Control Center CRD-Datei angegebenen FQDN verweist `EXTERNAL-IP` Des Schleppdienstes.



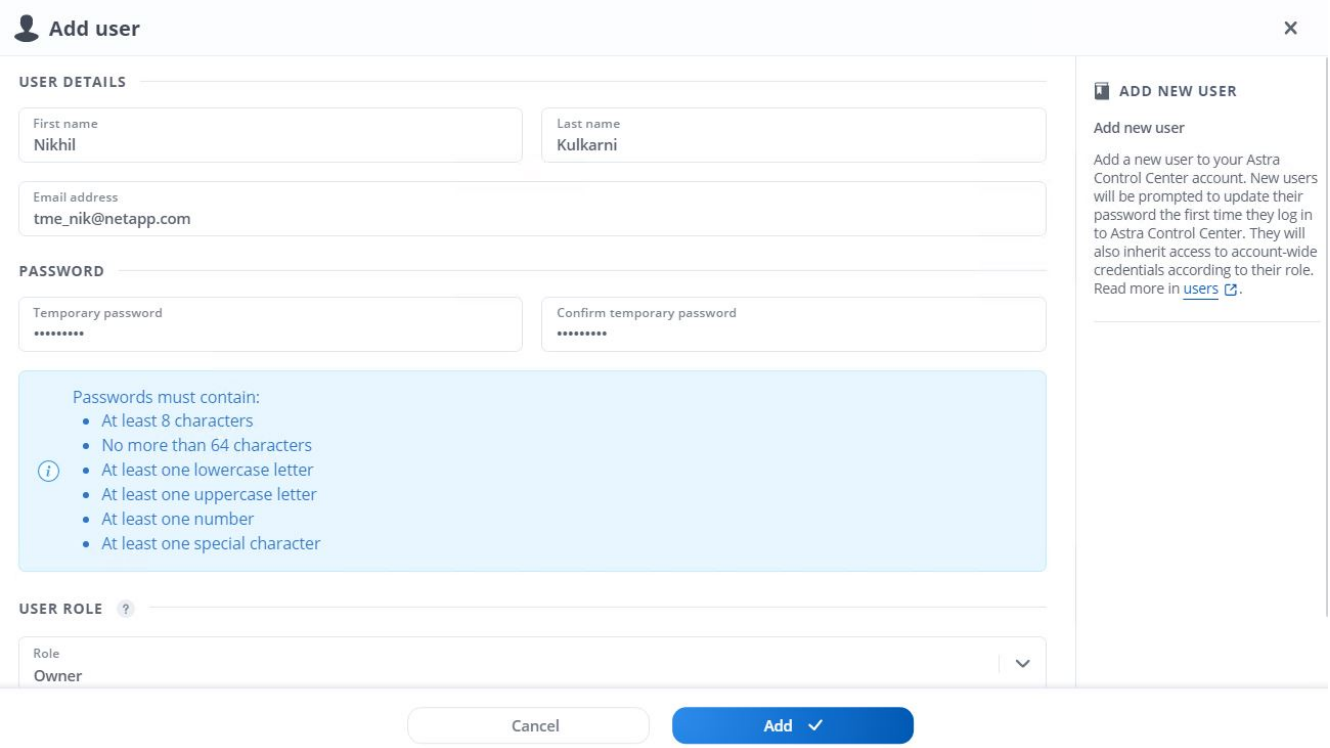
6. Melden Sie sich bei der Astra Control Center-GUI an, indem Sie den FQDN durchsuchen.



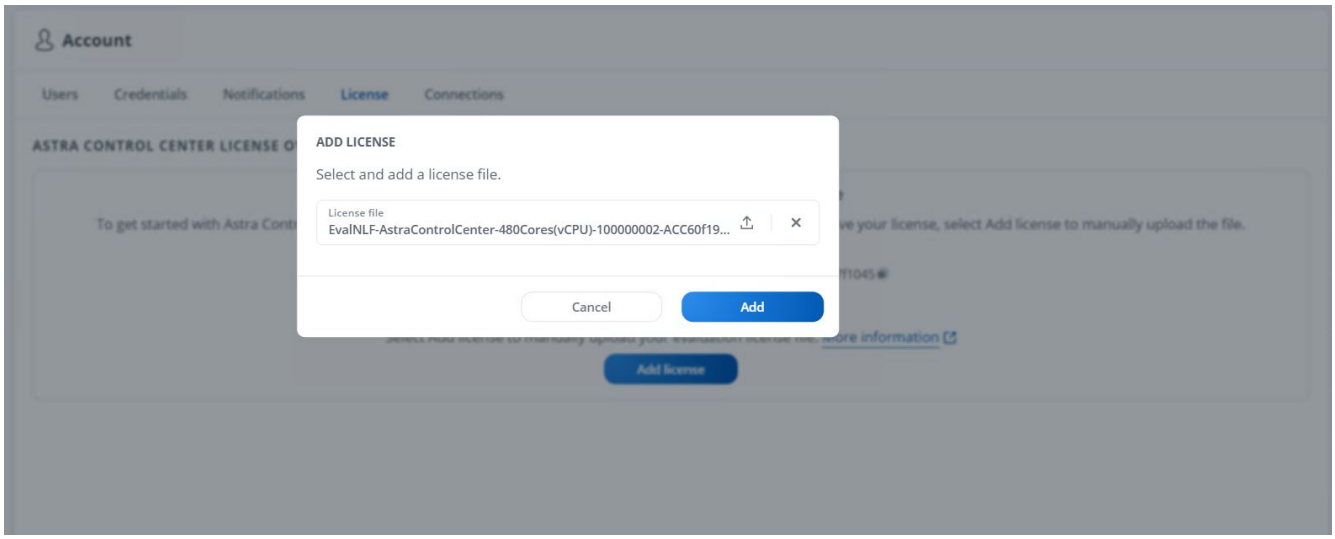
7. Wenn Sie sich zum ersten Mal über die in CRD angegebene Admin-E-Mail-Adresse bei der Benutzeroberfläche des Astra Control Center anmelden, müssen Sie das Passwort ändern.



8. Wenn Sie dem Astra Control Center einen Benutzer hinzufügen möchten, navigieren Sie zu Konto > Benutzer, klicken Sie auf Hinzufügen, geben Sie die Details des Benutzers ein und klicken Sie auf Hinzufügen.



9. Astra Control Center erfordert eine Lizenz, damit alle Funktionalitäten der IT funktionieren können. Um eine Lizenz hinzuzufügen, navigieren Sie zu Konto > Lizenz, klicken Sie auf Lizenz hinzufügen und laden Sie die Lizenzdatei hoch.



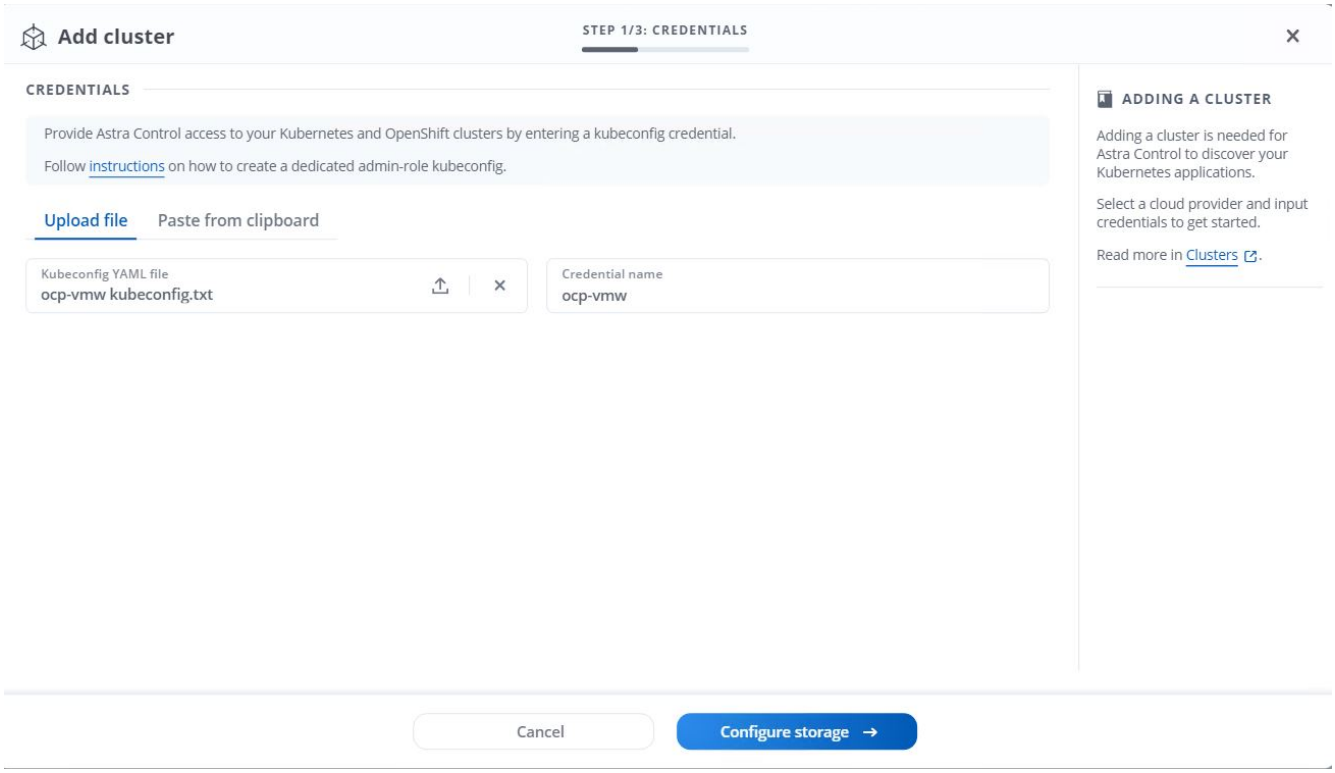
Bei Problemen mit der Installation oder Konfiguration von NetApp Astra Control Center steht die Wissensdatenbank mit bekannten Problemen zur Verfügung "Hier".

Registrieren Sie Ihre Red hat OpenShift-Cluster mit dem Astra Control Center

Damit das Astra Control Center Ihre Workloads managen kann, müssen Sie zunächst Ihren Red hat OpenShift-Cluster registrieren.

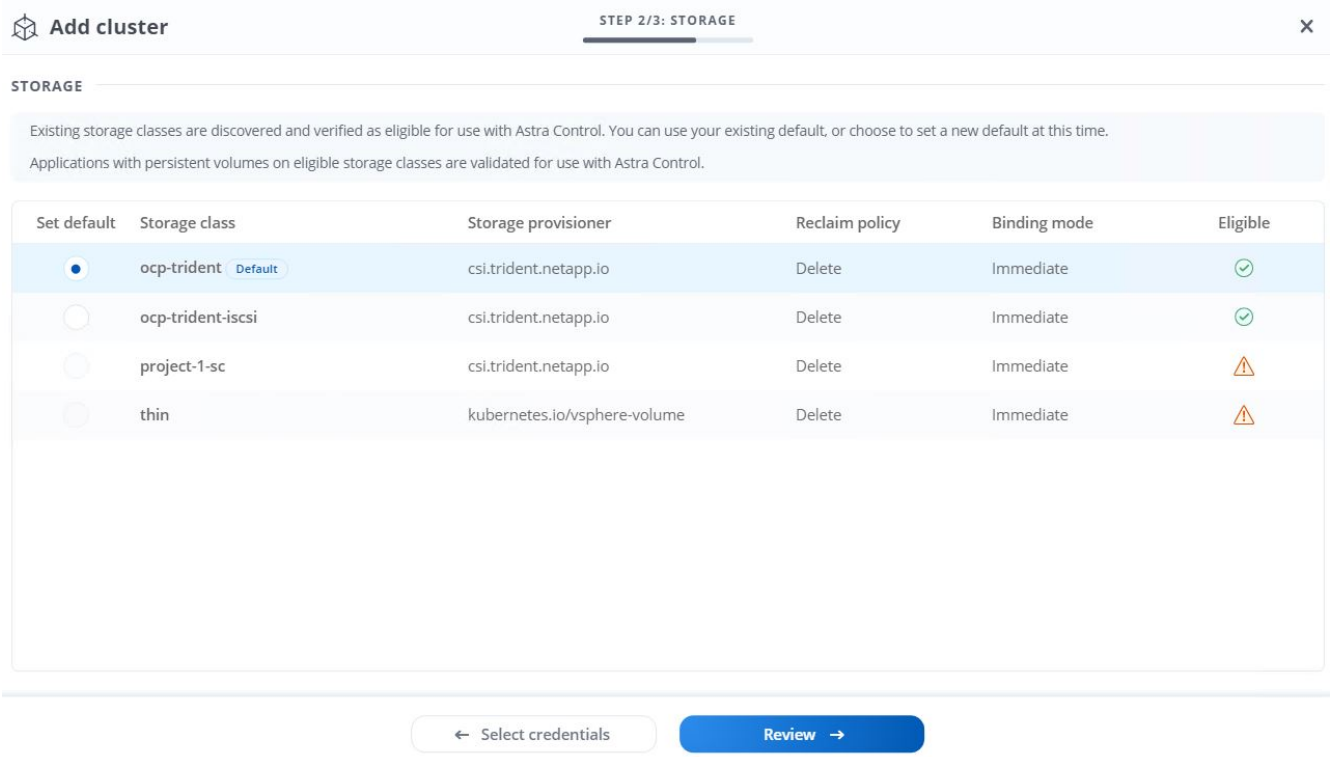
Red hat OpenShift-Cluster registrieren

1. Der erste Schritt besteht darin, die OpenShift Cluster zum Astra Control Center hinzuzufügen und zu verwalten. Wechseln Sie zu Cluster und klicken Sie auf Cluster hinzufügen, laden Sie die kubeconffig-Datei für den OpenShift-Cluster hoch, und klicken Sie auf Storage auswählen.

