



Descripción general de las integraciones de almacenamiento de NetApp

NetApp Solutions

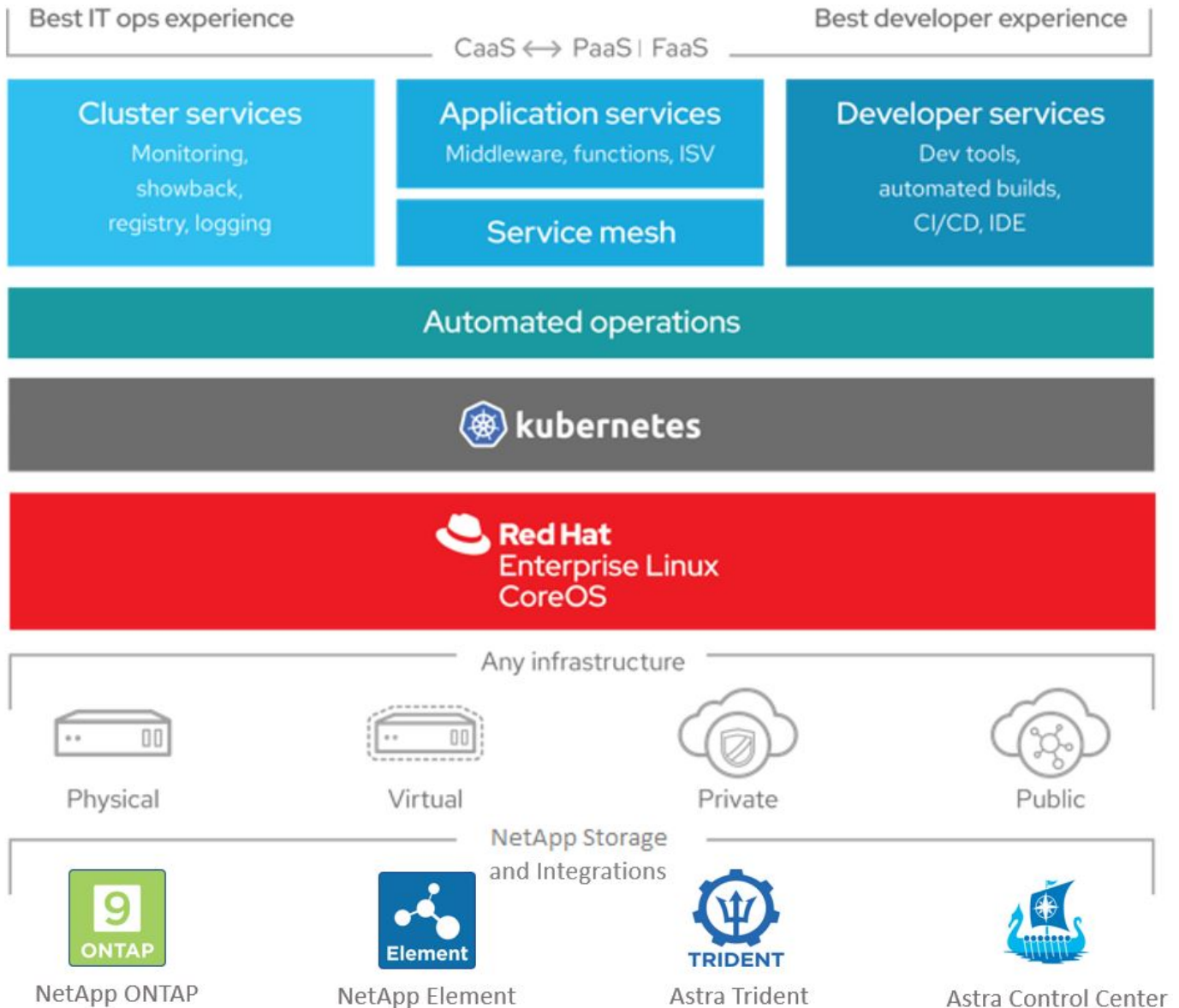
NetApp
April 26, 2024

Tabla de contenidos

- Descripción general de integración del almacenamiento de NetApp 1
 - Descripción general de Astra Control Center de NetApp 2
 - Descripción general de Astra Trident 30

Descripción general de integración del almacenamiento de NetApp

NetApp proporciona una serie de productos que le ayudan a orquestar y gestionar datos persistentes en entornos basados en contenedores, como Red Hat OpenShift.



Astra Control de NetApp ofrece un amplio conjunto de servicios de gestión de datos para aplicaciones y almacenamiento para cargas de trabajo con estado de Kubernetes, con la tecnología de protección de datos de NetApp. El servicio Astra Control está disponible para admitir cargas de trabajo con estado en puestas en marcha de Kubernetes nativas para el cloud. Astra Control Center está disponible para admitir cargas de trabajo con estado en implementaciones en las instalaciones, como Red Hat OpenShift. Si quiere más información, visite el sitio web de Astra Control de NetApp ["aquí"](#).

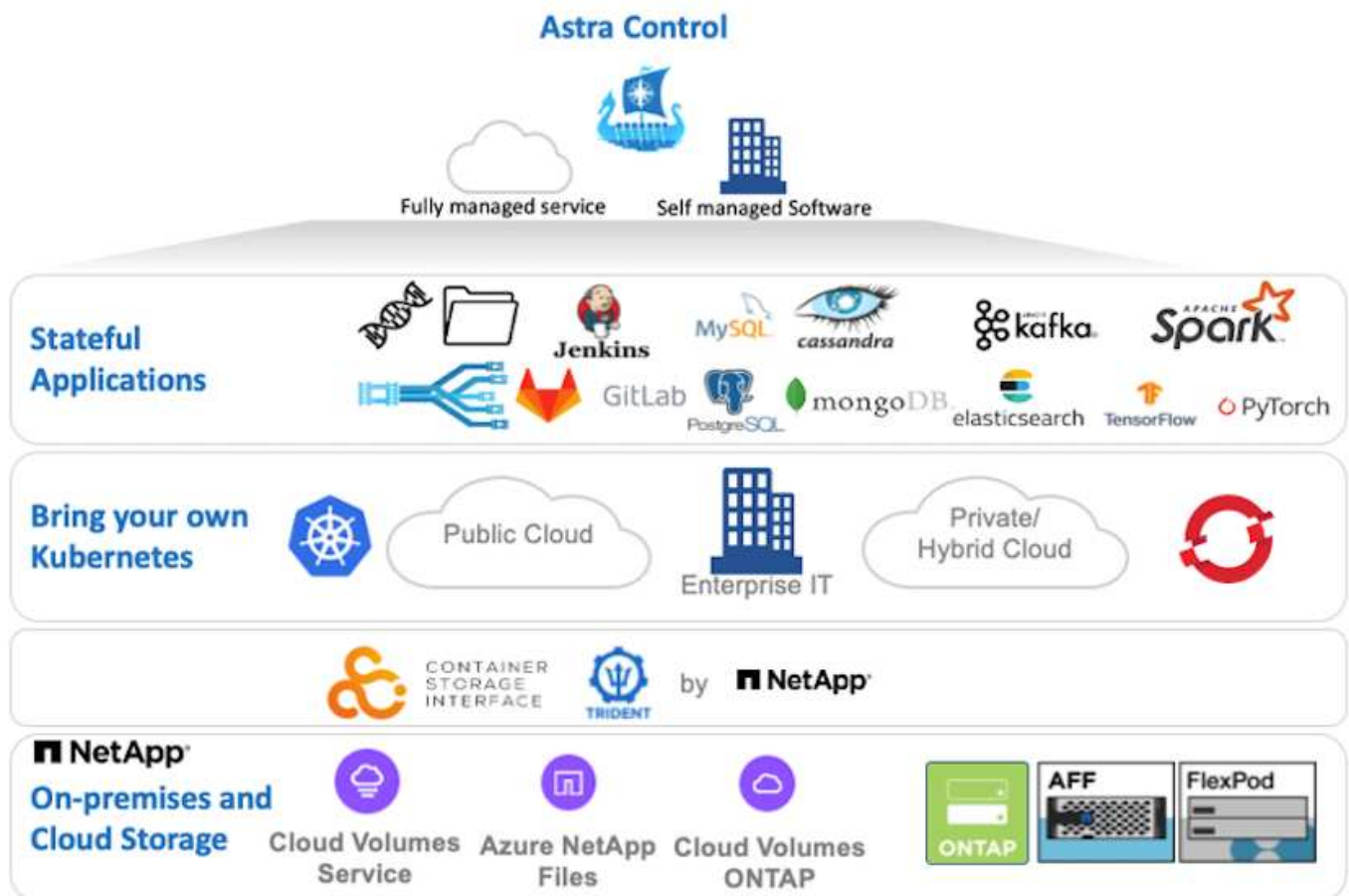
Astra Trident de NetApp es un orquestador de almacenamiento de código abierto y totalmente compatible para contenedores y distribuciones de Kubernetes, incluido Red Hat OpenShift. Si quiere más información, visite el sitio web de Astra Trident ["aquí"](#).

En las siguientes páginas encontrará información adicional sobre los productos de NetApp que se han validado para la gestión de aplicaciones y almacenamiento persistente en la solución Red Hat OpenShift con NetApp:

- ["Centro de control de Astra de NetApp"](#)
- ["Astra Trident de NetApp"](#)

Descripción general de Astra Control Center de NetApp

Astra Control Center de NetApp ofrece un amplio conjunto de servicios de gestión de datos para aplicaciones y almacenamiento para cargas de trabajo con estado de Kubernetes puestas en marcha en un entorno local con la tecnología de protección de datos de NetApp.