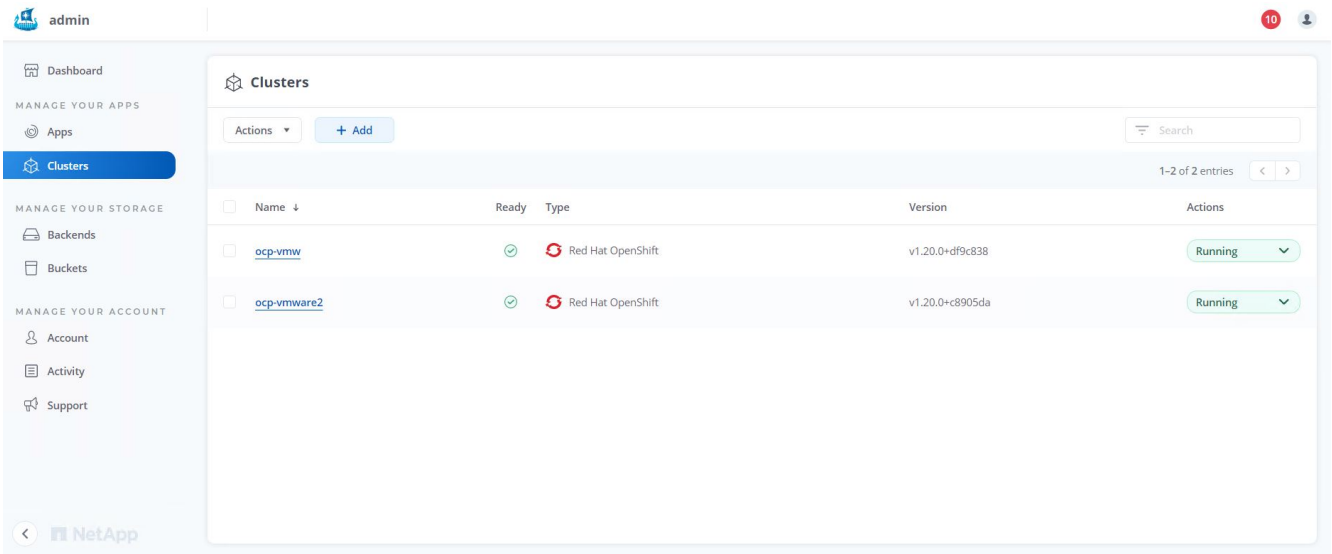


> Die kubeconfig-Datei kann zur Authentifizierung mit einem Benutzernamen und Passwort oder einem Token erzeugt werden. Token laufen nach begrenzter Zeit ab und lassen das registrierte Cluster möglicherweise nicht mehr erreichbar. NetApp empfiehlt, eine kubeconfig-Datei mit einem Benutzernamen und einem Passwort zu verwenden, um Ihre OpenShift-Cluster beim Astra Control Center zu registrieren.

2. Astra Control Center erkennt geeignete Storage-Klassen. Wählen Sie jetzt aus, wie Storage Volumes mithilfe von Trident durch eine SVM auf NetApp ONTAP bereitgestellt werden, und klicken Sie auf „Review“ (prüfen). Überprüfen Sie im nächsten Teilfenster die Details, und klicken Sie auf Cluster hinzufügen.

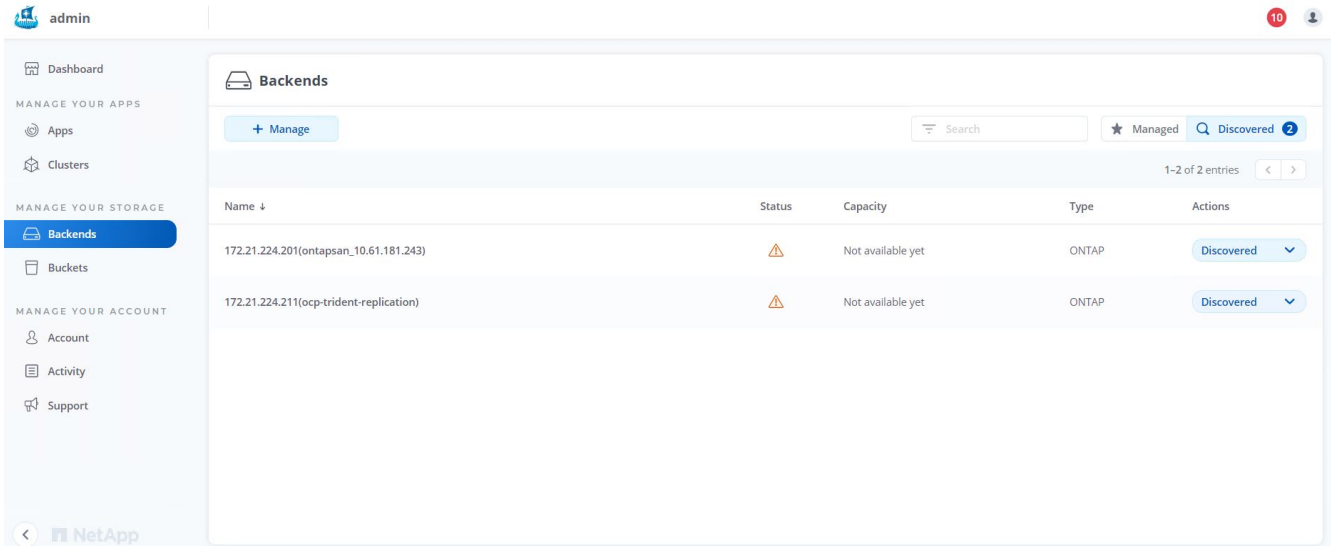


3. Registrieren Sie beide OpenShift-Cluster wie in Schritt 1 beschrieben. Wenn sie hinzugefügt werden, wechseln die Cluster zum Status Erkennung, während das Astra Control Center sie überprüft und die erforderlichen Agenten installiert. Der Cluster-Status ändert sich auf „ausgeführt“, nachdem sie erfolgreich registriert wurden.

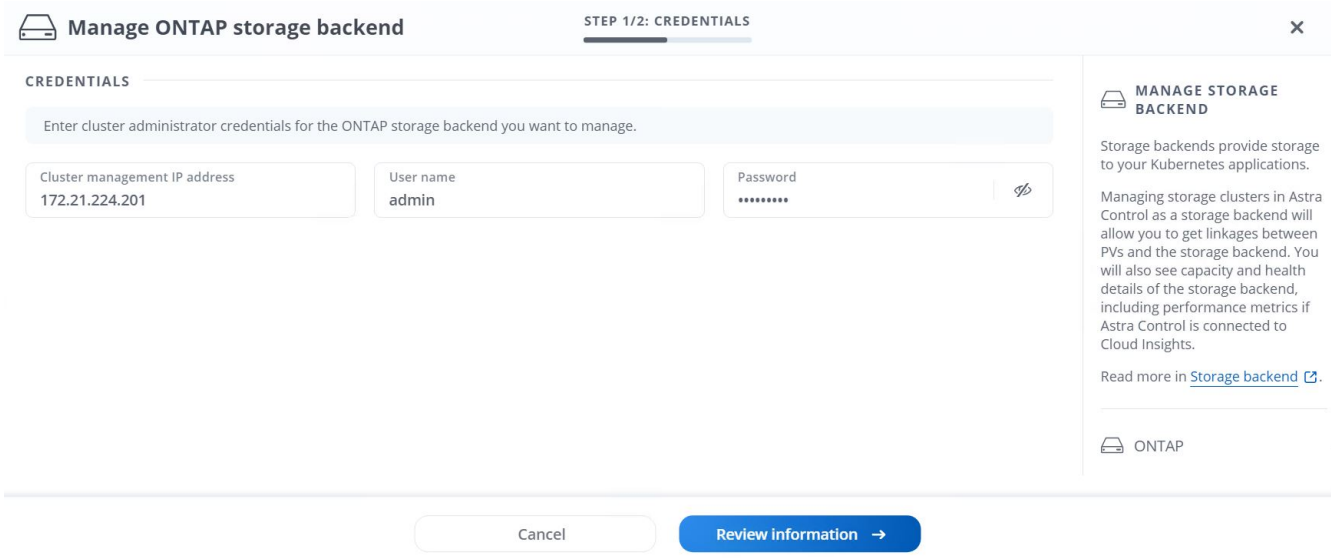


> Alle von Astra Control Center zu verwaltenden Red hat OpenShift-Cluster sollten Zugriff auf die Bildregistrierung haben, die für die Installation verwendet wurde, da die auf den verwalteten Clustern installierten Agenten die Bilder aus dieser Registrierung ziehen.

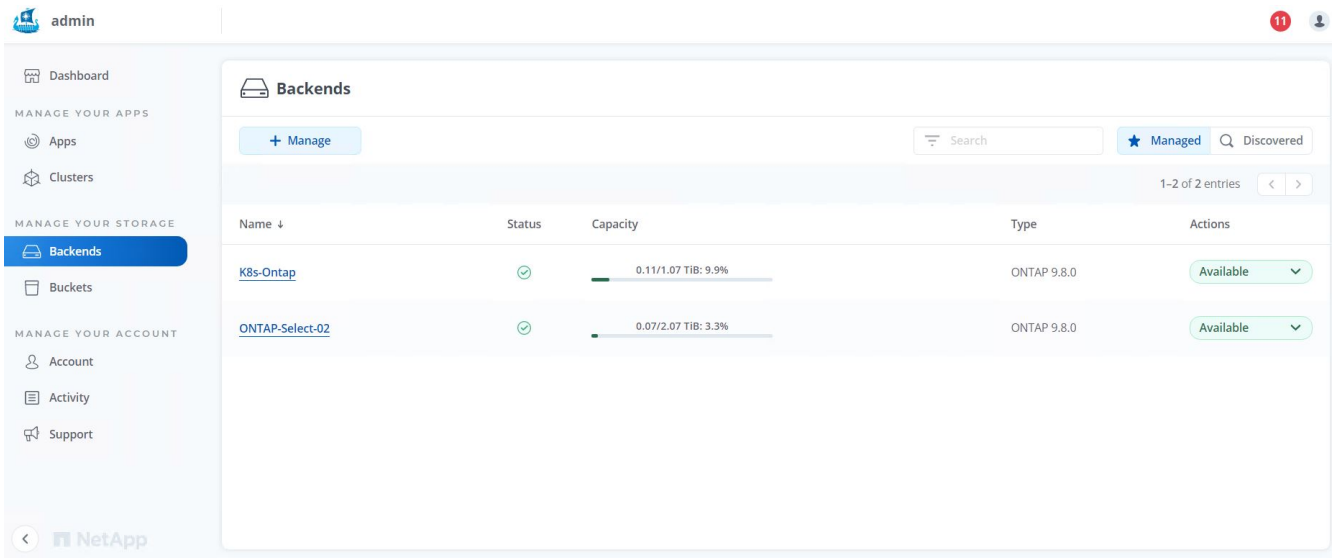
4. Importieren Sie ONTAP-Cluster als Storage-Ressourcen, die vom Astra Control Center als Back-Ends gemanagt werden sollen. Wenn dem Astra OpenShift-Cluster hinzugefügt und ein Storage-glass konfiguriert ist, erkennt und inspiziert er den ONTAP-Cluster automatisch auf der Basis der Storage-glass, importiert ihn aber nicht in das zu verwaltende Astra-Control-Center.



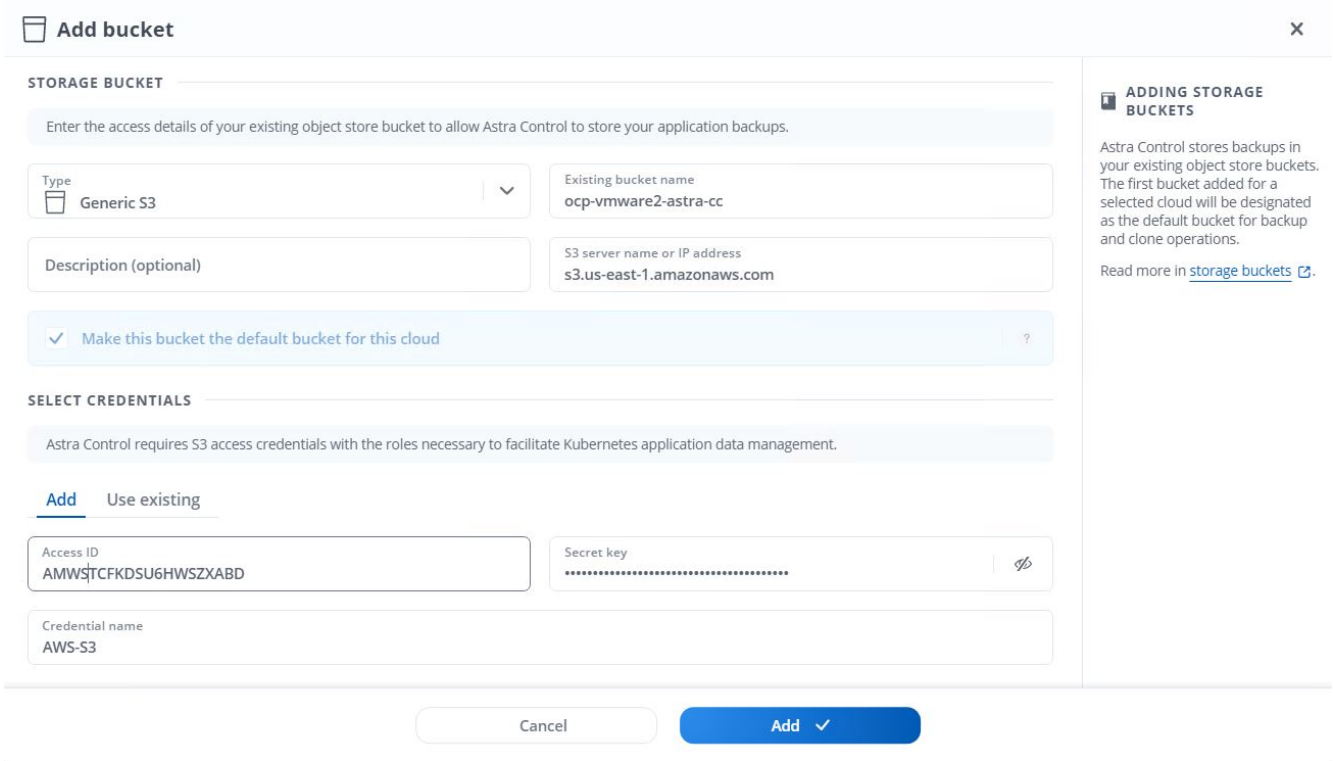
5. Um die ONTAP-Cluster zu importieren, gehen Sie zu Backend, klicken Sie auf das Dropdown-Menü und wählen Sie Verwalten neben dem zu verwaltenden ONTAP-Cluster aus. Geben Sie die ONTAP-Cluster-Anmeldeinformationen ein, klicken Sie auf Informationen überprüfen und klicken Sie dann auf Speicher-Backend importieren.



6. Nach dem Hinzufügen der Back-Ends ändert sich der Status in „verfügbar“. Diese Back-Ends enthalten jetzt Informationen über die persistenten Volumes im OpenShift-Cluster und die entsprechenden Volumes im ONTAP-System.



7. Für Backups und Restores in OpenShift-Clustern mit Astra Control Center müssen Sie einen Objekt-Storage-Bucket bereitstellen, der das S3-Protokoll unterstützt. Aktuell werden ONTAP S3, StorageGRID und AWS S3 unterstützt. Im Rahmen dieser Installation wird ein AWS S3-Bucket konfiguriert. Wechseln Sie zu Buckets, klicken Sie auf „Bucket hinzufügen“ und wählen Sie „Allgemeines S3“ aus. Geben Sie die Details zum S3-Bucket ein und erhalten Sie Zugangsdaten, um darauf zuzugreifen. Aktivieren Sie das Kontrollkästchen „Machen Sie diesen Bucket zum Standard-Bucket für die Cloud“ und klicken Sie dann auf Hinzufügen.

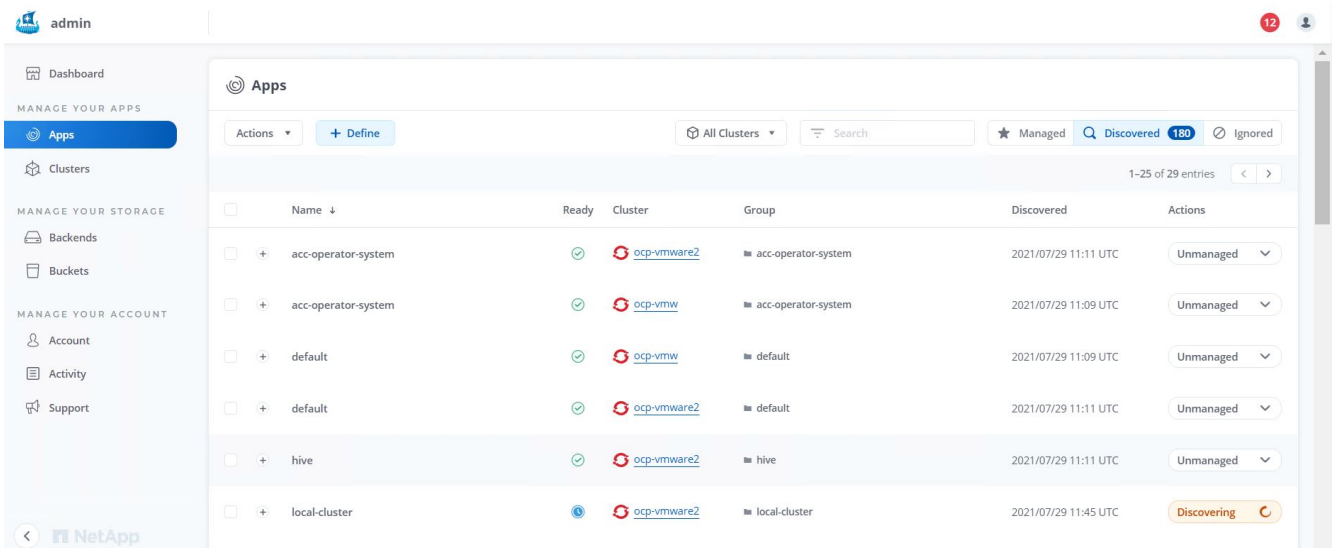


Wählen Sie die zu schützenden Applikationen aus

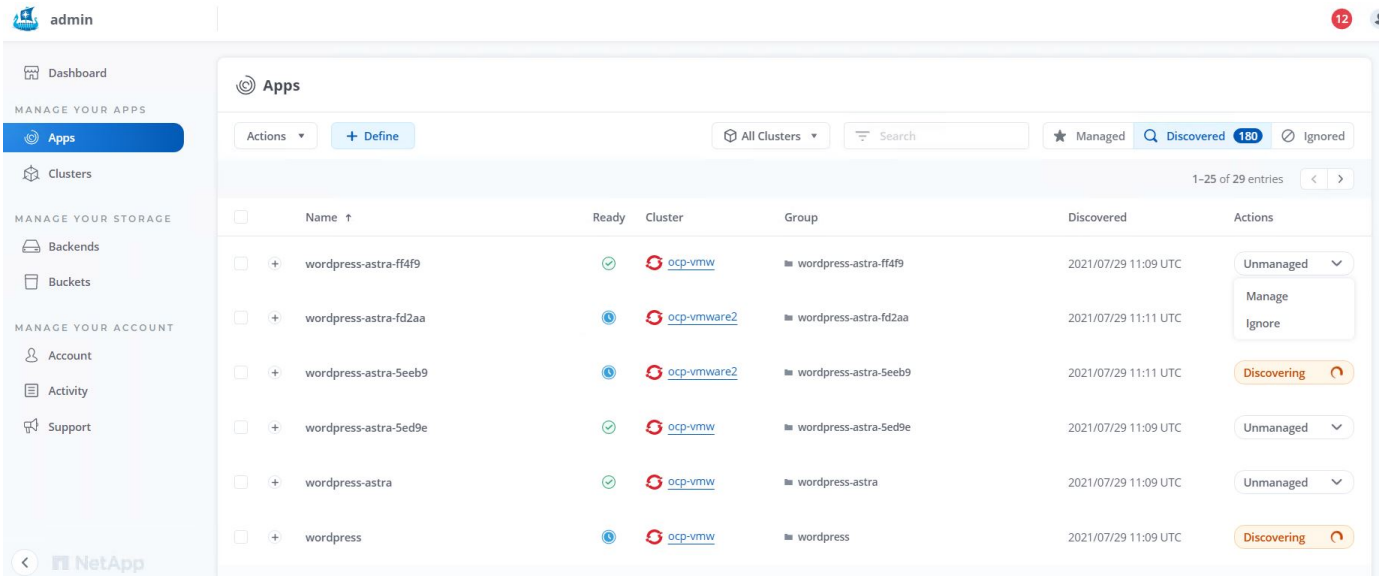
Nachdem Sie Ihre Red hat OpenShift-Cluster registriert haben, können Sie über das Astra Control Center die Anwendungen ermitteln, die bereitgestellt und verwaltet werden.

Management von Applikationen

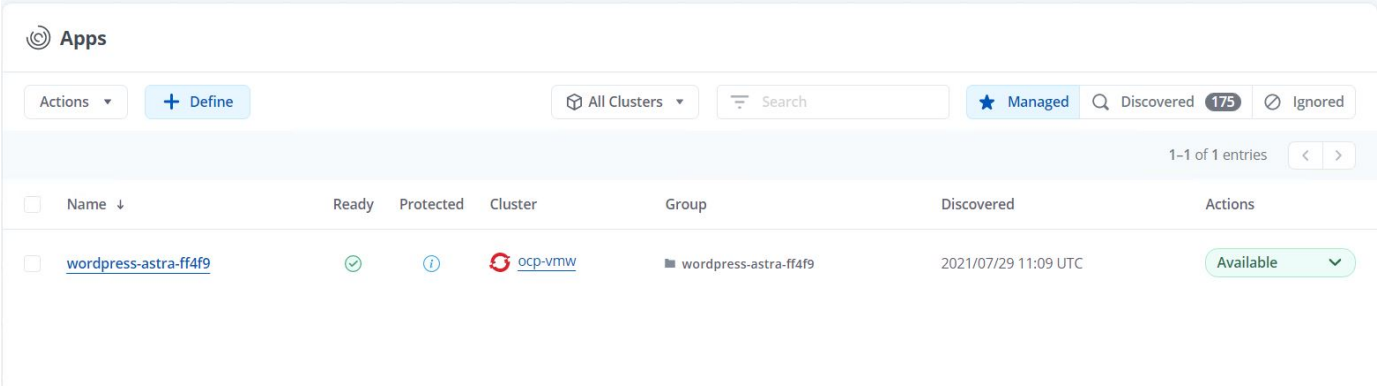
1. Nachdem die OpenShift-Cluster und ONTAP-Back-Ends beim Astra Control Center registriert wurden, beginnt das Kontrollzentrum automatisch die Anwendungen in allen Namespaces zu erkennen, die die mit dem angegebenen ONTAP-Backend konfigurierte Speicherageclass verwenden.