NetApp Astra Control Center se puede instalar en un clúster de Red Hat OpenShift que tiene el orquestador de almacenamiento Astra Trident puesto en marcha y configurado con clases de almacenamiento y back-ends de almacenamiento en sistemas de almacenamiento ONTAP de NetApp.

Para obtener información sobre la instalación y configuración de Astra Trident y su compatibilidad con Astra Control Center, consulte ["este documento aquí"](#).

En un entorno conectado a la nube, Astra Control Center utiliza Cloud Insights para proporcionar supervisión y telemetría avanzadas. Ante la ausencia de una conexión con Cloud Insights, la supervisión y la telemetría limitadas (métricas de 7 días) están disponibles y se exportan a herramientas de supervisión nativas de Kubernetes (Prometheus y Grafana) mediante extremos de métricas abiertos.

Astra Control Center está totalmente integrado en el ecosistema de AutoSupport y Active IQ de NetApp para

proporcionar soporte a los usuarios y proporcionar asistencia para la solución de problemas y mostrar las estadísticas de uso.

Además de la versión de pago de Astra Control Center, hay disponible una licencia de evaluación de 90 días. La versión de evaluación se admite a través del correo electrónico y la comunidad (canal Slack). Los clientes tienen acceso a éstos y a otros artículos de la base de conocimientos y a la documentación disponible en la consola de soporte del producto.

Para empezar a utilizar Astra Control Center de NetApp, visite ["Sitio web de Astra"](#).

Requisitos previos de instalación de Astra Control Center

1. Uno o más clústeres de Red Hat OpenShift. Actualmente se admiten las versiones 4.6 EUS y 4.7.
2. Astra Trident ya debe estar instalado y configurado en cada clúster de Red Hat OpenShift.
3. Uno o más sistemas de almacenamiento ONTAP de NetApp que ejecutan ONTAP 9.5 o superior.



Es recomendable que cada instalación de OpenShift en un sitio tenga una SVM dedicada para almacenamiento persistente. Las puestas en marcha de varios sitios requieren sistemas de almacenamiento adicionales.

4. Debe configurarse un back-end de almacenamiento de Trident en cada clúster de OpenShift con una SVM respaldada por un clúster de ONTAP.
5. Un StorageClass predeterminado configurado en cada clúster OpenShift con Astra Trident como proveedor de almacenamiento.
6. Se debe instalar y configurar un equilibrador de carga en cada clúster de OpenShift para que pueda equilibrarse la carga y exponer los servicios de OpenShift.



Consulte el enlace ["aquí"](#) para obtener información sobre balanceadores de carga que se han validado para este propósito.

7. Debe configurarse un registro de imagen privada para alojar las imágenes de Astra Control Center de NetApp.



Consulte el enlace ["aquí"](#) Para instalar y configurar un registro privado de OpenShift con este fin.

8. Debe tener acceso de administrador de clúster al clúster de Red Hat OpenShift.
9. Debe tener acceso de administrador a los clústeres de ONTAP de NetApp.
10. Una estación de trabajo de administración con docker o podman, trimentctl y las herramientas OC o kubectl instaladas y agregadas a su \$PATH.



Las instalaciones de Docker deben tener la versión de docker superior a 20.10 y las instalaciones de Podman deben tener una versión de podman superior a 3.0.

Instalar Astra Control Center

Uso de OperatorHub

1. Inicie sesión en el sitio de soporte de NetApp y descargue la versión más reciente de Astra Control Center de NetApp. Para ello, es necesario disponer de una licencia adjunta a su cuenta de NetApp. Después de descargar el tarball, transfíralo a la estación de trabajo de administración.



Para empezar con una licencia de prueba de Astra Control, visite "[Sitio de registro de Astra](#)".

2. Desembale la bola tar y cambie el directorio de trabajo a la carpeta resultante.

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

3. Antes de iniciar la instalación, empuje las imágenes de Astra Control Center hasta un registro de imágenes. Puede elegir hacer esto con Docker o Podman, las instrucciones para ambos se proporcionan en este paso.

Podman

- a. Exporte el FQDN del registro con el nombre de organización/espacio de nombres/proyecto como una variable de entorno 'sector'.

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

- b. Inicie sesión en el registro.

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



Si está utilizando kubeadmin usuario para iniciar sesión en el registro privado y, a continuación, utilizar token en lugar de password - `podman login -u ocp-user -p token --tls-verify=false astra-registry.apps.ocp-vmw.cie.netapp.com`.



También puede crear una cuenta de servicio, asignar el editor de Registro y/o la función de visor de Registro (en función de si necesita acceso de inserción/extracción) e iniciar sesión en el Registro mediante el token de la cuenta de servicio.

- c. Cree un archivo de script de shell y pegue el siguiente contenido en él.

```
[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
```



Si utiliza certificados que no son de confianza para el registro, edite la secuencia de comandos del shell y utilice `--tls-verify=false` para el comando `podman push` `podman push $REGISTRY/$(echo $astraImage | sed 's/[\\/]\\+/\\\\/') --tls-verify=false`.

d. Haga que el archivo sea ejecutable.

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

e. Ejecute el script shell.

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


Docker

- a. Exporte el FQDN del registro con el nombre de organización/espacio de nombres/proyecto como una variable de entorno 'sector'.

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

- b. Inicie sesión en el registro.

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



Si está utilizando kubeadmin usuario para iniciar sesión en el registro privado y, a continuación, utilizar token en lugar de password - `docker login -u ocp-user -p token astra-registry.apps.ocp-vmw.cie.netapp.com`.



También puede crear una cuenta de servicio, asignar el editor de Registro y/o la función de visor de Registro (en función de si necesita acceso de inserción/extracción) e iniciar sesión en el Registro mediante el token de la cuenta de servicio.

- c. Cree un archivo de script de shell y pegue el siguiente contenido en él.

```
[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. Haga que el archivo sea ejecutable.

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

e. Ejecute el script shell.

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

4. Cuando utilice registros de imágenes privadas de confianza pública, cargue los certificados TLS del registro de imágenes en los nodos OpenShift. Para ello, cree un mapa de configuración en el espacio de nombres de openshift-config mediante los certificados TLS y realice una revisión de la configuración de la imagen del clúster para que el certificado sea de confianza.

```
[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
```



Si está utilizando un registro interno OpenShift con certificados TLS predeterminados del operador Ingress con una ruta, debe seguir el paso anterior para aplicar el parche a los certificados en el nombre de host de la ruta. Para extraer los certificados del operador Ingress, puede utilizar el comando `oc extract secret/router-ca --keys=tls.crt -n openshift-ingress-operator`.

5. Cree un espacio de nombres `netapp-acc-operator` Para Astra Control Center.

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

namespace/netapp-acc-operator created
```

6. Cree un secreto con credenciales para iniciar sesión en el registro de imágenes `netapp-acc-operator` espacio de nombres.


```
[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. Inicie sesión en la consola de la GUI de Red Hat OpenShift con acceso `cluster-admin`.
8. Seleccione Administrador en la lista desplegable perspectiva.
9. Desplácese a operadores > OperatorHub y busque Astra.



10. Seleccione `netapp-acc-operator` mosaico y haga clic en `Install`.

**netapp-acc-operator**✕
21.12.63-1 provided by NetApp

Install

Latest version
21.12.63-1

Capability level
☒ Basic Install
☐ Seamless Upgrades
☐ Full Lifecycle
☐ Deep Insights
☐ Auto Pilot

Provider type
Certified

Provider
NetApp

Astra Control is an application-aware data management solution that manages, protects and moves data-rich Kubernetes workloads in both public clouds and on-premises.

Astra Control enables data protection, disaster recovery, and migration for your Kubernetes workloads, leveraging NetApp's industry-leading data management technology for snapshots, backups, replication and cloning.

How to deploy Astra Control

Refer to [Installation Procedure](#) to deploy Astra Control Center using the Operator.

Documentation

Refer to [Astra Control Center Documentation](#) to complete the setup and start managing applications.

11. En la pantalla instalar operador, acepte todos los parámetros predeterminados y haga clic en `Install`.

Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel *

- ☐ alpha
- ☒ stable

Installation mode *

- ☒ All namespaces on the cluster (default)
Operator will be available in all Namespaces.
- ☐ A specific namespace on the cluster
This mode is not supported by this Operator

Installed Namespace *

PR netapp-acc-operator (Operator recommended)

⚠ Namespace already exists

Namespace **netapp-acc-operator** already exists and will be used. Other users can already have access to this namespace.

Approval strategy *

- ☒ Automatic
- ☐ Manual

Install

Cancel

 **netapp-acc-operator**
provided by NetApp

Provided APIs

 **Astra Control Center**

AstraControlCenter is the Schema for the astracontrolcenters API

12. Espere a que finalice la instalación del operador.



netapp-acc-operator
21.12.63-1 provided by NetApp



Installing Operator

InstallWaiting: installing; waiting for deployment acc-operator-controller-manager to become ready; Waiting for rollout to finish: 0 of 1 updated replicas are available...

The Operator is being installed. This may take a few minutes.

[View installed Operators in Namespace netapp-acc-operator](#)

13. Una vez que la instalación del operador se realice correctamente, desplácese hasta hacer clic en View Operator.



netapp-acc-operator

21.12.63-1 provided by NetApp



Installed operator - ready for use

[View Operator](#)

[View installed Operators in Namespace netapp-acc-operator](#)

14. A continuación, haga clic en `Create Instance` En el mosaico del Centro de control de Astra del operador.

[Installed Operators](#) > [Operator details](#)



netapp-acc-operator

21.12.63-1 provided by NetApp

[Details](#)

[YAML](#)

[Subscription](#)

[Events](#)

[Astra Control Center](#)

Provided APIs

ACC Astra Control Center

AstraControlCenter is the Schema for the astracontrolcenters API

[+ Create instance](#)

15. Rellene el `Create AstraControlCenter` campos de formulario y haga clic en `Create`.
- Opcionalmente, edite el nombre de la instancia de Astra Control Center.
 - Opcionalmente, habilite o deshabilite el AutoSupport. Se recomienda conservar la funcionalidad de AutoSupport.
 - Introduzca el FQDN para Astra Control Center.

- d. Introduzca la versión de Astra Control Center; la última se muestra de forma predeterminada.
- e. Introduzca un nombre de cuenta para Astra Control Center y detalles de administración como nombre, apellidos y dirección de correo electrónico.
- f. Introduzca la política de reclamaciones de volúmenes, el valor predeterminado es Retain.
- g. En el Registro de imágenes, introduzca el FQDN del registro junto con el nombre de la organización que se le dio mientras presiona las imágenes al registro (en este ejemplo, `astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra`)
- h. Si utiliza un registro que requiere autenticación, introduzca el nombre secreto en la sección Image Registry (Registro de imágenes).
- i. Configurar las opciones de ampliación para los límites de recursos de Astra Control Center.
- j. Introduzca el nombre de la clase de almacenamiento si desea colocar las RVP en una clase de almacenamiento no predeterminada.
- k. Defina las preferencias de manejo de CRD.

Project: netapp-acc-operator ▼

Name *

Labels

Account Name *

Astra Control Center account name

Astra Address *

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 *

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

Email *

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

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

[Ansible] automatizado

1. Para utilizar los libros de estrategia de Ansible para poner en marcha Astra Control Center, necesita una máquina Ubuntu/RHEL con Ansible instalado. Siga los procedimientos ["aquí"](#) Para Ubuntu y RHEL.
2. Clone el repositorio de GitHub que aloja el contenido de Ansible.

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

3. Inicie sesión en el sitio de soporte de NetApp y descargue la versión más reciente de Astra Control Center de NetApp. Para ello, es necesario disponer de una licencia adjunta a su cuenta de NetApp. Después de descargar el tarball, transfíralo a la estación de trabajo.



Para empezar con una licencia de prueba de Astra Control, visite ["Sitio de registro de Astra"](#).

4. Cree o obtenga el archivo kubeconfig con acceso de administrador al clúster OpenShift en el que se va a instalar Astra Control Center.

5. Cambie el directorio a `na_astra_control_Suite`.

```
cd na_astra_control_suite
```

6. Edite el `vars/vars.yml` y rellene las variables con la información necesaria.

```
#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"

#Kubernetes/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. Ejecute el libro de estrategia para implementar Astra Control Center. El libro de estrategia requiere privilegios raíz para determinadas configuraciones.

Si el usuario que ejecuta el libro de estrategia es raíz o tiene sudo configurados sin contraseñas, ejecute el siguiente comando para ejecutar el libro de estrategia.

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

Si el usuario tiene configurado un acceso sudo basado en contraseña, ejecute el siguiente comando para ejecutar la libro de estrategia y, a continuación, introduzca la contraseña sudo.

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

Pasos posteriores a la instalación

1. La instalación puede tardar varios minutos en completarse. Verifique que todos los pods y servicios del `netapp-astra-cc` el espacio de nombres está activo y en funcionamiento.

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

2. Compruebe la `acc-operator-controller-manager` registros para garantizar que se completa la instalación.

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



El siguiente mensaje indica que la instalación de Astra Control Center se ha realizado correctamente.

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

3. El nombre de usuario para iniciar sesión en Astra Control Center es la dirección de correo electrónico del administrador que se proporciona en el archivo CRD y la contraseña es una cadena ACC- Se adjunta al UUID del Centro de control de Astra. Ejecute el siguiente comando:

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

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



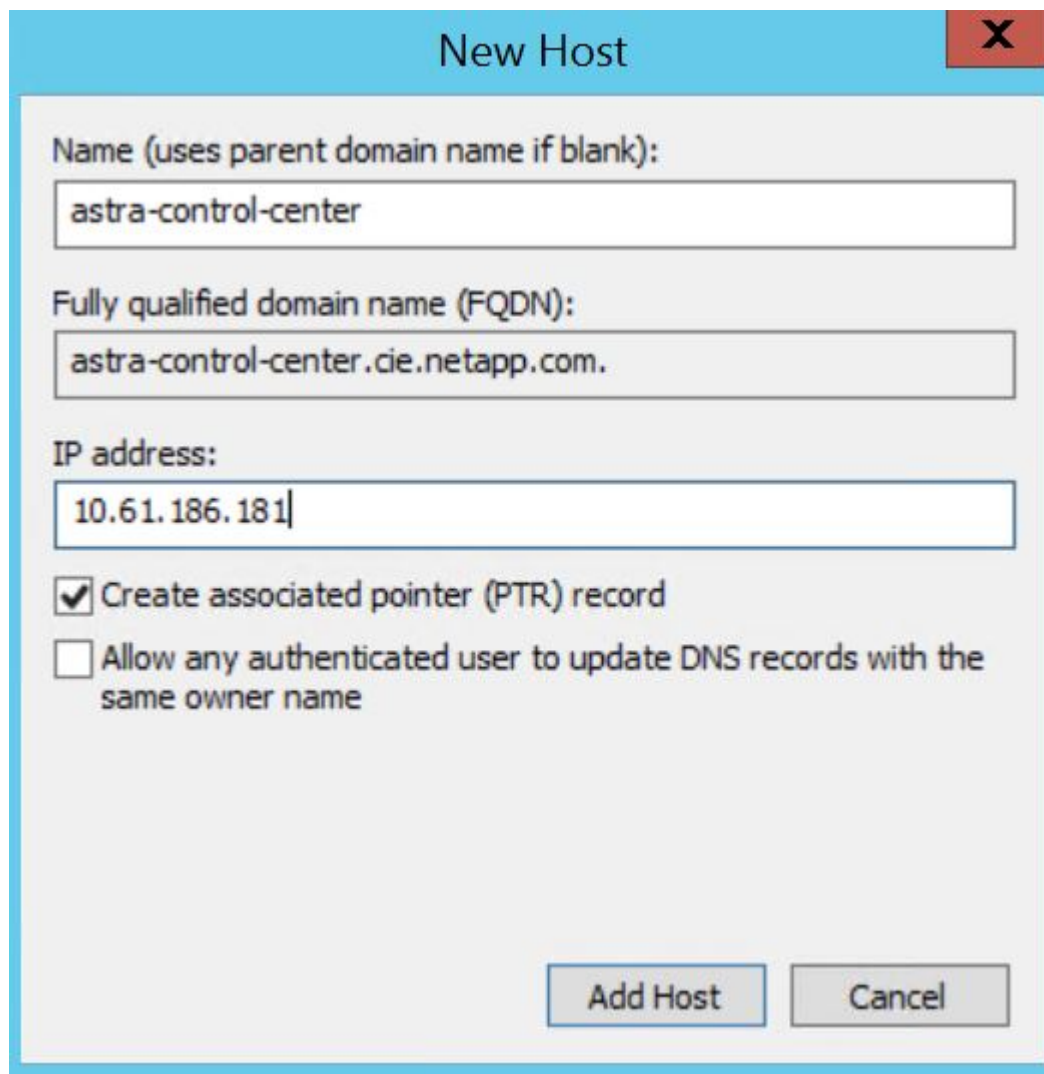
En este ejemplo, la contraseña es ACC-345c55a5-bf2e-21f0-84b8-b6f2bce5e95f.

4. Obtenga la IP del equilibrador de carga del servicio de Traefik.

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

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

5. Agregue una entrada en el servidor DNS apuntando al FQDN que se proporciona en el archivo CRD de Astra Control Center al `EXTERNAL-IP` del servicio de trafik.