2. Navigieren Sie zu Apps > entdeckt, und klicken Sie auf das Dropdown-Menü neben der Anwendung, die Sie mit Astra verwalten möchten. Klicken Sie dann auf Verwalten.



1. Die Anwendung wechselt in den Status „verfügbar“ und kann im Abschnitt „Apps“ unter der Registerkarte „verwaltet“ angezeigt werden.



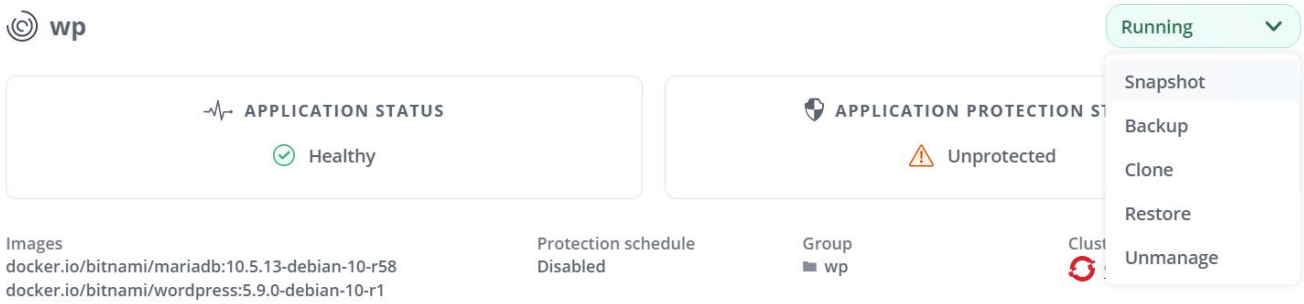
Sichern Sie Ihre Applikationen

Nachdem die Applikations-Workloads vom Astra Control Center gemanagt wurden, können Sie die Sicherungseinstellungen für diese Workloads konfigurieren.

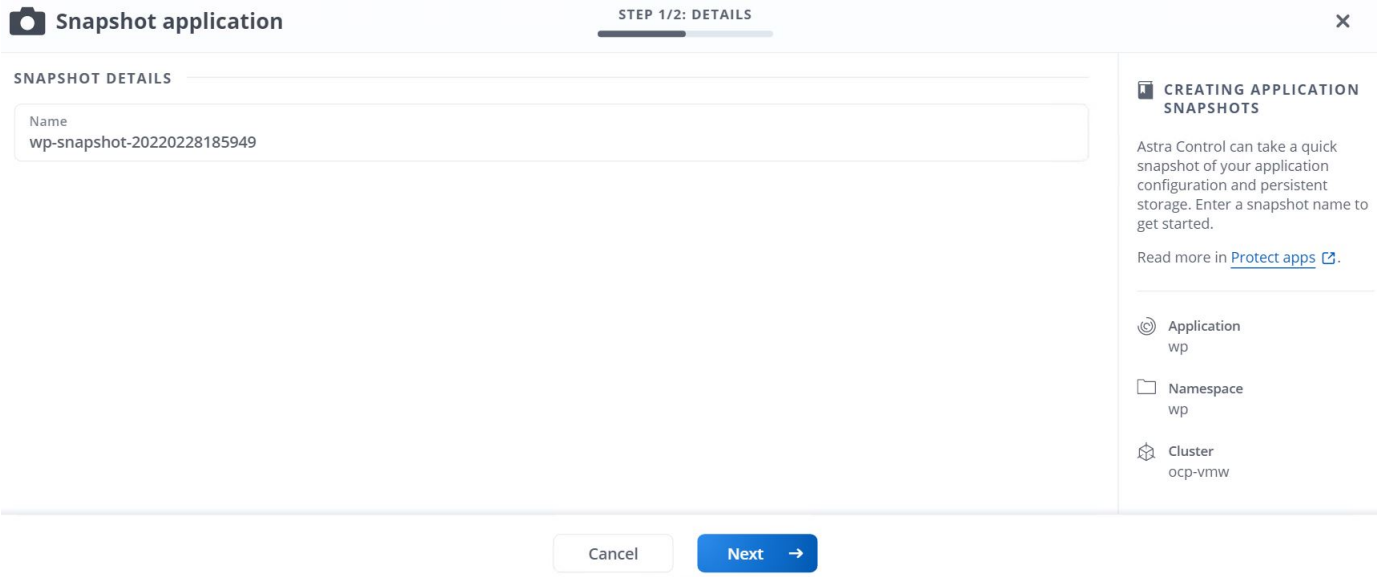
Erstellen eines Anwendungs-Snapshots

Ein Snapshot einer Applikation erstellt eine ONTAP Snapshot Kopie, mit der die Applikation auf Basis dieser Snapshot Kopie einem bestimmten Zeitpunkt wiederhergestellt oder geklont werden kann.

1. Um einen Snapshot der Anwendung zu erstellen, navigieren Sie zur Registerkarte Apps > Managed und klicken Sie auf die Anwendung, von der Sie eine Snapshot Kopie erstellen möchten. Klicken Sie auf das Dropdown-Menü neben dem Anwendungsnamen, und klicken Sie auf Snapshot.



2. Geben Sie die Snapshot-Details ein, klicken Sie auf Weiter und klicken Sie dann auf Snapshot. Es dauert etwa eine Minute, um den Snapshot zu erstellen, und der Status wird verfügbar, nachdem der Snapshot erfolgreich erstellt wurde.



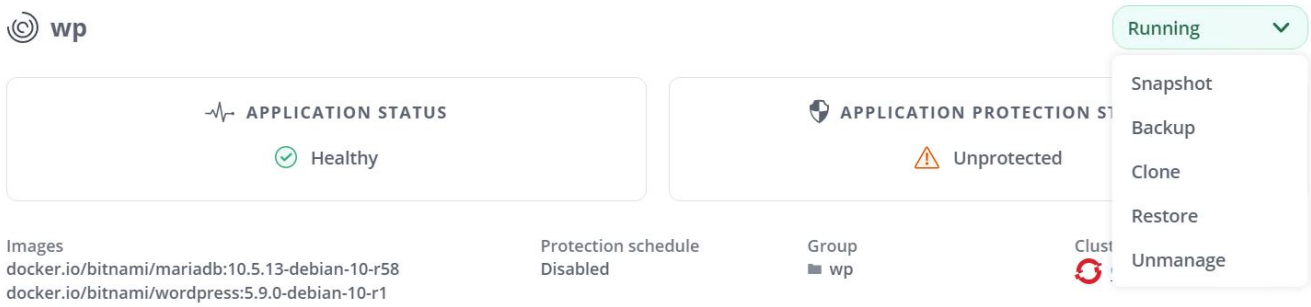
Erstellen eines Applikations-Backups

Ein Backup einer Applikation erfasst den aktiven Status der Applikation und die Konfiguration der Ressourcen des IT-Systems, deckt sie in Dateien ab und speichert sie in einem Remote-Objekt-Storage-Bucket.

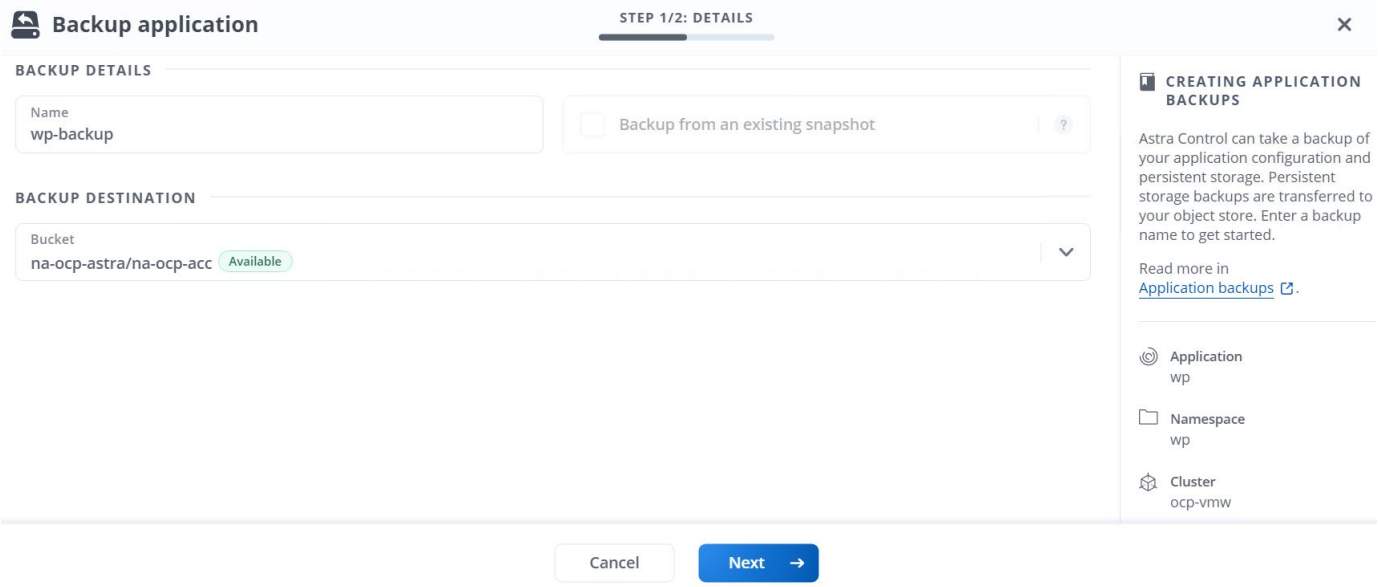
Für die Sicherung und Wiederherstellung von verwalteten Anwendungen im Astra Control Center müssen Sie die Superuser-Einstellungen für die ONTAP-Systeme als Voraussetzung konfigurieren. Geben Sie dazu die folgenden Befehle ein.

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

1. Um ein Backup der verwalteten Anwendung im Astra Control Center zu erstellen, navigieren Sie zur Registerkarte Apps > Managed und klicken Sie auf die Anwendung, von der Sie ein Backup durchführen möchten. Klicken Sie auf das Dropdown-Menü neben dem Anwendungsnamen, und klicken Sie auf Backup.



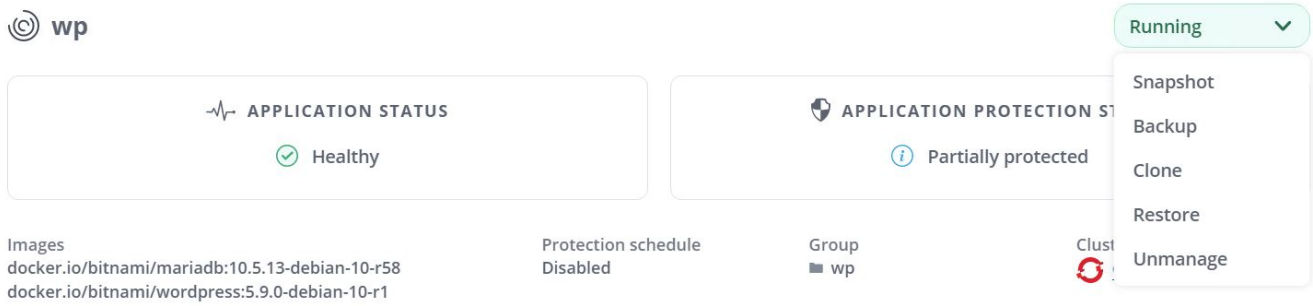
2. Geben Sie die Backup-Details ein, wählen Sie den Objekt-Storage-Bucket aus, der die Backup-Dateien enthält, klicken Sie auf Weiter und klicken Sie nach Überprüfung der Details auf Backup. Abhängig von der Größe der Applikation und den Daten kann das Backup mehrere Minuten dauern und der Status des Backups wird nach erfolgreichem Abschluss des Backups wieder verfügbar.