New Host

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

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

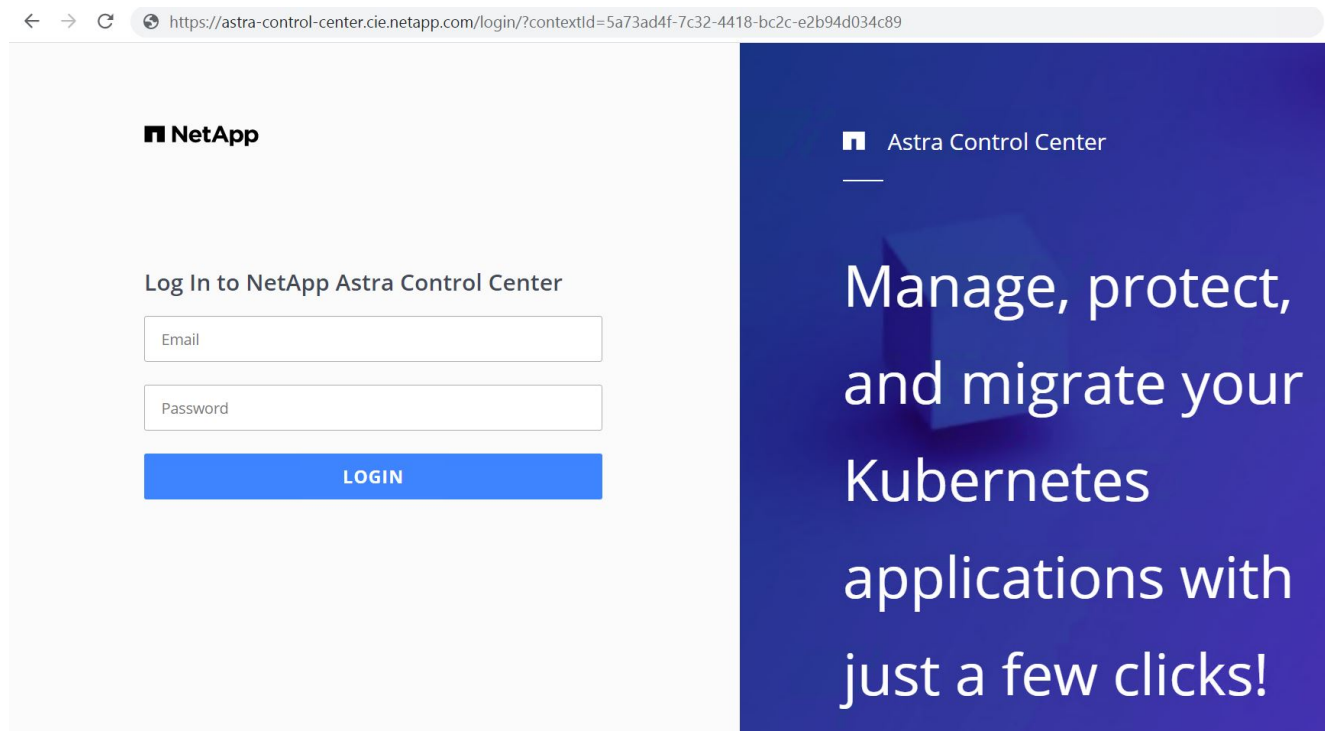
IP address:
10.61.186.181

☒ Create associated pointer (PTR) record

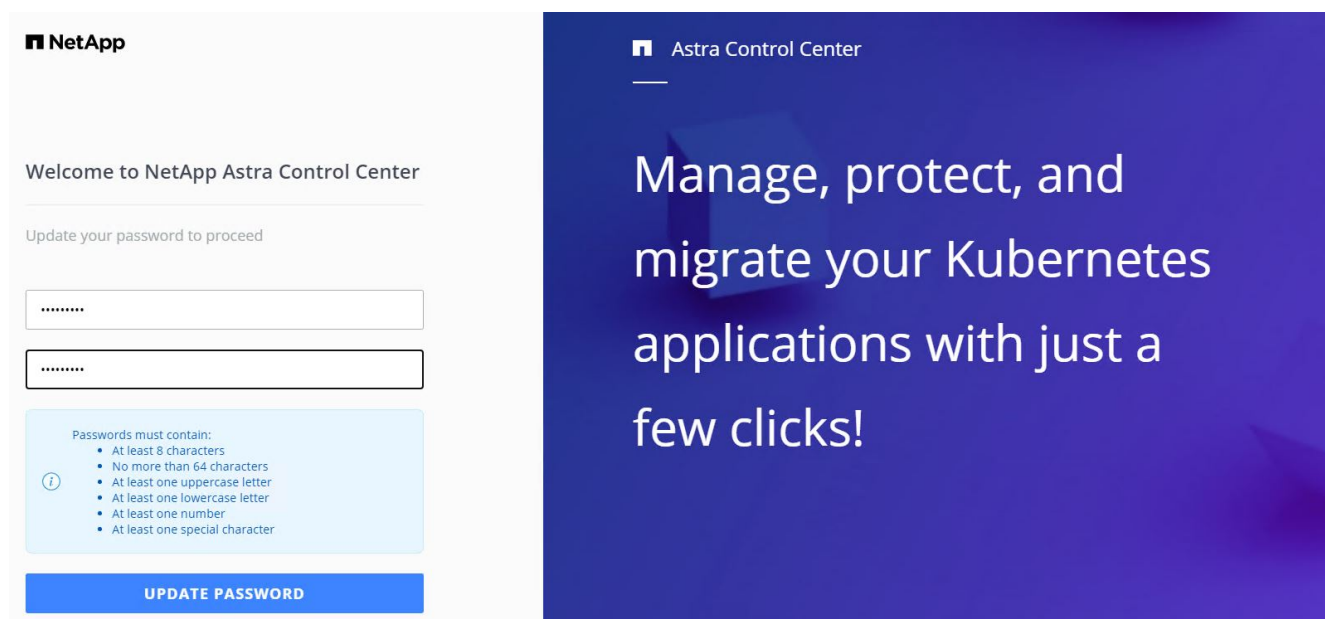
☐ Allow any authenticated user to update DNS records with the same owner name

Add Host Cancel

6. Inicie sesión en la GUI de Astra Control Center navegando por su FQDN.



7. Cuando inicie sesión en la GUI de Astra Control Center por primera vez con la dirección de correo electrónico de administrador proporcionada en CRD, deberá cambiar la contraseña.



8. Si desea agregar un usuario a Astra Control Center, desplácese a cuenta > usuarios, haga clic en Agregar, introduzca los detalles del usuario y haga clic en Agregar.

Add user
✕

USER DETAILS

First name

Nikhil

Last name

Kulkarni

Email address

tme_nik@netapp.com

PASSWORD

Temporary password

Confirm temporary password

ⓘ

Passwords must contain:

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

USER ROLE ⓘ

Role

Owner

▼

Cancel

Add ✓

ADD NEW USER

Add new user

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

9. Astra Control Center requiere una licencia para que funcionen todas las funciones de TI. Para añadir una licencia, vaya a cuenta > Licencia, haga clic en Añadir licencia y cargue el archivo de licencia.

Account

Users

Credentials

Notifications

License

Connections

ASTRA CONTROL CENTER LICENSE

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

Add license

ADD LICENSE

Select and add a license file.

License file

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

⬆

✕

Cancel

Add




Si tiene problemas con la instalación o la configuración de NetApp Astra Control Center, está disponible la base de conocimientos sobre problemas conocidos ["aquí"](#).

Registre sus clústeres de Red Hat OpenShift con Astra Control Center


Para habilitar Astra Control Center para gestionar sus cargas de trabajo, primero debe registrar su clúster Red Hat OpenShift.

Registre clústeres de Red Hat OpenShift

1. El primer paso es agregar los clústeres de OpenShift al Centro de control de Astra y gestionarlos. Vaya a Clusters y haga clic en Add a Cluster, cargue el archivo kubeconfig para el clúster OpenShift y haga clic en Select Storage.

 **Add cluster**

STEP 1/3: CREDENTIALS





CREDENTIALS

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


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

[Upload file](#) | Paste from clipboard

Kubeconfig YAML file
ocp-vmw kubeconfig.txt


 

Credential name
ocp-vmw

 **ADDING A CLUSTER**

Adding a cluster is needed for Astra Control to discover your Kubernetes applications.

Select a cloud provider and input credentials to get started.

Read more in [Clusters](#) .

Cancel

Configure storage →



El archivo kubeconfig se puede generar para autenticarse con un nombre de usuario y una contraseña o un token. Los tokens caducan tras una cantidad limitada de tiempo y es posible que no se pueda acceder al clúster registrado. NetApp recomienda utilizar un archivo kubeconfig con un nombre de usuario y una contraseña para registrar los clústeres de OpenShift en Astra Control Center.

2. Astra Control Center detecta las clases de almacenamiento elegibles. Ahora seleccione la forma en que storagegrid aprovisiona volúmenes mediante Trident con backup de una SVM en ONTAP de NetApp y haga clic en Review. En el panel siguiente, compruebe los detalles y haga clic en Add Cluster.

STORAGE

Existing storage classes are discovered and verified as eligible for use with Astra Control. You can use your existing default, or choose to set a new default at this time.
Applications with persistent volumes on eligible storage classes are validated for use with Astra Control.

Set default	Storage class	Storage provisioner	Reclaim policy	Binding mode	Eligible
<input checked="" type="radio"/>	ocp-trident <small>Default</small>	csi.trident.netapp.io	Delete	Immediate	
<input type="radio"/>	ocp-trident-iscsi	csi.trident.netapp.io	Delete	Immediate	
<input type="radio"/>	project-1-sc	csi.trident.netapp.io	Delete	Immediate	
<input type="radio"/>	thin	kubernetes.io/vsphere-volume	Delete	Immediate	

[← Select credentials](#)
[Review →](#)

3. Registre ambos clústeres de OpenShift como se describe en el paso 1. Cuando se añaden, los clústeres se mueven al estado de detección mientras Astra Control Center los inspecciona e instala los agentes necesarios. El estado del clúster cambia a en ejecución después de que se hayan registrado correctamente.

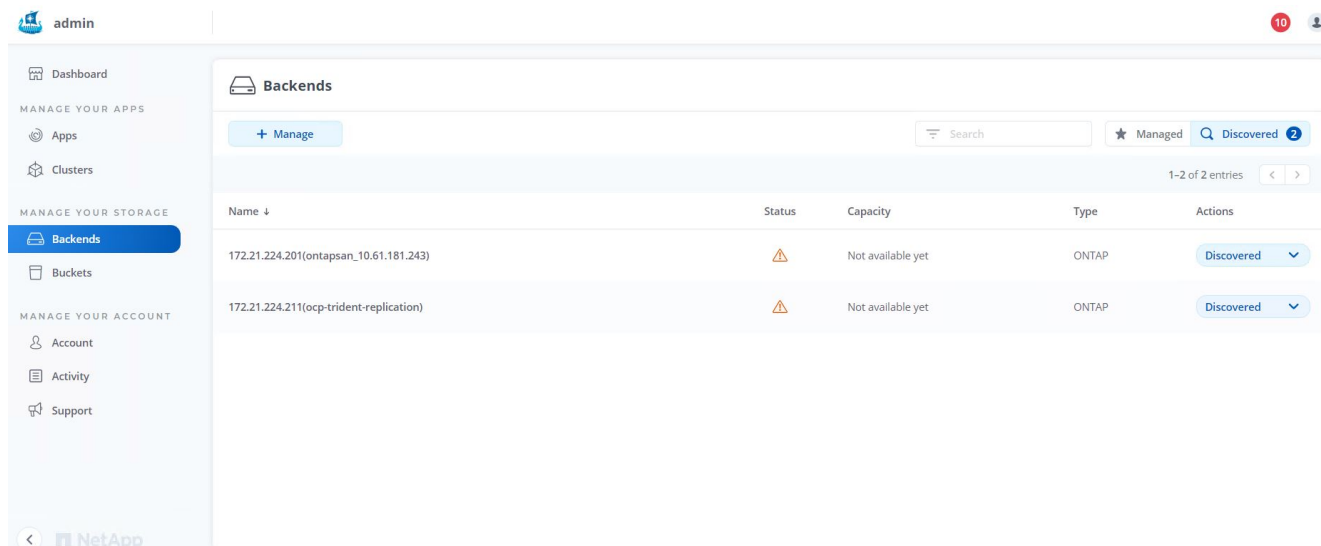
The screenshot shows the Astra Control Center interface. On the left is a sidebar with navigation links: Dashboard, Apps, Clusters (selected), Backends, Buckets, Account, Activity, and Support. The main panel is titled 'Clusters' and contains a table with the following data:

Name	Ready	Type	Version	Actions
ocp-vmw		Red Hat OpenShift	v1.20.0+df9c838	Running
ocp-vmware2		Red Hat OpenShift	v1.20.0+c8905da	Running



Todos los clústeres de Red Hat OpenShift que gestiona Astra Control Center deben tener acceso al registro de imágenes que se utilizó para su instalación, ya que los agentes instalados en los clústeres gestionados extraen las imágenes de ese registro.

4. Importe clústeres de ONTAP como recursos de almacenamiento que Astra Control Center gestiona como back-ends. Cuando se agregan clústeres de OpenShift a Astra y se configura un storagegrid, detecta e inspecciona automáticamente el clúster de ONTAP para respaldar el storagegrid pero no lo importa en el Centro de control de Astra para su gestión.



- Para importar los clústeres de ONTAP, vaya a Back-ends, haga clic en el menú desplegable y seleccione Manage junto al clúster de ONTAP que se va a gestionar. Introduzca las credenciales del clúster de ONTAP, haga clic en revisar información y, a continuación, haga clic en Importar back-end de almacenamiento.

Manage ONTAP storage backend STEP 1/2: CREDENTIALS

CREDENTIALS

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

Cluster management IP address: 172.21.224.201

User name: admin

Password: *****

MANAGE STORAGE BACKEND

Storage backends provide storage to your Kubernetes applications.

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

Read more in [Storage backend](#).

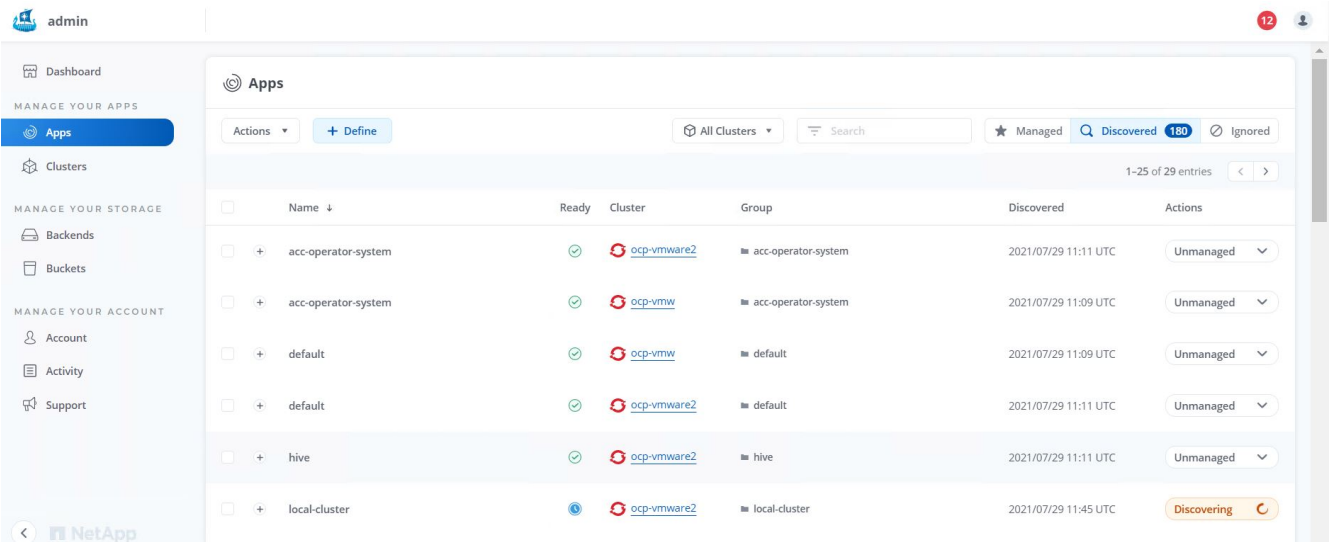
ONTAP

Cancel Review information →

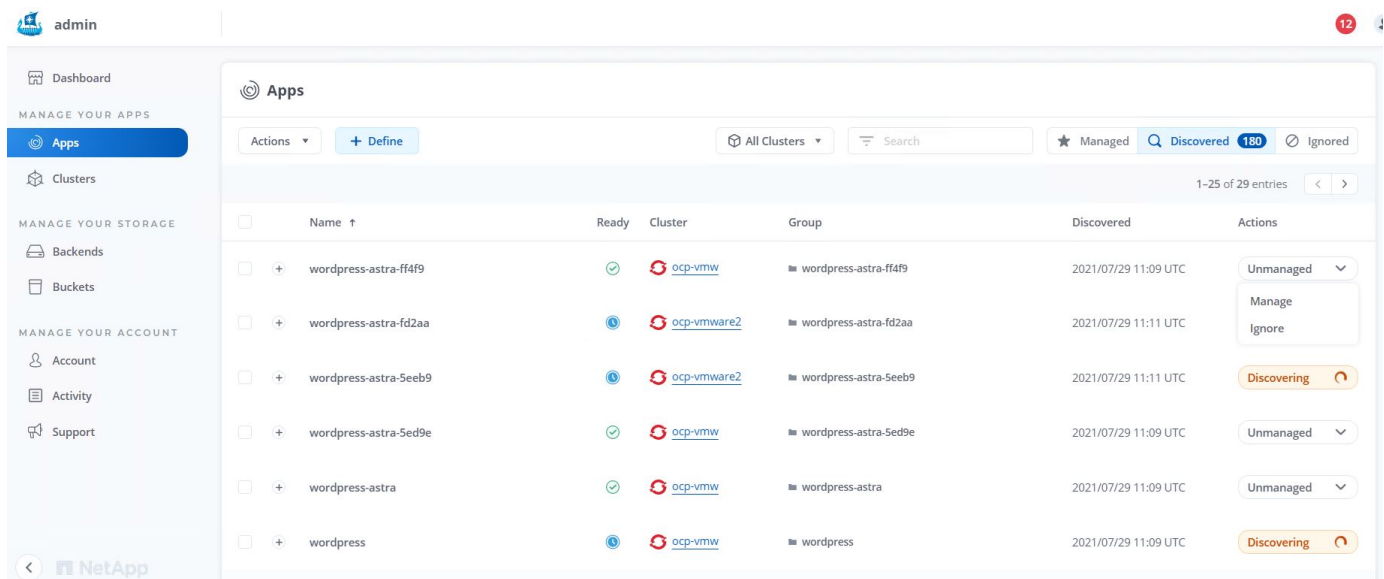
- Una vez añadidos los back-ends, el estado cambia a Available. Estos back-ends ahora tienen información sobre los volúmenes persistentes en el clúster de OpenShift y los volúmenes correspondientes en el sistema ONTAP.

Gestione las aplicaciones

1. Una vez registrados los clústeres de OpenShift y los back-ends de ONTAP con el Centro de control de Astra, el centro de control inicia automáticamente el descubrimiento de las aplicaciones en todos los espacios de nombres que utilizan el sistema storageclass configurado con el back-end de ONTAP especificado.



2. Desplácese a aplicaciones > descubiertas y haga clic en el menú desplegable situado junto a la aplicación que desea gestionar mediante Astra. A continuación, haga clic en gestionar.