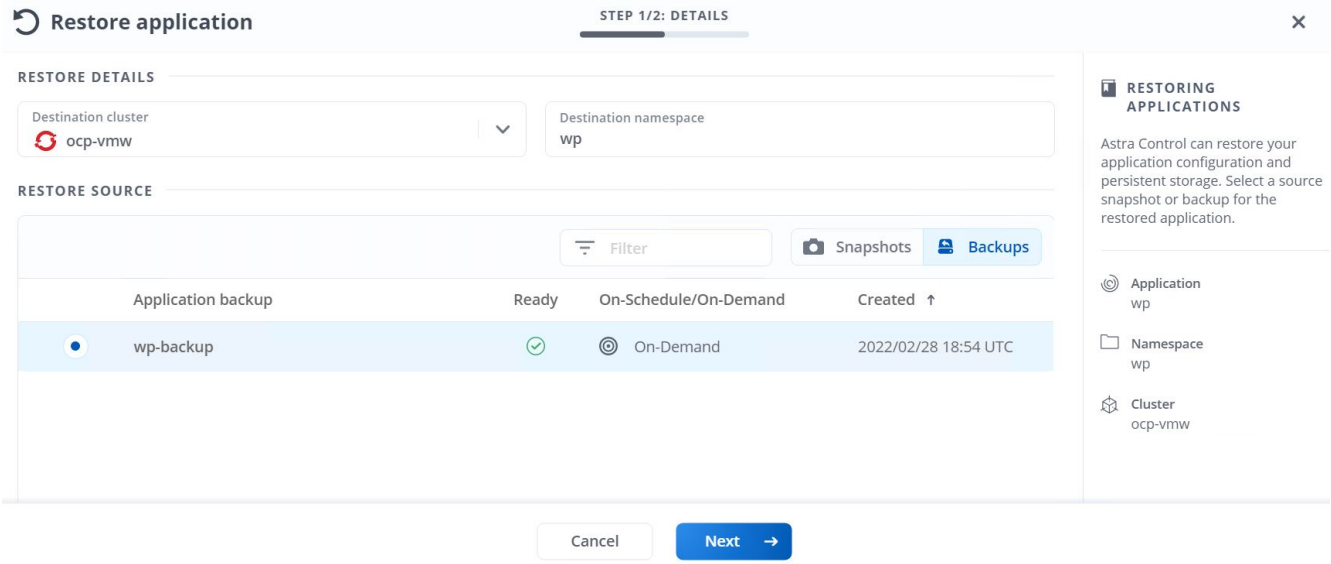
Wiederherstellen einer Anwendung

Auf Knopfdruck können Sie eine Applikation zum Zwecke der Applikationssicherung und Disaster Recovery im selben Cluster oder zu einem Remote-Cluster wiederherstellen, was den Namespace Ursprung im selben Cluster hat.

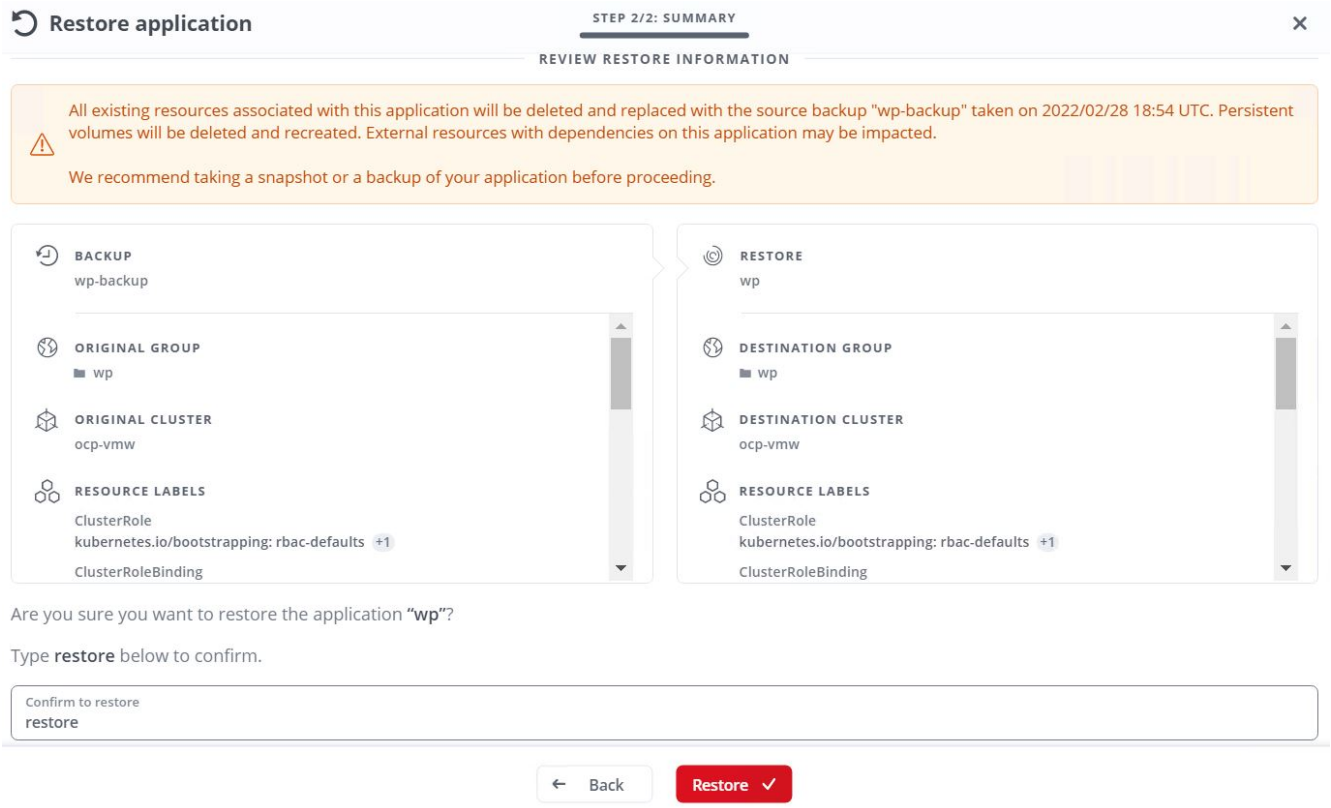
1. Um eine Anwendung wiederherzustellen, navigieren Sie zu „Apps“ > „Managed“, und klicken Sie auf die betreffende Anwendung. Klicken Sie auf das Dropdown-Menü neben dem Anwendungsnamen, und klicken Sie auf `Restore`.



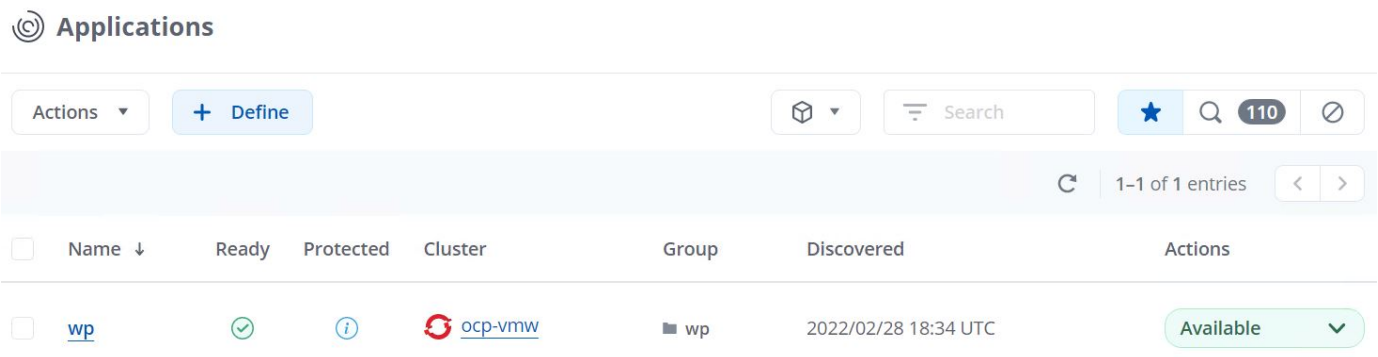
2. Geben Sie den Namen des Restore Namespace ein, wählen Sie den Cluster aus, in dem Sie ihn wiederherstellen möchten, und wählen Sie aus einem vorhandenen Snapshot oder aus einem Backup der Applikation aus, ob Sie ihn wiederherstellen möchten. Klicken Sie Auf Weiter.



3. Geben Sie im Prüfungsfenster ein `restore` Und klicken Sie auf Wiederherstellen, nachdem Sie die Details geprüft haben.



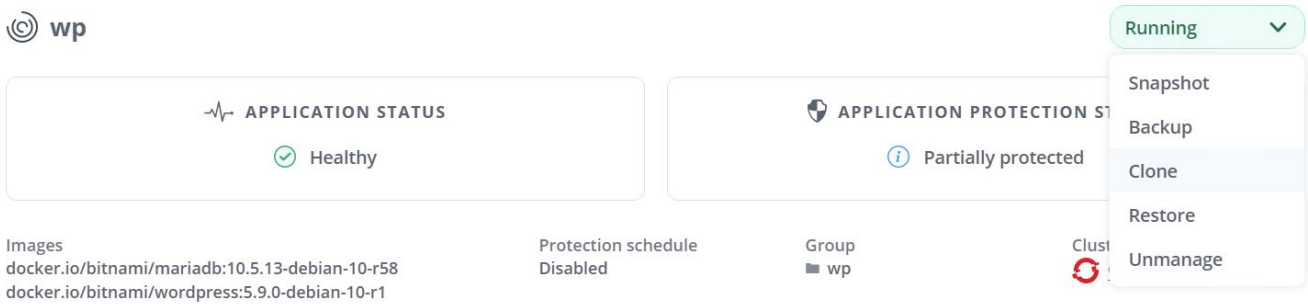
4. Die neue Applikation geht in den Status Wiederherstellen, während Astra Control Center die Anwendung auf dem ausgewählten Cluster wiederherstellt. Nachdem alle Ressourcen der Anwendung installiert und von Astra erkannt wurden, geht die Anwendung in den verfügbaren Zustand.



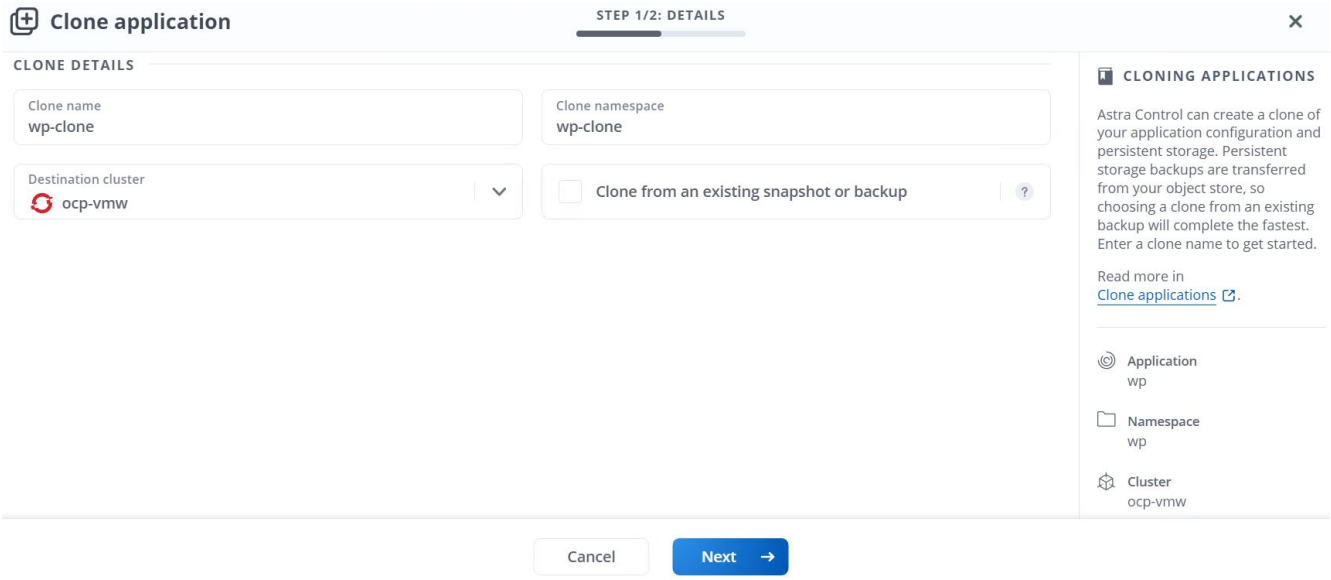
Klonen einer Applikation

Sie können eine Applikation zu Entwicklungs-/Testzwecken oder zur Sicherung von Applikationen und Disaster Recovery in einem entfernten Cluster klonen. Das Klonen einer Applikation im selben Cluster im selben Storage-Back-End nutzt die NetApp FlexClone Technologie, die VES sofort klont und Storage-Platz einspart.