1. La aplicación entra en el estado disponible y se puede ver en la ficha gestionado de la sección aplicaciones.

Apps

Actions ▾

+ Define

All Clusters ▾

⌵

Search

★ Managed


🔍 Discovered 175

🚫 Ignored

1-1 of 1 entries

<

>

<input type="checkbox"/>	Name ↓	Ready	Protected	Cluster	Group	Discovered	Actions
<input type="checkbox"/>	wordpress-astra-ff4f9	✔	ℹ	 ocp-vmw	■ wordpress-astra-ff4f9	2021/07/29 11:09 UTC	Available ▾

Proteja sus aplicaciones

Una vez que Astra Control Center gestiona las cargas de trabajo de las aplicaciones, puede configurar los ajustes de protección para esas cargas de trabajo.

Creación de una instantánea de aplicación

Una copia Snapshot de una aplicación crea una copia Snapshot de ONTAP que se puede utilizar para restaurar o clonar la aplicación en un momento específico según esa copia Snapshot.

1. Para tomar una instantánea de la aplicación, desplácese a la ficha aplicaciones > gestionado y haga clic en la aplicación de la que desea realizar una copia snapshot. Haga clic en el menú desplegable junto al nombre de la aplicación y haga clic en Snapshot.

APPLICATION STATUS

Healthy

APPLICATION PROTECTION STATUS

Unprotected

Images

docker.io/bitnami/mariadb:10.5.13-debian-10-r58

docker.io/bitnami/wordpress:5.9.0-debian-10-r1

Protection schedule

Disabled

Group

■ wp

Cluster

Running ▾

Snapshot

Backup

Clone

Restore

Unmanage

2. Introduzca los detalles de la snapshot, haga clic en Siguiente y luego en Snapshot. La creación de la snapshot tarda aproximadamente un minuto y el estado cambia a disponible después de que se cree correctamente la snapshot.

Snapshot application

STEP 1/2: DETAILS

X

SNAPSHOT DETAILS

Name

wp-snapshot-20220228185949

CREATING APPLICATION SNAPSHOTS

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

Read more in [Protect apps](#).

Application

wp

Namespace

wp

Cluster

ocp-vmw

Cancel

Next →

Crear un backup de aplicación

Un backup de una aplicación captura el estado activo de la aplicación y la configuración de sus recursos de TI, los coloca en archivos y los almacena en un bloque de almacenamiento de objetos remotos.

Para realizar la copia de seguridad y la restauración de las aplicaciones gestionadas en el Centro de control de Astra, debe configurar los ajustes de superusuario para los sistemas ONTAP de respaldo como requisito previo. Para ello, introduzca los comandos siguientes.

```
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. Para crear una copia de seguridad de la aplicación gestionada en Astra Control Center, desplácese a la ficha aplicaciones > administradas y haga clic en la aplicación de la que desea realizar una copia de seguridad. Haga clic en el menú desplegable junto al nombre de la aplicación y haga clic en copia de seguridad.

wp

Running ▼

APPLICATION STATUS

Healthy

APPLICATION PROTECTION STATUS

Unprotected

Images

docker.io/bitnami/mariadb:10.5.13-debian-10-r58
 docker.io/bitnami/wordpress:5.9.0-debian-10-r1

Protection schedule

Disabled

Group

wp

Cluster

ocp-vmw

Snapshot

Backup

Clone

Restore

Unmanage

2. Introduzca los detalles de la copia de seguridad, seleccione el bloque de almacenamiento de objetos donde se retengan los archivos de copia de seguridad, haga clic en Siguiente y, tras revisar los detalles, haga clic en Backup. Según el tamaño de la aplicación y los datos, el backup puede tardar varios minutos y el estado del backup pasa a estar disponible después de que el backup se haya completado correctamente.

Backup application

STEP 1/2: DETAILS

X

BACKUP DETAILS

Name

wp-backup

☐ Backup from an existing snapshot

BACKUP DESTINATION

Bucket

na-ocp-astra/na-ocp-acc Available

CREATING APPLICATION BACKUPS

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

Read more in [Application backups](#).

Application

wp

Namespace

wp

Cluster

ocp-vmw

Cancel

Next →

Restaurar una aplicación

Con solo pulsar un botón, puede restaurar una aplicación en el espacio de nombres de origen del mismo clúster o en un clúster remoto para realizar tareas de protección de aplicaciones y recuperación ante desastres.

1. Para restaurar una aplicación, desplácese a la ficha aplicaciones > gestionadas y haga clic en la aplicación en cuestión. Haga clic en el menú desplegable junto al nombre de la aplicación y haga clic en Restore.

wp

Running

APPLICATION STATUS

Healthy

APPLICATION PROTECTION STATUS

Partially protected

Images

docker.io/bitnami/mariadb:10.5.13-debian-10-r58

docker.io/bitnami/wordpress:5.9.0-debian-10-r1

Protection schedule

Disabled

Group

wp

Cluster

ocp-vmw

Snapshot

Backup

Clone

Restore

Unmanage

2. Introduzca el nombre del espacio de nombres de la restauración, seleccione el clúster donde desea restaurarlo y elija si desea restaurarlo desde la copia de Snapshot existente o desde el backup de la aplicación. Haga clic en Siguiente.

Restore application

STEP 1/2: DETAILS

RESTORE DETAILS

Destination cluster

ocp-vmw

Destination namespace

wp

RESTORE SOURCE

Filter

Snapshots

Backups

Application backup	Ready	On-Schedule/On-Demand	Created ↑
wp-backup	✓	On-Demand	2022/02/28 18:54 UTC

RESTORING APPLICATIONS

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

- Application wp
- Namespace wp
- Cluster ocp-vmw

Cancel

Next →

- En el panel de revisión, introduzca `restore` Y haga clic en Restaurar después de haber revisado los detalles.

Restore application

STEP 2/2: SUMMARY

REVIEW RESTORE INFORMATION

⚠

All existing resources associated with this application will be deleted and replaced with the source backup "wp-backup" taken on 2022/02/28 18:54 UTC. Persistent volumes will be deleted and recreated. External resources with dependencies on this application may be impacted.

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

BACKUP

wp-backup

ORIGINAL GROUP

wp

ORIGINAL CLUSTER

ocp-vmw

RESOURCE LABELS

ClusterRole

kubernetes.io/bootstrapping: rbac-defaults +1

ClusterRoleBinding

RESTORE

wp

DESTINATION GROUP

wp

DESTINATION CLUSTER

ocp-vmw

RESOURCE LABELS

ClusterRole

kubernetes.io/bootstrapping: rbac-defaults +1

ClusterRoleBinding

Are you sure you want to restore the application "wp"?

Type **restore** below to confirm.

Confirm to restore

restore

← Back

Restore ✓

- La nueva aplicación pasa al estado de restauración mientras Astra Control Center restaura la aplicación en el clúster seleccionado. Una vez que todos los recursos de la aplicación son instalados y detectados por Astra, la aplicación pasa al estado disponible.

Actions

+ Define

Search

★

🔍

110

⌵

🔄

1-1 of 1 entries

<

>

<input type="checkbox"/>	Name ↓	Ready	Protected	Cluster	Group	Discovered	Actions
<input type="checkbox"/>	wp	✓	ℹ	ocp-vmw	wp	2022/02/28 18:34 UTC	Available ⌵

Clonar una aplicación

Es posible clonar una aplicación en el clúster de origen o en un clúster remoto para fines de desarrollo/pruebas o protección de aplicaciones y recuperación ante desastres. La clonación de una aplicación dentro del mismo clúster en el mismo back-end de almacenamiento utiliza la tecnología FlexClone de NetApp, que clona las RVP de forma instantánea y ahorra espacio de almacenamiento.

1. Para clonar una aplicación, vaya a la ficha aplicaciones > administradas y haga clic en la aplicación en cuestión. Haga clic en el menú desplegable junto al nombre de la aplicación y haga clic en Clonar.

wp

APPLICATION STATUS
 Healthy

APPLICATION PROTECTION STATUS
 Partially protected

Images
 docker.io/bitnami/mariadb:10.5.13-debian-10-r58
 docker.io/bitnami/wordpress:5.9.0-debian-10-r1

Protection schedule
 Disabled

Group
 wp

Cluster
 ocp-vmw

Running ▾
 Snapshot
 Backup
 Clone
 Restore
 Unmanage

2. Introduzca los detalles del nuevo espacio de nombres, seleccione el clúster al que desea clonarlo y elija si desea clonarlo desde una copia de Snapshot existente o un backup o el estado actual de la aplicación. A continuación, haga clic en Siguiente y en Clonar en el panel de revisión una vez que haya revisado los detalles.

Clone application

STEP 1/2: DETAILS

✕

CLONE DETAILS

Clone name
 wp-clone

Clone namespace
 wp-clone

Destination cluster
 ocp-vmw

☐ Clone from an existing snapshot or backup

CLONING APPLICATIONS

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

Read more in [Clone applications](#)

Application
 wp

Namespace
 wp

Cluster
 ocp-vmw

Cancel

Next →



3. La nueva aplicación pasa al estado de descubrimiento mientras Astra Control Center crea la aplicación en el clúster seleccionado. Una vez que todos los recursos de la aplicación son instalados y detectados por Astra, la aplicación pasa al estado disponible.

Applications

Actions ▾ + Define

📦 ▾ Search 110

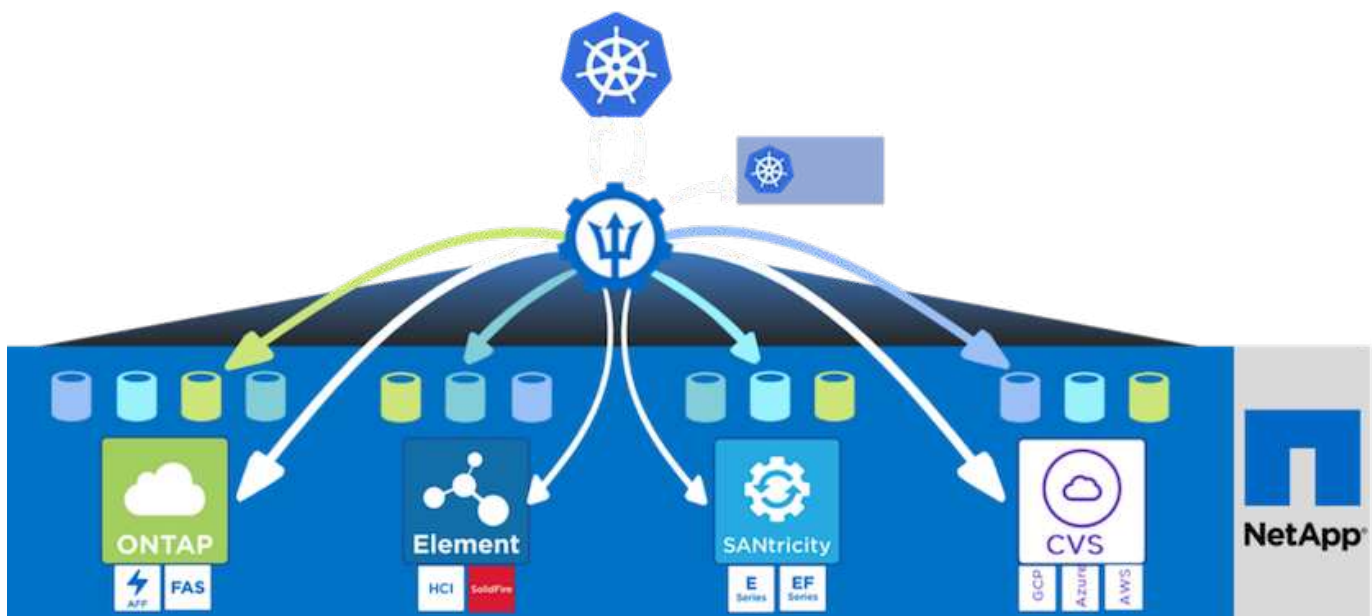
1-2 of 2 entries

<input type="checkbox"/>	Name ↓	Ready	Protected	Cluster	Group	Discovered	Actions
<input type="checkbox"/>	wp	✓	ⓘ	 ocp-vmw	wp	2022/02/28 18:34 UTC	Available ▾
<input type="checkbox"/>	wp-clone	✓	⚠	 ocp-vmw	wp-clone	2022/02/28 19:21 UTC	Available ▾

Descripción general de Astra Trident

Astra Trident es un orquestador de almacenamiento de código abierto y totalmente compatible para contenedores y distribuciones de Kubernetes, incluido Red Hat OpenShift. Trident funciona con toda la cartera de almacenamiento de NetApp, incluidos los sistemas de almacenamiento ONTAP y Element de NetApp, y también admite conexiones NFS e iSCSI. Trident acelera el flujo de trabajo de DevOps al permitir que los usuarios finales aprovisionen y gestionen el almacenamiento desde sus sistemas de almacenamiento de NetApp sin necesidad de intervención del administrador de almacenamiento.

Un administrador puede configurar varios back-ends de almacenamiento a partir de necesidades de proyectos y modelos de sistema de almacenamiento que permiten funciones de almacenamiento avanzadas, como compresión, tipos de disco específicos o niveles de calidad de servicio que garantizan un cierto nivel de rendimiento. Una vez definidas estos back-ends pueden ser utilizados por los desarrolladores en sus proyectos para crear reclamaciones de volumen persistente (RVP) y conectar almacenamiento persistente a sus contenedores bajo demanda.



Astra Trident tiene un rápido ciclo de desarrollo y, al igual que Kubernetes, se publica cuatro veces al año.

La última versión de Astra Trident se lanzó en enero de 2022 en 22.01. Existe una matriz de compatibilidad con la versión de Trident probada en la que se puede encontrar la distribución de Kubernetes ["aquí"](#).

A partir del lanzamiento de la versión 20.04, el operador de Trident realiza la configuración de Trident. El operador facilita las puestas en marcha a gran escala y ofrece soporte adicional, incluida la reparación automática de los pods que se implementan como parte de la instalación de Trident.

Con la versión 21.01, se puso a disposición un gráfico Helm para facilitar la instalación del operador Trident.

Descargue Astra Trident

Para instalar Trident en el clúster de usuarios implementado y aprovisionar un volumen persistente, complete los siguientes pasos:

1. Descargue el archivo de instalación en la estación de trabajo de administración y extraiga el contenido. La versión actual de Trident es la 22.01, que se puede descargar ["aquí"](#).