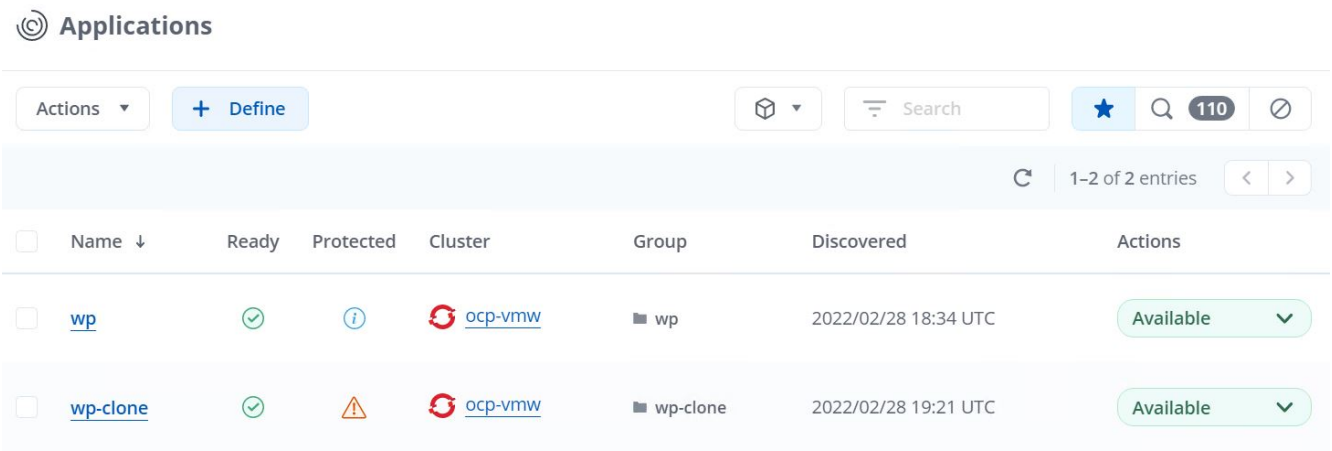
1. Um eine Anwendung zu klonen, navigieren Sie zur Registerkarte Apps > Managed und klicken Sie auf die betreffende Anwendung. Klicken Sie auf das Dropdown-Menü neben dem Anwendungsnamen und klicken Sie auf Klonen.



2. Geben Sie die Details zum neuen Namespace ein, wählen Sie den Cluster aus, in dem Sie ihn klonen möchten, und legen Sie fest, ob er aus einem vorhandenen Snapshot oder einem Backup oder dem aktuellen Status der Applikation geklont werden soll. Klicken Sie dann auf Weiter, und klicken Sie auf den Fensterbereich Klonen im Überprüfungsbereich, sobald Sie die Details geprüft haben.



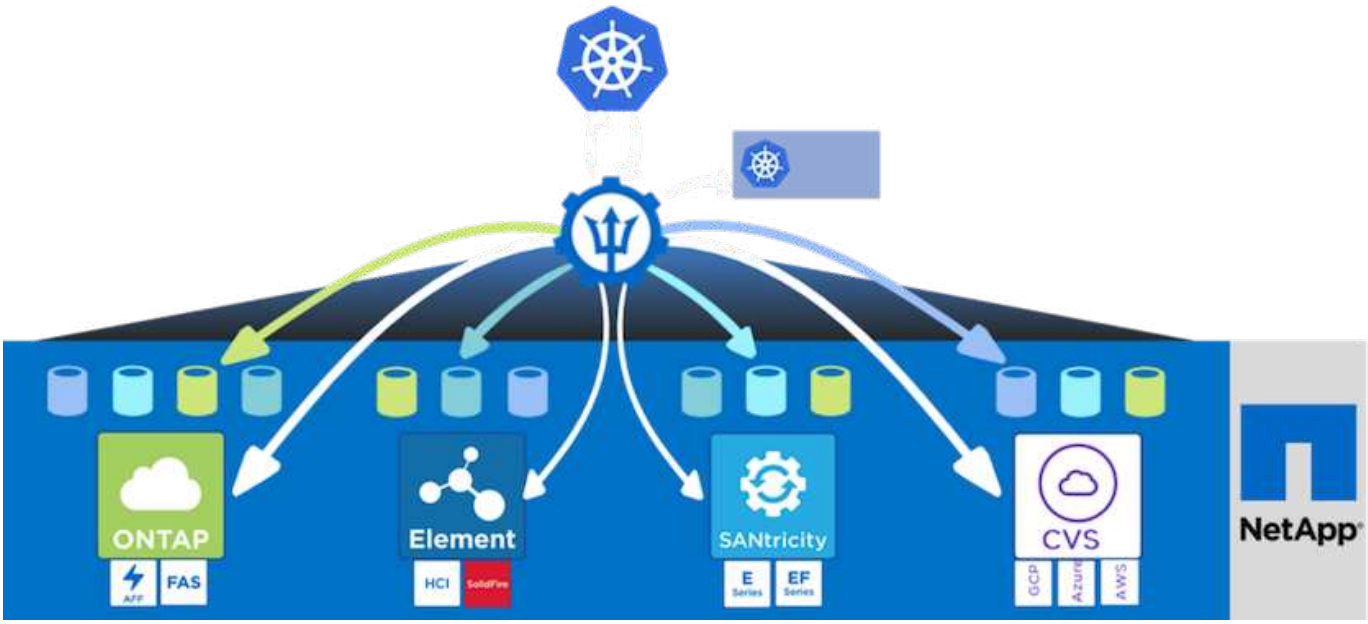
3. Die neue Applikation geht in den Entdeckungszustand, während Astra Control Center die Anwendung im ausgewählten Cluster erstellt. Nachdem alle Ressourcen der Anwendung installiert und von Astra erkannt wurden, geht die Anwendung in den verfügbaren Zustand.



Astra Trident – Überblick

Astra Trident ist ein Open-Source- und vollständig unterstützter Storage-Orchestrator für Container und Kubernetes-Distributionen, einschließlich Red hat OpenShift. Trident kann mit dem gesamten NetApp Storage-Portfolio eingesetzt werden, einschließlich NetApp ONTAP und Element Storage-Systeme. Es unterstützt auch NFS- und iSCSI-Verbindungen. Trident beschleunigt den DevOps-Workflow, da Endbenutzer Storage über ihre NetApp Storage-Systeme bereitstellen und managen können, ohne dass ein Storage-Administrator eingreifen muss.

Ein Administrator kann verschiedene Storage-Back-Ends basierend auf den Projektanforderungen und Storage-Systemmodellen konfigurieren, die erweiterte Storage-Funktionen wie Komprimierung, bestimmte Festplattentypen oder QoS-Level ermöglichen, die eine bestimmte Performance garantieren. Nach ihrer Definition können diese Back-Ends von Entwicklern in ihren Projekten verwendet werden, um persistente Volume Claims (PVCs) zu erstellen und persistenten Storage nach Bedarf an ihre Container anzubinden.



Astra Trident verfügt über einen schnellen Entwicklungszyklus, und genau wie Kubernetes bereits viermal im

Jahr veröffentlicht.

Die neueste Version von Astra Trident ist die 22.01 Version von Januar 2022. Eine Support-Matrix, in der die Version von Trident getestet wurde, mit der Kubernetes Distribution zu finden ist "Hier".

Ab Version 20.04 wird die Trident-Einrichtung vom Trident Operator durchgeführt. Der Operator vereinfacht umfangreiche Implementierungen und bietet zusätzlichen Support einschließlich Selbstreparatur für Pods, die im Rahmen der Trident-Installation bereitgestellt werden.

In der Version 21.01 wurde ein Helm Chart zur Erleichterung der Installation des Trident Operators zur Verfügung gestellt.

Laden Sie Astra Trident Herunter

Um Trident auf dem implementierten Benutzer-Cluster zu installieren und ein persistentes Volume bereitzustellen, gehen Sie die folgenden Schritte durch:

1. Laden Sie das Installationsarchiv auf die Admin-Arbeitsstation herunter und extrahieren Sie den Inhalt. Die aktuelle Version von Trident ist 22.01, die heruntergeladen werden kann "Hier".

```
[netapp-user@rhel7 ~]$ wget
https://github.com/NetApp/trident/releases/download/v22.01.0/trident-
installer-22.01.0.tar.gz
--2021-05-06 15:17:30--
https://github.com/NetApp/trident/releases/download/v22.01.0/trident-
installer-22.01.0.tar.gz
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-
releases.githubusercontent.com/77179634/a4fa9f00-a9f2-11eb-9053-
98e8e573d4ae?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20210506%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20210506T191643Z&X-Amz-Expires=300&X-
Amz-
Signature=8a49a2a1e08c147d1ddd8149ce45a5714f9853fee19bb1c507989b9543eb36
30&X-Amz-
SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-
content-disposition=attachment%3B%20filename%3Dtrident-installer-
22.01.0.tar.gz&response-content-type=application%2Foctet-stream
[following]
--2021-05-06 15:17:30--  https://github-
releases.githubusercontent.com/77179634/a4fa9f00-a9f2-11eb-9053-
98e8e573d4ae?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20210506%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20210506T191643Z&X-Amz-Expires=300&X-
Amz-
Signature=8a49a2a1e08c147d1ddd8149ce45a5714f9853fee19bb1c507989b9543eb36
30&X-Amz-
```

```
SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-
content-disposition=attachment%3B%20filename%3Dtrident-installer-
22.01.0.tar.gz&response-content-type=application%2Foctet-stream
Resolving github-releases.githubusercontent.com (github-
releases.githubusercontent.com)... 185.199.108.154, 185.199.109.154,
185.199.110.154, ...
Connecting to github-releases.githubusercontent.com (github-
releases.githubusercontent.com)|185.199.108.154|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 38349341 (37M) [application/octet-stream]
Saving to: 'trident-installer-22.01.0.tar.gz'

100%[=================================================================
===============================================>] 38,349,341  88.5MB/s
in 0.4s

2021-05-06 15:17:30 (88.5 MB/s) - 'trident-installer-22.01.0.tar.gz'
saved [38349341/38349341]
```

2. Extrahieren Sie die Trident Installation aus dem heruntergeladenen Paket.

```
[netapp-user@rhel7 ~]$ tar -xzf trident-installer-22.01.0.tar.gz
[netapp-user@rhel7 ~]$ cd trident-installer/
[netapp-user@rhel7 trident-installer]$
```

Installieren Sie den Trident Operator mit Helm

1. Legen Sie zunächst den Speicherort des Benutzer-Clusters fest `kubeconfig` Datei als Umgebungsvariable, damit Sie nicht darauf verweisen müssen, weil Trident keine Option hat, diese Datei zu übergeben.

```
[netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/ocp-
install/auth/kubeconfig
```

2. Führen Sie den Helm-Befehl aus, um den Trident-Operator aus dem Tarball im Steuerverzeichnis zu installieren, während der Dreizack-Namespace in Ihrem Benutzer-Cluster erstellt wird.

```
[netapp-user@rhel7 trident-installer]$ helm install trident
helm/trident-operator-22.01.0.tgz --create-namespace --namespace trident
NAME: trident
LAST DEPLOYED: Fri May  7 12:54:25 2021
NAMESPACE: trident
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Thank you for installing trident-operator, which will deploy and manage
NetApp's Trident CSI
storage provisioner for Kubernetes.

Your release is named 'trident' and is installed into the 'trident'
namespace.
Please note that there must be only one instance of Trident (and
trident-operator) in a Kubernetes cluster.

To configure Trident to manage storage resources, you will need a copy
of tridentctl, which is
available in pre-packaged Trident releases.  You may find all Trident
releases and source code
online at https://github.com/NetApp/trident.

To learn more about the release, try:

  $ helm status trident
  $ helm get all trident
```

3. Sie können überprüfen, ob Trident erfolgreich installiert wurde, indem Sie die Pods prüfen, die im Namespace ausgeführt werden, oder die tridentctl-Binärdatei verwenden, um die installierte Version zu überprüfen.

```
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-5z45l                   1/2     Running   2          30s
trident-csi-696b685cf8-htdb2        6/6     Running   0          30s
trident-csi-b74p2                   2/2     Running   0          30s
trident-csi-lrw4n                   2/2     Running   0          30s
trident-operator-7c748d957-gr2gw    1/1     Running   0          36s

[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident version
+----------------+----------------+
| SERVER VERSION | CLIENT VERSION |
+----------------+----------------+
| 22.01.0          | 22.01.0          |
+----------------+----------------+
```

In einigen Fällen müssen die Kundenumgebungen möglicherweise die Anpassungen der Trident Implementierung erfordern. In diesen Fällen kann der Trident-Operator manuell installiert und die enthaltenen Manifeste aktualisiert werden, um die Implementierung anzupassen.

Trident Operator kann manuell installiert werden

1. Legen Sie zunächst den Speicherort des Benutzer-Clusters fest `kubeconfig` Datei als Umgebungsvariable, damit Sie nicht darauf verweisen müssen, weil Trident keine Option hat, diese Datei zu übergeben.

```
[netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/ocp-install/auth/kubeconfig
```

2. Der `trident-installer` Das Verzeichnis enthält Manifeste für die Definition aller erforderlichen Ressourcen. Erstellen Sie mithilfe der entsprechenden Manifeste das `TridentOrchestrator` Benutzerdefinierte Ressourcendefinition.

```
[netapp-user@rhel7 trident-installer]$ oc create -f deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
customresourcedefinition.apiextensions.k8s.io/tridentorchestrators.trident.netapp.io created
```

3. Wenn nicht vorhanden ist, erstellen Sie mithilfe des angegebenen Manifests einen Trident Namespace im Cluster.

```
[netapp-user@rhel7 trident-installer]$ oc apply -f deploy/namespace.yaml
namespace/trident created
```

4. Erstellen Sie die Ressourcen, die für die Trident-Operator, wie z. B. ein, erforderlich sind `ServiceAccount` Für den Operator A `ClusterRole` Und `ClusterRoleBinding` Bis zum `ServiceAccount`, Eine engagierte `PodSecurityPolicy`, Oder der Operator selbst.

```
[netapp-user@rhel7 trident-installer]$ oc create -f deploy/bundle.yaml
serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created
```

5. Sie können den Status des Bedieners überprüfen, nachdem er mit den folgenden Befehlen bereitgestellt wurde:

```
[netapp-user@rhel7 trident-installer]$ oc get deployment -n trident
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
trident-operator   1/1     1            1           23s
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-operator-66f48895cc-lzczk   1/1     Running   0          41s
```

6. Mit dem implementierten Operator können wir nun Trident installieren. Dazu muss ein erstellt werden `TridentOrchestrator`.

```
[netapp-user@rhel7 trident-installer]$ oc create -f
deploy/crds/tridentorchestrator_cr.yaml
tridentorchestrator.trident.netapp.io/trident created
[netapp-user@rhel7 trident-installer]$ oc describe torc trident
Name:         trident
Namespace:
Labels:       <none>
Annotations:  <none>
API Version:  trident.netapp.io/v1
Kind:         TridentOrchestrator
Metadata:
  Creation Timestamp:  2021-05-07T17:00:28Z
  Generation:          1
  Managed Fields:
    API Version:  trident.netapp.io/v1
    Fields Type:  FieldsV1
    fieldsV1:
      f:spec:
        .:
        f:debug:
```

```
        f:namespace:
    Manager:      kubectl-create
    Operation:    Update
    Time:         2021-05-07T17:00:28Z
    API Version:  trident.netapp.io/v1
    Fields Type:  FieldsV1
    fieldsV1:
      f:status:
        .:
        f:currentInstallationParams:
          .:
          f:IPv6:
          f:autosupportHostname:
          f:autosupportImage:
          f:autosupportProxy:
          f:autosupportSerialNumber:
          f:debug:
          f:enableNodePrep:
          f:imagePullSecrets:
          f:imageRegistry:
          f:k8sTimeout:
          f:kubeletDir:
          f:logFormat:
          f:silenceAutosupport:
          f:tridentImage:
        f:message:
        f:namespace:
        f:status:
        f:version:
    Manager:         trident-operator
    Operation:       Update
    Time:            2021-05-07T17:00:28Z
  Resource Version:  931421
  Self Link:
/apis/trident.netapp.io/v1/tridentorchestrators/trident
  UID:               8a26a7a6-dde8-4d55-9b66-a7126754d81f
Spec:
  Debug:      true
  Namespace:  trident
Status:
  Current Installation Params:
    IPv6:                       false
    Autosupport Hostname:
    Autosupport Image:          netapp/trident-autosupport:21.01
    Autosupport Proxy:
    Autosupport Serial Number:
```

```
    Debug:                        true
    Enable Node Prep:             false
    Image Pull Secrets:
    Image Registry:
    k8sTimeout:           30
    Kubelet Dir:          /var/lib/kubelet
    Log Format:           text
    Silence Autosupport:  false
    Trident Image:        netapp/trident:22.01.0
  Message:                Trident installed
  Namespace:              trident
  Status:                 Installed
  Version:                v22.01.0
Events:
  Type    Reason      Age   From                        Message
  ----    ------      ----  ----                        -------
  Normal  Installing  80s   trident-operator.netapp.io  Installing
Trident
  Normal  Installed   68s   trident-operator.netapp.io  Trident
installed
```

7. Sie können überprüfen, ob Trident erfolgreich installiert wurde, indem Sie die Pods prüfen, die im Namespace ausgeführt werden, oder die tridentctl-Binärdatei verwenden, um die installierte Version zu überprüfen.

```
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-bb64c6cb4-lmd6h         6/6     Running   0          82s
trident-csi-gn59q                   2/2     Running   0          82s
trident-csi-m4szj                   2/2     Running   0          82s
trident-csi-sb9k9                   2/2     Running   0          82s
trident-operator-66f48895cc-lzczk   1/1     Running   0          2m39s

[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident version
+----------------+----------------+
| SERVER VERSION | CLIENT VERSION |
+----------------+----------------+
| 22.01.0          | 22.01.0          |
+----------------+----------------+
```

Worker-Nodes für Storage vorbereiten

NFS

Bei den meisten Kubernetes-Distributionen kommen Pakete und Utilities zur standardmäßig installierten NFS-

Back-Ends einschließlich Red hat OpenShift zum Einsatz.

Bei NFSv3 gibt es jedoch keinen Mechanismus, um die Parallelität zwischen dem Client und dem Server auszuhandeln. Daher muss die maximale Anzahl der clientseitigen sunrpc-Slot-Tabelleneinträge manuell mit dem unterstützten Wert auf dem Server synchronisiert werden, um die beste Leistung für die NFS-Verbindung zu gewährleisten, ohne dass der Server die Fenstergröße der Verbindung verringern muss.

Bei ONTAP ist die unterstützte maximale Anzahl von sunrpc-Slot-Tabelleneinträgen 128, d.h. ONTAP kann 128 gleichzeitige NFS-Anfragen gleichzeitig verarbeiten. Standardmäßig hat Red hat CoreOS/Red hat Enterprise Linux jedoch maximal 65,536 Sunrpc Slot-Tabelleneinträge pro Verbindung. Dieser Wert muss auf 128 gesetzt werden. Dies kann mit Machine Config Operator (MCO) in OpenShift geschehen.

Gehen Sie wie folgt vor, um die maximalen Einträge in den OpenShift Worker Nodes zu ändern:

1. Melden Sie sich bei der OCP-Webkonsole an, und navigieren Sie zu „Compute“ > „Machine Configs“. Klicken Sie Auf Maschinenkonfiguration Erstellen. Kopieren Sie die YAML-Datei und fügen Sie sie ein, und klicken Sie auf Erstellen.

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 98-worker-nfs-rpc-slot-tables
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
        - contents:
            source: data:text/plain;charset=utf-
8; base64,b3B0aW9ucyBzdW5ycGMgdGNwX21heF9zbG90X3RhYmxlX2VudHJpZXM9MTI4Cg=
=
          filesystem: root
          mode: 420
          path: /etc/modprobe.d/sunrpc.conf
```

2. Nach der Erstellung des MCO muss die Konfiguration auf alle Arbeitsknoten angewendet und nacheinander neu gestartet werden. Der gesamte Vorgang dauert etwa 20 bis 30 Minuten. Überprüfen Sie, ob die Maschinenkonfiguration mit angewendet wird `oc get mcp` Und stellen Sie sicher, dass der Konfigurationspool für die Maschinenkonfiguration für die Arbeitnehmer aktualisiert wird.

```
[netapp-user@rhel7 openshift-deploy]$ oc get mcp
NAME     CONFIG                                     UPDATED   UPDATING
DEGRADED
master   rendered-master-a520ae930e1d135e0dee7168   True      False
False
worker   rendered-worker-de321b36eeba62df41feb7bc   True      False
False
```

ISCSI

Um die Worker-Knoten vorzubereiten, die die Zuordnung von Block-Speicher-Volumes über das iSCSI-Protokoll ermöglichen, müssen Sie die erforderlichen Pakete installieren, um diese Funktionalität zu unterstützen.

In Red hat OpenShift wird dieser Vorgang durch Anwendung eines MCO (Machine Config Operator) auf das Cluster durchgeführt, nachdem es bereitgestellt wurde.

Führen Sie die folgenden Schritte aus, um die Worker-Knoten für die Ausführung von iSCSI-Diensten zu konfigurieren:

1. Melden Sie sich bei der OCP-Webkonsole an, und navigieren Sie zu „Compute“ > „Machine Configs“. Klicken Sie Auf Maschinenkonfiguration Erstellen. Kopieren Sie die YAML-Datei und fügen Sie sie ein, und klicken Sie auf Erstellen.

   Wenn Sie kein Multipathing verwenden:

   ```
   apiVersion: machineconfiguration.openshift.io/v1
   kind: MachineConfig
   metadata:
     labels:
       machineconfiguration.openshift.io/role: worker
     name: 99-worker-element-iscsi
   spec:
     config:
       ignition:
         version: 3.2.0
       systemd:
         units:
           - name: iscsid.service
             enabled: true
             state: started
     osImageURL: ""
   ```

   Bei Verwendung von Multipathing:

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 99-worker-ontap-iscsi
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
      - contents:
          source: data:text/plain;charset=utf-
8; base64,ZGVmYXVsdHMgewogICAgICAgIHVzZXJfZnJpZW5kbHlfbmFtZXMgbm8KICAgICA
gICBmaW5kX211bHRpcGF0aHMgbm8KfQoKYmxhY2tsaXN0X2V4Y2VwdGlvbnMgewogICAgICA
gIHByb3BlcnR5ICIoU0NTSV9JREVOVF98SURfV1dOKSIKfQoKYmxhY2tsaXN0IHsKfQoK
          verification: {}
        filesystem: root
        mode: 400
        path: /etc/multipath.conf
    systemd:
      units:
        - name: iscsid.service
          enabled: true
          state: started
        - name: multipathd.service
          enabled: true
          state: started
  osImageURL: ""
```

2. Nach der Erstellung der Konfiguration dauert es etwa 20 bis 30 Minuten, die Konfiguration auf die Worker-Nodes anzuwenden und erneut zu laden. Überprüfen Sie, ob die Maschinenkonfiguration mit angewendet wird `oc get mcp` Und stellen Sie sicher, dass der Konfigurationspool für die Maschinenkonfiguration für die Arbeitnehmer aktualisiert wird. Sie können sich auch bei den Worker-Nodes anmelden, um zu bestätigen, dass der iscsid-Service ausgeführt wird (und der Multipathd-Dienst wird ausgeführt, wenn Multipathing verwendet wird).

```
[netapp-user@rhel7 openshift-deploy]$ oc get mcp
NAME     CONFIG                                     UPDATED   UPDATING
DEGRADED
master   rendered-master-a520ae930e1d135e0dee7168   True      False
False
worker   rendered-worker-de321b36eeba62df41feb7bc   True      False
False

[netapp-user@rhel7 openshift-deploy]$ ssh core@10.61.181.22 sudo
systemctl status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled;
vendor preset: disabled)
   Active: active (running) since Tue 2021-05-26 13:36:22 UTC; 3 min ago
     Docs: man:iscsid(8)
           man:iscsiadm(8)
 Main PID: 1242 (iscsid)
   Status: "Ready to process requests"
    Tasks: 1
   Memory: 4.9M
      CPU: 9ms
   CGroup: /system.slice/iscsid.service
           └─1242 /usr/sbin/iscsid -f

[netapp-user@rhel7 openshift-deploy]$ ssh core@10.61.181.22 sudo
systemctl status multipathd
 ● multipathd.service - Device-Mapper Multipath Device Controller
   Loaded: loaded (/usr/lib/systemd/system/multipathd.service; enabled;
vendor preset: enabled)
   Active: active (running) since Tue 2021-05-26 13:36:22 UTC; 3 min ago
  Main PID: 918 (multipathd)
    Status: "up"
    Tasks: 7
    Memory: 13.7M
    CPU: 57ms
    CGroup: /system.slice/multipathd.service
            └─918 /sbin/multipathd -d -s
```

Es ist auch möglich zu bestätigen, dass die MachineConfig erfolgreich angewendet wurde und die Dienste wie erwartet durch Ausführen der gestartet wurden `oc debug` Befehl mit den entsprechenden Flags.