```
[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. Extraiga la instalación de Trident del paquete descargado.

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

Instale el operador Trident con Helm

1. En primer lugar, defina la ubicación del clúster de usuarios kubeconfig Archivo como variable de entorno para no tener que referirla, porque Trident no tiene opción para pasar este archivo.

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

2. Ejecute el comando Helm para instalar al operador Trident desde el tarball en el directorio helm mientras crea el espacio de nombres de trident en su clúster de usuarios.

```
[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. Puede verificar que Trident se haya instalado correctamente comprobando los pods que se ejecutan en el espacio de nombres o mediante el binario `tridentctl` para comprobar la versión instalada.

```
[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



En algunos casos, los entornos del cliente pueden requerir la personalización de la puesta en marcha de Trident. En estos casos, también es posible instalar manualmente el operador Trident y actualizar los manifiestos incluidos para personalizar la implementación.

Instale manualmente el operador de Trident

1. Primero, establezca la ubicación del clúster de usuarios kubeconfig Archivo como variable de entorno para no tener que referirla, porque Trident no tiene opción para pasar este archivo.

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

2. La `trident-installer` el directorio contiene manifiestos para definir todos los recursos necesarios. Con los manifiestos adecuados, cree la `TridentOrchestrator` definición de recursos personalizados.

```
[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. Si no existe ninguno, cree un espacio de nombres Trident en el clúster mediante el manifiesto proporcionado.

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

4. Cree los recursos necesarios para la puesta en marcha del operador Trident, como, por ejemplo, la

ServiceAccount para el operador, un ClusterRole y.. ClusterRoleBinding para la ServiceAccount, un dedicado `PodSecurityPolicy`o el propio operador.

```
[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. Puede comprobar el estado del operador después de desplegarlo con los siguientes comandos:

```
[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. Con el operador puesto en marcha, ahora podemos utilizarlo para instalar Trident. Esto requiere crear un 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. Puede verificar que Trident se haya instalado correctamente comprobando los pods que se ejecutan en el espacio de nombres o mediante el binario `tridentctl` para comprobar la versión instalada.

```

[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        |
+-----+-----+

```

Prepare los nodos de trabajo para el almacenamiento

NFS

La mayoría de las distribuciones de Kubernetes incluyen los paquetes y utilidades para montar los back-ends de NFS instalados de forma predeterminada, incluido Red Hat OpenShift.

Sin embargo, para NFSv3, no hay ningún mecanismo para negociar la concurrencia entre el cliente y el servidor. Por ello, el número máximo de entradas de la tabla de ranuras `sunrpc` del lado del cliente debe

sincronizarse manualmente con el valor compatible del servidor para garantizar el mejor rendimiento de la conexión NFS sin que el servidor tenga que reducir el tamaño de la ventana de la conexión.

En el caso de ONTAP, el número máximo admitido de entradas de la tabla de ranuras de sunrpc es de 128, es decir, ONTAP puede atender 128 solicitudes de NFS simultáneas en un momento. Sin embargo, de forma predeterminada, Red Hat CoreOS/Red Hat Enterprise Linux tiene un máximo de 65,536 entradas de tabla de ranuras sunrpc por conexión. Tenemos que establecer este valor en 128 y esto se puede hacer usando el operador de configuración de máquina (MCO) en OpenShift.

Para modificar el número máximo de entradas de la tabla de ranuras de sunrpc en los nodos de trabajo de OpenShift, realice los pasos siguientes:

1. Inicie sesión en la consola web de OCP y desplácese hasta Compute > Machine Configs. Haga clic en Crear configuración de máquina. Copie y pegue el archivo YAML y haga clic en Crear.