Erstellen von Storage-System-Back-Ends

Nach Abschluss der Installation des Astra Trident Operator müssen Sie das Backend für die spezifische NetApp Storage-Plattform konfigurieren, die Sie verwenden. Folgen Sie den Links unten, um mit der Einrichtung und Konfiguration von Astra Trident fortzufahren.

- "NetApp ONTAP NFS"
- "NetApp ONTAP iSCSI"
- "NetApp Element iSCSI"

Konfiguration von NetApp ONTAP NFS

Um die Trident Integration mit dem NetApp ONTAP Storage-System zu aktivieren, müssen Sie ein Back-End erstellen, das die Kommunikation mit dem Storage-System ermöglicht.

1. Im heruntergeladenen Installationsarchiv stehen Beispiele für Backend-Dateien zur Verfügung `sample-input` Ordnerhierarchie. Kopieren Sie für NetApp ONTAP-Systeme, die NFS bereitstellen, den `backend-ontap-nas.json` Datei in Ihr Arbeitsverzeichnis und bearbeiten Sie die Datei.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/ontap-nas/backend-ontap-nas.json ./
[netapp-user@rhel7 trident-installer]$ vi backend-ontap-nas.json
```

2. BackendName, Management LIF, Daten LIF, svm, Benutzername, bearbeiten Und Kennwortwerte in dieser Datei.

```
{
    "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": "cluster-admin",
    "password": "password"
}
```

> Als Best Practice empfiehlt es sich, den benutzerdefinierten BackendName-Wert als Kombination aus storageDriverName und der DatenLIF zu definieren, die NFS bedienen, um die einfache Identifizierung zu erleichtern.

3. Führen Sie mit dieser Backend-Datei den folgenden Befehl aus, um Ihr erstes Backend zu erstellen.

```
[netapp-user@rhel7 trident-installer]$ ./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 |
+------------------------+----------------
+--------------------------------------+--------+---------+
```

4. Wenn das Back-End erstellt wird, müssen Sie als nächstes eine Storage-Klasse erstellen. Wie beim Backend gibt es auch eine Beispiel-Speicherklassendatei, die für die im Ordner Sample-Inputs verfügbare Umgebung bearbeitet werden kann. Kopieren Sie ihn in das Arbeitsverzeichnis und nehmen Sie die erforderlichen Änderungen an dem erstellten Backend vor.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

5. Die einzige Bearbeitung, die zu dieser Datei gemacht werden muss, ist das zu definieren `backendType` Wert für den Namen des Speichertreibers aus dem neu erstellten Back-End. Notieren Sie auch den Wert des Namensfelds, auf den in einem späteren Schritt verwiesen werden muss.

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

> Es gibt ein optionales Feld mit dem Namen `fsType` Das ist in dieser Datei definiert. Diese Zeile kann in NFS-Back-Ends gelöscht werden.

6. Führen Sie die aus `oc` Befehl zum Erstellen der Storage-Klasse.

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

7. Nach Erstellung der Storage-Klasse müssen Sie dann die erste Forderung für ein persistentes Volume (PVC) erstellen. Es gibt ein Beispiel `pvc-basic.yaml` Datei, mit der diese Aktion ausgeführt werden kann, die sich auch in Sample-Inputs befindet.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-
basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

8. Die einzige Bearbeitung, die zu dieser Datei gemacht werden muss, ist sicherzustellen, dass die `storageClassName` Feld entspricht dem gerade erstellten. Die PVC-Definition kann je nach Bedarf des bereitzustellenden Workloads weiter angepasst werden.

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

9. Erstellen Sie das PVC, indem Sie die ausstellen `oc` Befehl. Die Erstellung kann je nach Größe des erstellten Sicherungsvolumens einige Zeit in Anspruch nehmen, sodass Sie den Prozess nach Abschluss beobachten können.

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

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME    STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
basic   Bound    pvc-b4370d37-0fa4-4c17-bd86-94f96c94b42d   1Gi
RWO            basic-csi      7s
```

NetApp ONTAP iSCSI-Konfiguration

Um die Trident Integration mit dem NetApp ONTAP Storage-System zu aktivieren, müssen Sie ein Back-End erstellen, das die Kommunikation mit dem Storage-System ermöglicht.

1. Im heruntergeladenen Installationsarchiv stehen Beispiele für Backend-Dateien zur Verfügung `sample-input` Ordnerhierarchie. Kopieren Sie bei NetApp ONTAP-Systemen, die iSCSI bereitstellen, das `backend-ontap-san.json` Datei in Ihr Arbeitsverzeichnis und bearbeiten Sie die Datei.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/ontap-san/backend-ontap-san.json ./
[netapp-user@rhel7 trident-installer]$ vi backend-ontap-san.json
```

2. Bearbeiten Sie die Werte ManagementLIF, dataLIF, svm, Benutzername und Passwort in dieser Datei.

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "172.21.224.201",
  "dataLIF": "10.61.181.240",
  "svm": "trident_svm",
  "username": "admin",
  "password": "password"
}
```

3. Führen Sie mit dieser Backend-Datei den folgenden Befehl aus, um Ihr erstes Backend zu erstellen.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-san.json
+------------------------+----------------
+--------------------------------------+--------+---------+
|          NAME          | STORAGE DRIVER |                  UUID
| STATE  | VOLUMES |
+------------------------+----------------
+--------------------------------------+--------+---------+
| ontapsan_10.61.181.241 | ontap-san      | 6788533c-7fea-4a35-b797-
fb9bb3322b91 | online |       0 |
+------------------------+----------------
+--------------------------------------+--------+---------+
```

4. Wenn das Back-End erstellt wird, müssen Sie als nächstes eine Storage-Klasse erstellen. Wie beim Backend gibt es auch eine Beispiel-Speicherklassendatei, die für die im Ordner Sample-Inputs verfügbare Umgebung bearbeitet werden kann. Kopieren Sie ihn in das Arbeitsverzeichnis und nehmen Sie die erforderlichen Änderungen an dem erstellten Backend vor.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

5. Die einzige Bearbeitung, die zu dieser Datei gemacht werden muss, ist das zu definieren `backendType` Wert für den Namen des Speichertreibers aus dem neu erstellten Back-End. Notieren Sie auch den Wert des Namensfelds, auf den in einem späteren Schritt verwiesen werden muss.

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

> Es gibt ein optionales Feld mit dem Namen `fsType` Das ist in dieser Datei definiert. In iSCSI-Back-Ends kann dieser Wert auf einen bestimmten Linux-Dateisystem-Typ (XFS, ext4 usw.) gesetzt oder gelöscht werden, damit OpenShift entscheiden kann, welches Dateisystem verwendet werden soll.

6. Führen Sie die aus `oc` Befehl zum Erstellen der Storage-Klasse.

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

7. Nach Erstellung der Storage-Klasse müssen Sie dann die erste Forderung für ein persistentes Volume (PVC) erstellen. Es gibt ein Beispiel `pvc-basic.yaml` Datei, mit der diese Aktion ausgeführt werden kann, die sich auch in Sample-Inputs befindet.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-
basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

8. Die einzige Bearbeitung, die zu dieser Datei gemacht werden muss, ist sicherzustellen, dass die `storageClassName` Feld entspricht dem gerade erstellten. Die PVC-Definition kann je nach Bedarf des bereitzustellenden Workloads weiter angepasst werden.

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

9. Erstellen Sie das PVC, indem Sie die ausstellen `oc` Befehl. Die Erstellung kann je nach Größe des erstellten Sicherungsvolumens einige Zeit in Anspruch nehmen, sodass Sie den Prozess nach Abschluss beobachten können.

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

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME    STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
basic   Bound    pvc-7ceac1ba-0189-43c7-8f98-094719f7956c   1Gi
RWO            basic-csi      3s
```

ISCSI-Konfiguration von NetApp Element

Um die Trident Integration mit dem NetApp Element Storage-System zu aktivieren, müssen Sie ein Backend erstellen, das die Kommunikation mit dem Storage-System über das iSCSI-Protokoll ermöglicht.

1. Im heruntergeladenen Installationsarchiv stehen Beispiele für Backend-Dateien zur Verfügung `sample-input` Ordnerhierarchie. Kopieren Sie für NetApp Element-Systeme, die iSCSI-Server bereitstellen, das `backend-solidfire.json` Datei in Ihr Arbeitsverzeichnis und bearbeiten Sie die Datei.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/solidfire/backend-solidfire.json ./
[netapp-user@rhel7 trident-installer]$ vi ./backend-solidfire.json
```

   a. Bearbeiten Sie den Benutzer-, das Kennwort und den MVIP-Wert auf dem `EndPoint` Linie.

   b. Bearbeiten Sie das `SVIP` Wert:

```
 {
    "version": 1,
    "storageDriverName": "solidfire-san",
    "Endpoint": "https://trident:password@172.21.224.150/json-
rpc/8.0",
    "SVIP": "10.61.180.200:3260",
    "TenantName": "trident",
    "Types": [{"Type": "Bronze", "Qos": {"minIOPS": 1000, "maxIOPS":
2000, "burstIOPS": 4000}},
              {"Type": "Silver", "Qos": {"minIOPS": 4000, "maxIOPS":
6000, "burstIOPS": 8000}},
              {"Type": "Gold", "Qos": {"minIOPS": 6000, "maxIOPS":
8000, "burstIOPS": 10000}}]
}
```

2. Führen Sie mit dieser Back-End-Datei den folgenden Befehl aus, um Ihr erstes Backend zu erstellen.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-solidfire.json
+------------------------+----------------
+--------------------------------------+--------+---------+
|          NAME          | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+------------------------+----------------
+--------------------------------------+--------+---------+
| solidfire_10.61.180.200 | solidfire-san  | b90783ee-e0c9-49af-8d26-
3ea87ce2efdf | online |       0 |
+------------------------+----------------
+--------------------------------------+--------+---------+
```

3. Wenn das Back-End erstellt wird, müssen Sie als nächstes eine Storage-Klasse erstellen. Wie beim Backend gibt es auch eine Beispiel-Speicherklassendatei, die für die im Ordner Sample-Inputs verfügbare Umgebung bearbeitet werden kann. Kopieren Sie ihn in das Arbeitsverzeichnis und nehmen Sie die erforderlichen Änderungen an dem erstellten Backend vor.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

4. Die einzige Bearbeitung, die zu dieser Datei gemacht werden muss, ist das zu definieren `backendType` Wert für den Namen des Speichertreibers aus dem neu erstellten Back-End. Notieren Sie auch den Wert des Namensfelds, auf den in einem späteren Schritt verwiesen werden muss.

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

> Es gibt ein optionales Feld mit dem Namen `fsType` Das ist in dieser Datei definiert. In iSCSI-Back-Ends kann dieser Wert auf einen bestimmten Linux-Dateisystem-Typ (XFS, ext4 usw.) gesetzt werden. OpenShift kann auch gelöscht werden, damit OpenShift entscheiden kann, welches Dateisystem verwendet werden soll.

5. Führen Sie die aus `oc` Befehl zum Erstellen der Storage-Klasse.

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

6. Nach Erstellung der Storage-Klasse müssen Sie dann die erste Forderung für ein persistentes Volume (PVC) erstellen. Es gibt ein Beispiel `pvc-basic.yaml` Datei, mit der diese Aktion ausgeführt werden kann, die sich auch in Sample-Inputs befindet.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-
basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

7. Die einzige Bearbeitung, die zu dieser Datei gemacht werden muss, ist sicherzustellen, dass die `storageClassName` Feld entspricht dem gerade erstellten. Die PVC-Definition kann je nach Bedarf des bereitzustellenden Workloads weiter angepasst werden.

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

8. Erstellen Sie das PVC, indem Sie die ausstellen `oc` Befehl. Die Erstellung kann je nach Größe des erstellten Sicherungsvolumens einige Zeit in Anspruch nehmen, sodass Sie den Prozess nach Abschluss beobachten können.

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

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME    STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
basic   Bound    pvc-3445b5cc-df24-453d-a1e6-b484e874349d   1Gi
RWO            basic-csi      5s
```