```
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. Una vez creada la MCO, la configuración debe aplicarse a todos los nodos de trabajo y reiniciarse uno por uno. Todo el proceso tarda aproximadamente de 20 a 30 minutos. Compruebe si la configuración de la máquina se aplica mediante `oc get mcp` y asegúrese de que el grupo de configuración de la máquina para los trabajadores esté actualizado.

```
[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

Para preparar los nodos de trabajo y permitir la asignación de volúmenes de almacenamiento en bloques mediante el protocolo iSCSI, debe instalar los paquetes necesarios para que admitan dicha funcionalidad.

En Red Hat OpenShift, esto se maneja aplicando una MCO (operador de configuración de máquina) a su clúster después de desplegarla.

Para configurar los nodos de trabajo para que ejecuten servicios iSCSI, lleve a cabo los siguientes pasos:

1. Inicie sesión en la consola web de OCP y desplácese hasta Compute > Machine Configs. Haga clic en Crear configuración de máquina. Copie y pegue el archivo YAML y haga clic en Crear.

Cuando no se utiliza el acceso múltiple:

```
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: ""
```

Cuando se utiliza el acceso múltiple:

```

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,ZGVmYXVsdHMgewogICAgICAgIHVzZXJfZnJpZW5kbHlfbmFtZXNMgbm8KICAgICAgICBmaW5kX211bHRpcGF0aHMgbm8KfQoKYmxhY2tsaXN0X2V4Y2VwdGlvbnMgewogICAgICAgIHByb3BlcnR5ICIoU0NTSV9JREV0VF98SURfV1dOKSikfQoKYmxhY2tsaXN0IHsKfQoK
          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. Una vez creada la configuración, se tarda entre 20 y 30 minutos aproximadamente en aplicar la configuración a los nodos de trabajo y volver a cargarlos. Compruebe si la configuración de la máquina se aplica mediante `oc get mcp` y asegúrese de que el grupo de configuración de la máquina para los trabajadores esté actualizado. También puede iniciar sesión en los nodos de trabajo para confirmar que el servicio `iscsid` se está ejecutando (y el servicio `multipathd` se está ejecutando si se utiliza la función `multivía`).

```
[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
```



También es posible confirmar que MachineConfig se ha aplicado correctamente y que los servicios se han iniciado como se esperaba ejecutando el `oc debug` comando con los indicadores apropiados.

Cree back-ends del sistema de almacenamiento

Una vez finalizada la instalación del operador de Astra Trident, debe configurar el back-end para la plataforma de almacenamiento específica de NetApp que esté usando. Siga los siguientes enlaces para continuar con la instalación y configuración de Astra Trident.

- ["NFS de ONTAP de NetApp"](#)
- ["ISCSI de ONTAP de NetApp"](#)
- ["ISCSI de NetApp Element"](#)

Configuración NFS de ONTAP de NetApp

Para habilitar la integración de Trident con el sistema de almacenamiento ONTAP de NetApp, debe crear un back-end que permita la comunicación con el sistema de almacenamiento.

1. Hay archivos de fondo de ejemplo disponibles en el archivo de instalación descargado en `sample-input` jerarquía de carpetas. Para los sistemas ONTAP de NetApp que sirven NFS, copie el `backend-ontap-nas.json` archivar en el directorio de trabajo y editar el archivo.

```
[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. Editar la `backendName`, `managementLIF`, `dataLIF`, `svm`, nombre de usuario, y los valores de contraseña en este archivo.

```
{
  "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"
}
```



Se recomienda definir el valor de `backendName` personalizado como una combinación de `storageDriverName` y `DataLIF` que sirve NFS para una identificación sencilla.

3. Con este archivo back-end en su lugar, ejecute el siguiente comando para crear su primer back-end.

```
[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. Con el back-end creado, debe crear después una clase de almacenamiento. Al igual que con el backend, existe un archivo de clase de almacenamiento de ejemplo que se puede editar para el entorno disponible en la carpeta de entradas de ejemplo. Cópielo en el directorio de trabajo y realice las modificaciones necesarias para reflejar el backend creado.

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

- La única edición que se debe realizar en este archivo es definir `backendType` valor asignado al nombre del controlador de almacenamiento desde el back-end recién creado. Observe también el valor del campo de nombre, al que se debe hacer referencia en un paso posterior.

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



Hay un campo opcional llamado `fsType` que se define en este archivo. Esta línea se puede eliminar en los back-ends de NFS.

6. Ejecute el `oc` comando para crear la clase de almacenamiento.

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

7. Con la clase de almacenamiento creada, debe crear la primera reclamación de volumen persistente (RVP). Hay una muestra `pvc-basic.yaml` archivo que se puede utilizar para realizar esta acción ubicada también en entradas de ejemplo.

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

8. La única edición que se debe realizar en este archivo es asegurarse de que `storageClassName` el campo coincide con el que se acaba de crear. La definición de PVC se puede personalizar aún más según sea necesario para el aprovisionamiento de la carga de trabajo.

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

9. Cree el PVC emitiendo el `oc` comando. La creación puede tardar un poco de tiempo, según el tamaño del volumen de backup que se esté creando, para que pueda ver el proceso a medida que finalice.

```
[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
```

Configuración de iSCSI de ONTAP de NetApp

Para habilitar la integración de Trident con el sistema de almacenamiento ONTAP de NetApp, debe crear un back-end que permita la comunicación con el sistema de almacenamiento.

1. Hay archivos de fondo de ejemplo disponibles en el archivo de instalación descargado en `sample-input` jerarquía de carpetas. Para los sistemas ONTAP de NetApp que proporcionan servicio iSCSI, copie el `backend-ontap-san.json` archivar en el directorio de trabajo y editar el archivo.

```
[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. Edite los valores managementLIF, dataLIF, svm, username y password de este archivo.

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

3. Con este archivo back-end en su lugar, ejecute el siguiente comando para crear su primer back-end.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-san.json
```

NAME	STORAGE DRIVER	UUID
ontapsan_10.61.181.241	ontap-san	6788533c-7fea-4a35-b797-fb9bb3322b91
online	0	

4. Con el back-end creado, debe crear después una clase de almacenamiento. Al igual que con el backend, existe un archivo de clase de almacenamiento de ejemplo que se puede editar para el entorno disponible en la carpeta de entradas de ejemplo. Cópielo en el directorio de trabajo y realice las modificaciones necesarias para reflejar el backend creado.

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

- La única edición que se debe realizar en este archivo es definir `backendType` valor asignado al nombre del controlador de almacenamiento desde el back-end recién creado. Observe también el valor del campo de nombre, al que se debe hacer referencia en un paso posterior.

```

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

```



Hay un campo opcional llamado `fsType` que se define en este archivo. En los back-ends de iSCSI, este valor se puede establecer en un tipo de sistema de archivos de Linux específico (XFS, ext4, etc.) o se puede eliminar para permitir a OpenShift decidir qué sistema de archivos utilizar.

6. Ejecute el `oc` comando para crear la clase de almacenamiento.

```

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

```

7. Con la clase de almacenamiento creada, debe crear la primera reclamación de volumen persistente (RVP). Hay una muestra `pvc-basic.yaml` archivo que se puede utilizar para realizar esta acción ubicada también en entradas de ejemplo.

```

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

```

8. La única edición que se debe realizar en este archivo es asegurarse de que `storageClassName` el campo coincide con el que se acaba de crear. La definición de PVC se puede personalizar aún más según sea necesario para el aprovisionamiento de la carga de trabajo.

```

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

```


9. Cree el PVC emitiendo el `oc` comando. La creación puede tardar un poco de tiempo, según el tamaño del volumen de backup que se esté creando, para que pueda ver el proceso a medida que finalice.

```
[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
```

Configuración de iSCSI de NetApp Element

Para habilitar la integración de Trident con el sistema de almacenamiento NetApp Element, debe crear un back-end que permita la comunicación con el sistema de almacenamiento mediante el protocolo iSCSI.

1. Hay archivos de fondo de ejemplo disponibles en el archivo de instalación descargado en `sample-input` jerarquía de carpetas. Para sistemas NetApp Element que sirven iSCSI, copie el `backend-solidfire.json` archivar en el directorio de trabajo y editar el archivo.

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

- a. Edite el valor de usuario, contraseña y MVIP de en el `EndPoint` línea.
- b. Edite el `SVIP` valor.

```
{
  "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. Con este archivo back-end en su sitio, ejecute el siguiente comando para crear su primer back-end.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create backend -f backend-solidfire.json
```

NAME	STATE	VOLUMES	STORAGE DRIVER	UUID
solidfire_10.61.180.200	online	0	solidfire-san	b90783ee-e0c9-49af-8d26-3ea87ce2efdf

3. Con el back-end creado, debe crear después una clase de almacenamiento. Al igual que con el backend, existe un archivo de clase de almacenamiento de ejemplo que se puede editar para el entorno disponible en la carpeta de entradas de ejemplo. Cópielo en el directorio de trabajo y realice las modificaciones necesarias para reflejar el backend creado.

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

4. La única edición que se debe realizar en este archivo es definir `backendType` valor asignado al nombre del controlador de almacenamiento desde el back-end recién creado. Observe también el valor del campo de nombre, al que se debe hacer referencia en un paso posterior.

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



Hay un campo opcional llamado `fsType` que se define en este archivo. En los back-ends de iSCSI, este valor se puede establecer en un tipo de sistema de archivos de Linux específico (XFS, ext4, etc.), o se puede eliminar para permitir a OpenShift decidir qué sistema de archivos utilizar.

5. Ejecute el `oc` comando para crear la clase de almacenamiento.

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

6. Con la clase de almacenamiento creada, debe crear la primera reclamación de volumen persistente (RVP). Hay una muestra `pvc-basic.yaml` archivo que se puede utilizar para realizar esta acción ubicada también en entradas de ejemplo.

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

7. La única edición que se debe realizar en este archivo es asegurarse de que `storageClassName` el campo coincide con el que se acaba de crear. La definición de PVC se puede personalizar aún más según sea necesario para el aprovisionamiento de la carga de trabajo.

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

8. Cree el PVC emitiendo el `oc` comando. La creación puede tardar un poco de tiempo, según el tamaño del volumen de backup que se esté creando, para que pueda ver el proceso a medida que finalice.

```
[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
```

Información de copyright

Copyright © 2024 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPCIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.